Adaptation, Learning, and Optimization over Networks

Ali H. Sayed
University of California at Los Angeles

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Machine Learning

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

A. H. Sayed. *Adaptation, Learning, and Optimization over Networks*. Foundations and Trends® in Machine Learning, vol. 7, no. 4-5, pp. 311–801, 2014.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-60198-850-8

Foundations and Trends® in Machine Learning
Volume 7, Issue 4-5, 2014

Editorial Board

Foundations and Trends® in Machine Learning
Vol. 7, No. 4-5 (2014) 311–801

DOI: 10.1561/2200000051

now
the essence of knowledge

Adaptation, Learning, and Optimization over Networks

Ali H. Sayed
University of California at Los Angeles

Contents

Abstract

This work deals with the topic of information processing over graphs. The presentation is largely self-contained and covers results that relate to the analysis and design of multi-agent networks for the distributed solution of optimization, adaptation, and learning problems from streaming data through localized interactions among agents. The results derived in this work are useful in comparing network topologies against each other, and in comparing networked solutions against centralized or batch implementations. There are many good reasons for the peaked interest in distributed implementations, especially in this day and age when the word "network" has become commonplace whether one is referring to social networks, power networks, transportation networks, biological networks, or other types of networks. Some of these reasons have to do with the benefits of cooperation in terms of improved performance and improved resilience to failure. Other reasons deal with privacy and secrecy considerations where agents may not be comfortable sharing their data with remote fusion centers. In other situations, the data may already be available in dispersed locations, as happens with cloud computing. One may also be interested in learning through data mining from big data sets. Motivated by these considerations, this work examines the limits of performance of distributed stochastic-gradient solutions and discusses procedures that help bring forth their potential more fully. The presentation adopts a useful statistical framework and derives performance results that elucidate the mean-square stability, convergence, and steady-state behavior of the learning networks. The work also illustrates how distributed processing over graphs gives rise to some revealing phenomena due to the coupling effect among the agents. These phenomena are discussed in the context of adaptive networks, along with examples from a variety of areas including distributed sensing, intrusion detection, distributed estimation, online adaptation, network system theory, and machine learning.

A. H. Sayed. *Adaptation, Learning, and Optimization over Networks*. Foundations and Trends® in Machine Learning, vol. 7, no. 4-5, pp. 311–801, 2014.

DOI: 10.1561/2200000051.

1

Motivation and Notation

1.1 Introduction

Network science is a fascinating field that is evolving rapidly across many domains [15, 19, 92, 121, 154, 178, 207]. As remarked in [207], and for long, classical system and learning theories have focused on optimizing stand-alone systems or learners with great success. Nevertheless, progress in recent decades in the biological sciences [16, 50, 131, 146], animal behavior studies [7, 50, 79, 90, 187, 219], and the neuroscience of the brain [20, 49, 225], has revealed remarkable patterns of organization and structured complexity in the behavior of biological networks, animal groups, and in the dynamics of brain connectivity. These studies have brought forward notable examples of complex systems that derive their sophistication from coordination among simpler units and from the aggregation and processing of decentralized pieces of information. While each unit in these systems is not capable of sophisticated behavior on its own, it is the interaction among the constituents that leads to systems that are resilient to failure and that are capable of adjusting their behavior in response to changes in their environment.

These discoveries have motivated diligent efforts towards a deeper understanding of information processing, adaptation, and learning over

complex networks in several disciplines including machine learning, optimization, control, economics, biological sciences, information sciences, and the social sciences. A common goal in these investigations has been to develop theory and tools that enable the design of networks with sophisticated learning and processing abilities, such as networks that are able to solve important inference and optimization tasks in a distributed manner by relying on agents that interact locally and do not rely on fusion centers to collect and process their information.

1.2 Biological Networks

Examples abound for the viability of such designs in the realm of biological networks. Nature is laden with examples of networks exhibiting sophisticated behavior that arises from interactions among agents of limited abilities. For example, fish schools are unusually skilled at navigating their environment with remarkable discipline and at configuring the topology of their school in the face of danger from predators [79, 187]; when a predator is sighted or sensed, the entire school of fish adjusts its configuration to let the predator through and then coalesces again to continue its schooling behavior. It is reasonable to assume that this complex behavior is the result of sensing information spreading fast across the school of fish through local interactions among adjacent members of the school. Likewise, in bee swarms, it is observed that only a small fraction of the agents (about 5%) are informed and this small fraction of agents is still capable of guiding an entire swarm of bees to their new hive [12, 22, 125, 219]. It is a remarkable property of biological networks and animal groups that sophisticated behavior is able to arise from simple interactions among limited agents [119, 199, 228].

1.3 Distributed Processing

Motivated by these observations, this work deals with the topic of information processing over graphs and how collaboration among agents in a network can lead to superior adaptation and learning performance. The presentation covers results and tools that relate to the analysis and design of networks that are able to solve optimization, adaptation, and

learning problems in an efficient and distributed manner from streaming data through localized interactions among their agents.

The treatment extends the presentation from [207] in several directions[1] and covers three intertwined topics: (a) how to perform distributed *optimization* over networks; (b) how to perform distributed *adaptation* over networks; and (c) how to perform distributed *learning* over networks. In these three domains, we examine and compare the advantages and limitations of non-cooperative, centralized, and distributed *stochastic-gradient* solutions. In the non-cooperative mode of operation, agents act independently of each other in their pursuit of their desired objective. In the centralized mode of operation, agents transmit their (collected or processed) data to a fusion center, which is capable of processing the data centrally. The fusion center then shares the results of the analysis back with the distributed agents. While centralized solutions can be powerful, they still suffer from some limitations. First, in real-time applications where agents collect data continuously, the repeated exchange of information back and forth between the agents and the fusion center can be costly especially when these exchanges occur over wireless links or require nontrivial routing resources. Second, in some sensitive applications, agents may be reluctant to share their data with remote centers for various reasons including privacy and secrecy considerations. More importantly perhaps, centralized solutions have a critical point of failure: if the central processor fails, then this solution method collapses altogether.

Distributed implementations, on the other hand, pursue the desired objective through *localized* interactions among the agents. In the distributed mode of operation, agents are connected by a topology and they are permitted to share information only with their immediate neighbors. There are many good reasons for the peaked interest in such distributed solutions, especially in this day and age when the word "network" has become commonplace whether one is referring to social networks, power networks, transportation networks, biological networks, or other types of networks. Some of these reasons have to do

[1] The author is grateful to IEEE for allowing reproduction of material from [207] in this work.

with the benefits of cooperation in terms of improved performance and improved robustness and resilience to failure. Other reasons deal with privacy and secrecy considerations where agents may not be comfortable sharing their data with remote fusion centers. In other situations, the data may already be available in dispersed locations, as happens with cloud computing. One may also be interested in learning and extracting information through data mining from large data sets. Decentralized learning procedures offer an attractive approach to dealing with such large data sets. Decentralized mechanisms can also serve as important enablers for the design of robotic swarms, which can assist in the exploration of disaster areas.

For these various reasons, we devote some good effort in this work towards quantifying the limits of performance of distributed solutions and towards discussing design procedures that can bring forth their potential more fully. Our emphasis is on solutions that are able to learn from streaming data. In particular, we shall study three families of distributed strategies: (a) incremental strategies, (b) consensus strategies, and (c) diffusion strategies — see Chapter 7. We shall derive expressions that quantify the behavior of the distributed algorithms and use the expressions to compare their performance and to illustrate under what conditions network cooperation is beneficial to the learning and adaptation process. While the social benefit, defined as the average performance across the network, generally improves through cooperation, it is not necessarily the case that the individual agents will always benefit from cooperation: some agents may see their performance degrade relative to the non-cooperative mode of operation [214, 276]. This observation will motivate us to seek optimized combination policies that enable all agents in a network to enhance their performance through cooperation.

1.4 Adaptive Networks

We shall study distributed solutions in the context of adaptive networks [207, 208, 214], which consist of a collection of agents *with* adaptation and learning abilities. The agents are linked together through a topol-

ogy and they interact with each other through localized *in-network* processing to solve inference and optimization problems in a fully distributed and online manner. The continuous sharing and diffusion of information across the network enables the agents to respond in real-time to drifts in the data and to changes in the network topology. Such networks are scalable, robust to node and link failures, and are particularly suitable for learning from big data sets by tapping into the power of collaboration among distributed agents. The networks are also endowed with cognitive abilities [108, 207] due to the sensing abilities of their agents, their interactions with their neighbors, and an embedded feedback mechanism for acquiring and refining information. Each agent is not only capable of experiencing the environment directly, but it also receives information through interactions with its neighbors and processes this information to drive its learning process.

Adaptive networks are well-suited to perform decentralized information processing tasks. They are also well-suited to model several forms of complex behavior exhibited by biological [16, 50, 131, 146] and social networks [15, 77, 92, 121, 229] such as fish schooling [187], prey-predator maneuvers [105, 170], bird formations [110, 119], bee swarming [12, 22, 125, 219], bacteria motility [25, 188, 257], and social and economic interactions [98, 103]. Examples of references that discuss applications of the *diffusion* distributed algorithms studied in this work to problems involving biological and social networks include [56, 65, 155, 212, 214, 245, 246, 249, 275]. Examples of references that discuss applications of *consensus* implementations include [2, 18, 64, 80, 118, 122, 123, 180, 183, 184, 198, 199, 254]. We do not discuss biological networks in this work and refer the reader instead to the above references; the survey article [214] provides some further motivation.

1.5 Organization

This work is largely self-contained. It provides an extended treatment of topics presented in condensed form in the survey [207], and of several other additional topics. For maximal benefit, readers may review

first the background material in Appendices A through G on complex gradient vectors and Hessian matrices, convex functions, mean-value theorems, Lipschitz conditions, matrix theory, and logistic regression.

In preparation for the study of multi-agent networks, Chapters 2–4 review some fundamental results on optimization, adaptation, and learning by *single* stand-alone agents. The emphasis is on stochastic-gradient constructions. The presentation in these chapters provides insights that will be useful in our subsequent study of adaptation and learning by a collection of networked agents. This latter study is more demanding due to the coupling among interacting agents, and due to the fact that networks are generally sparsely connected. The results in this work will help clarify the effect of network topology on performance and will develop tools that enable designers to compare various strategies against each other and against the centralized solution.

1.6 Notation and Symbols

All vectors are column vectors, with the exception of the regression vector (denoted by the letters u or $\boldsymbol{u}$), which will be taken to be a row vector for convenience of presentation. Table 1.1 lists the main conventions used in our exposition. In particular, note that we use ***boldface*** letters to refer to random quantities and *normal* font to refer to their realizations or deterministic quantities. We also use T for matrix or vector transposition and $*$ for complex-conjugate transposition.

Moreover, for generality, we treat the case in which the variables of interest are generally *complex-valued*; when necessary, we show how the results simplify in the real case. Some subtle differences in the analysis arise when dealing with complex data. These differences would be masked if we focus exclusively on real-valued data. Moreover, studying design problems with complex data is relevant for many fields, especially in the domain of signal processing and communications problems.

Table 1.1: List of notation and symbols used in the text and appendices.

$\mathbb{R}$	Field of real numbers.
$\mathbb{C}$	Field of complex numbers.
$\mathbb{1}$	Column vector with all its entries equal to one.
I_M	Identity matrix of size $M \times M$.
$\boldsymbol{d}$	Boldface notation denotes random variables.
d	Normal font denotes realizations of random variables.
A	Capital letters denote matrices.
a	Small letters denote vectors or scalars.
α	Greek letters denote scalars.
$d(i)$	Small letters with parenthesis denote scalars.
d_i	Small letters with subscripts denote vectors.
T	Matrix transposition.
$*$	Complex-conjugate transposition.
$\mathrm{Re}(z)$	Real part of complex number z.
$\mathrm{Im}(z)$	Imaginary part of complex number z.
$\mathrm{col}\{a, b\}$	Column vector with entries a and b.
$\mathrm{diag}\{a, b\}$	Diagonal matrix with entries a and b.
$\mathrm{vec}\{A\}$	Vector obtained by stacking the columns of A.
$\mathrm{bvec}\{\mathcal{A}\}$	Vector obtained by vectorizing and stacking blocks of $\mathcal{A}$.
$\|x\|$	Euclidean norm of its vector argument.
$\|x\|_\Sigma^2$	Weighted square value $x^*\Sigma x$.
$\|A\|$	Two-induced norm of matrix A, also equal to $\sigma_{\max}(A)$.
$\|A\|_1$	Maximum absolute column sum of matrix A.
$\|A\|_\infty$	Maximum absolute row sum of matrix A.
$A \geq 0$	Matrix A is non-negative definite.
$A > 0$	Matrix A is positive-definite.
$\rho(A)$	Spectral radius of matrix A.
$\lambda_{\max}(A)$	Maximum eigenvalue of the Hermitian matrix A.
$\lambda_{\min}(A)$	Minimum eigenvalue of the Hermitian matrix A.
$\sigma_{\max}(A)$	Maximum singular value of A.
$A \otimes B$	Kronecker product of A and B.
$\mathcal{A} \otimes_b \mathcal{B}$	Block Kronecker product of block matrices $\mathcal{A}$ and $\mathcal{B}$.
$a \preceq b$	Element-wise comparison of the entries of vectors a and b.
$\delta_{k,\ell}$	Kronecker delta sequence: 1 when $k = \ell$ and 0 when $k \neq \ell$.
$\alpha = O(\mu)$	Signifies that $\|\alpha\| \leq c\|\mu\|$ for some constant $c > 0$.
$\alpha = o(\mu)$	Signifies that $\alpha/\mu \to 0$ as $\mu \to 0$.
$\alpha(\mu) \doteq \beta(\mu)$	Signifies that $\alpha(\mu)$ and $\beta(\mu)$ agree to first order in μ.
$\limsup_{n\to\infty} a(n)$	Limit superior of the sequence $a(n)$.
$\liminf_{n\to\infty} a(n)$	Limit inferior of the sequence $a(n)$.

2

Optimization by Single Agents

In this chapter we review the class of gradient-descent algorithms, which are among the most successful iterative techniques for the solution of optimization problems by stand-alone single agents. The presentation summarizes some classical results and provides insights that are useful for our later study of the more demanding scenario of optimization by networked agents. We consider initially the case of real-valued arguments [207] and extend the results to the complex domain as well. We also consider both cases of constant step-sizes and decaying step-sizes.

2.1 Risk and Loss Functions

Thus, let $J(w) \in \mathbb{R}$ denote a real-valued (cost or utility or risk) function of a real-valued vector argument, $w \in \mathbb{R}^M$. It is common in adaptation and learning applications for $J(w)$ to be constructed as the expectation of some loss function, $Q(w; \boldsymbol{x})$, where the ***boldface*** variable $\boldsymbol{x}$ is used to denote some random data, say,

$$J(w) = \mathbb{E}\, Q(w; \boldsymbol{x}) \tag{2.1}$$

and the expectation is evaluated over the distribution of $\boldsymbol{x}$ [207]. Following the notation introduced in Appendices A and B, we denote the gradient vectors of $J(w)$ relative to w and w^{T} by the following row and column vectors, respectively, where the first expression is also referred to as the Jacobian of $J(w)$ relative to w:

$$\nabla_w J(w) \triangleq \left[\frac{\partial J(w)}{\partial w_1} \quad \frac{\partial J(w)}{\partial w_2} \quad \cdots \quad \frac{\partial J(w)}{\partial w_M} \right] \tag{2.2}$$

$$\nabla_{w^{\mathsf{T}}} J(w) \triangleq [\nabla_w J(w)]^{\mathsf{T}} \tag{2.3}$$

These definitions are in terms of the partial derivatives of $J(w)$ relative to the individual entries of w:

$$w \triangleq \mathrm{col}\{w_1, w_2, \ldots, w_M\} \tag{2.4}$$

Likewise, the Hessian matrix of $J(w)$ with respect to w is defined as the following $M \times M$ symmetric matrix:

$$\nabla_w^2 J(w) \triangleq \nabla_{w^{\mathsf{T}}}[\nabla_w J(w)] = \nabla_w[\nabla_{w^{\mathsf{T}}} J(w)] \tag{2.5}$$

which is constructed from two successive gradient operations.

Example 2.1 (Mean-square-error costs)**.** Let $\boldsymbol{d}$ denote a zero-mean scalar random variable with variance $\sigma_d^2 = \mathbb{E}\,\boldsymbol{d}^2$ and let $\boldsymbol{u}$ denote a zero-mean $1 \times M$ random vector with covariance matrix $R_u = \mathbb{E}\,\boldsymbol{u}^{\mathsf{T}}\boldsymbol{u} > 0$. The combined quantities $\{\boldsymbol{d}, \boldsymbol{u}\}$ represent the random variable $\boldsymbol{x}$ referred to in (2.1). The cross-covariance vector is denoted by $r_{du} = \mathbb{E}\,\boldsymbol{d}\boldsymbol{u}^{\mathsf{T}}$. We formulate the problem of estimating $\boldsymbol{d}$ from $\boldsymbol{u}$ in the linear least-mean-squares sense or, equivalently, the problem of seeking the vector w^o that minimizes the quadratic cost function:

$$J(w) \triangleq \mathbb{E}\,(\boldsymbol{d} - \boldsymbol{u}w)^2 = \sigma_d^2 - 2r_{du}^{\mathsf{T}} w + w^{\mathsf{T}} R_u w \tag{2.6}$$

This cost corresponds to the following choice for the loss function:

$$Q(w; \boldsymbol{x}) \triangleq (\boldsymbol{d} - \boldsymbol{u}w)^2 = \boldsymbol{d}^2 - 2\boldsymbol{d}\boldsymbol{u}w + w^{\mathsf{T}}\boldsymbol{u}^{\mathsf{T}}\boldsymbol{u}w \tag{2.7}$$

Such quadratic costs are widely used in estimation and adaptation problems [107, 133, 205, 206, 262]. They are also widely used as quadratic risk functions in machine learning applications [37, 233]. The gradient vector and Hessian matrix of $J(w)$ are easily seen to be:

$$\nabla_w J(w) = 2\,(R_u w - r_{du})^{\mathsf{T}}, \qquad \nabla_w^2 J(w) = 2R_u \tag{2.8}$$

Figure 2.1 illustrates the mean-square-error cost (2.6) for the two-dimensional case, $M = 2$. The individual entries of $w \in \mathbb{R}^M$ are denoted by $w = \text{col}\{w_1, w_2\}$. The plot is generated by using $\sigma_d^2 = 0.5$, a diagonal covariance matrix, R_u, whose entries are generated randomly from within the interval $[1, 10]$, and a cross-covariance vector, r_{du}, whose entries are also generated randomly within the range $[0, 1]$.

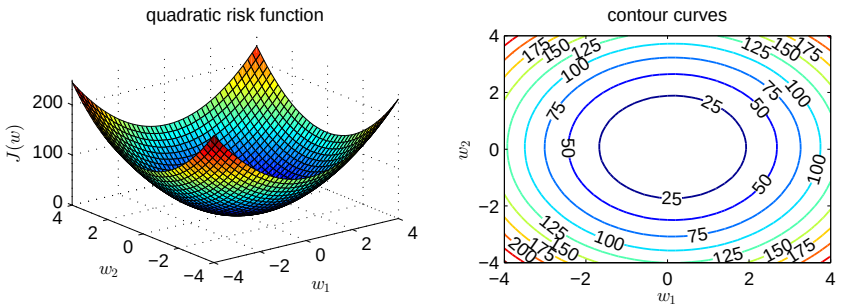

Figure 2.1: Illustration of the mean-square-error cost (2.6) for the two-dimensional case, $M = 2$ (left), along with the corresponding contour curves (right). The plots are generated by using $\sigma_d^2 = 0.5$, and randomly-generated diagonal covariance matrix, R_u, and cross-covariance vector r_{du}.

■

Example 2.2 (Logistic or log-loss risks)**.** Let $\boldsymbol{\gamma}$ denote a binary random variable that assumes the values ± 1, and let $\boldsymbol{h}$ denote an $M \times 1$ random (feature) vector with $R_h = \mathbb{E}\,\boldsymbol{h}\boldsymbol{h}^{\mathsf{T}}$. The combined quantities $\{\boldsymbol{\gamma}, \boldsymbol{h}\}$ represent the random variable $\boldsymbol{x}$ referred to in (2.1). In the context of machine learning and pattern classification problems [37, 115, 233], the variable $\boldsymbol{\gamma}$ designates the class that feature vector $\boldsymbol{h}$ belongs to. In these problems, one seeks the vector w^o that minimizes the regularized logistic risk function — see Appendix G:

$$J(w) \triangleq \frac{\rho}{2}\|w\|^2 + \mathbb{E}\left\{\ln\left(1 + e^{-\boldsymbol{\gamma}\boldsymbol{h}^{\mathsf{T}}w}\right)\right\} \tag{2.9}$$

where $\rho > 0$ is some regularization parameter, $\ln(\cdot)$ is the natural logarithm function, and $\|w\|^2 = w^{\mathsf{T}}w$. The risk (2.9) corresponds to the following choice for the loss function:

$$Q(w; \boldsymbol{x}) \triangleq \frac{\rho}{2}\|w\|^2 + \ln\left(1 + e^{-\boldsymbol{\gamma}\boldsymbol{h}^{\mathsf{T}}w}\right) \tag{2.10}$$

Once w^o is recovered, its value can be used to classify new feature vectors, say, $\{\boldsymbol{h}_\ell\}$, into classes $+1$ or -1. This can be achieved, for example, by assigning feature vectors with $\boldsymbol{h}_\ell^{\mathsf{T}}w^o \geq 0$ to one class and feature vectors with $\boldsymbol{h}_\ell^{\mathsf{T}}w^o < 0$

to another class. Assuming the distribution of $\{\boldsymbol{\gamma}, \boldsymbol{h}\}$ is such that it permits the exchange of the expectation and differentiation operations, it can be verified that for the above $J(w)$:

$$\nabla_w J(w) = \rho w^{\mathsf{T}} - \mathbb{E}\left\{\boldsymbol{\gamma}\boldsymbol{h}^{\mathsf{T}}\left(\frac{e^{-\boldsymbol{\gamma}\boldsymbol{h}^{\mathsf{T}}w}}{1+e^{-\boldsymbol{\gamma}\boldsymbol{h}^{\mathsf{T}}w}}\right)\right\} \tag{2.11}$$

$$\nabla_w^2 J(w) = \rho I_M + \mathbb{E}\left\{\boldsymbol{h}\boldsymbol{h}^{\mathsf{T}}\left(\frac{e^{-\boldsymbol{\gamma}\boldsymbol{h}^{\mathsf{T}}w}}{\left(1+e^{-\boldsymbol{\gamma}\boldsymbol{h}^{\mathsf{T}}w}\right)^2}\right)\right\} \tag{2.12}$$

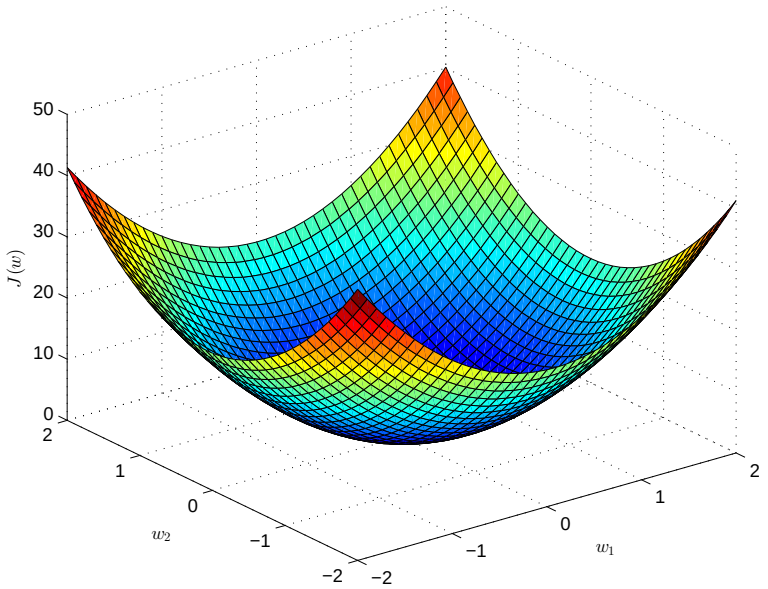

Figure 2.2: Illustration of the logistic risk (2.9) for $M = 2$ and $\rho = 10$. The plot is generated by approximating the expectation in (2.9) by the sample average over 100 repeated realizations for the random variables $\{\boldsymbol{\gamma}, \boldsymbol{h}\}$.

Figure 2.2 illustrates the logistic risk function (2.9) for the two-dimensional case, $M = 2$, and using $\rho = 10$. The individual entries of $w \in \mathbb{R}^2$ are denoted by $w = \mathrm{col}\{w_1, w_2\}$. The plot is generated by approximating the expectation in (2.9) by means of a sample average over 100 repeated realizations for the random variables $\{\boldsymbol{\gamma}, \boldsymbol{h}\}$. Specifically, a total of 100 binary realizations are generated for $\boldsymbol{\gamma}$, where the values ± 1 are assumed with equal probability, and 100 Gaussian realizations are generated for $\boldsymbol{h}$ with mean vectors $+\mathbb{1}$ and $-\mathbb{1}$ for the classes $\boldsymbol{\gamma} = +1$ and $\boldsymbol{\gamma} = -1$, respectively.

■

2.2 Conditions on Risk Function

Stochastic gradient algorithms are powerful iterative procedures for solving optimization problems of the form

$$w^o = \arg\min_{w} \; J(w) \tag{2.13}$$

While the analysis that follows can be pursued under more relaxed conditions (see, e.g., the treatments in [32, 190, 191, 243]), it is sufficient for our purposes to require $J(w)$ to be strongly-convex and twice-differentiable with respect to w. Recall from property (C.18) in the appendix that the cost function $J(w)$ is said to be $\nu-$strongly convex if, and only if, its Hessian matrix is sufficiently bounded away from zero [29, 45, 177, 190]:

$$J(w) \text{ is } \nu-\text{strongly convex} \iff \nabla_w^2 J(w) \geq \nu I_M > 0 \tag{2.14}$$

for all w and for some scalar $\nu > 0$. Strong convexity is a useful condition in the context of adaptation and learning from streaming data because it helps guard against ill-conditioning in the algorithms; it also helps ensure that $J(w)$ has a *unique* global minimum, say, at location w^o; there will be no other minima, maxima, or saddle points. In addition, as we are going to see later in (2.23), it is well-known that strong convexity endows gradient-descent algorithms with geometric (i.e., exponential) convergence rates in the order of $O(\alpha^i)$, for some $0 \leq \alpha < 1$ and where i is the iteration index [32, 190]. For comparison purposes, when the function $J(w)$ is only convex but not necessarily strongly convex, then from the same property (C.18) we know that convexity is equivalent to the following condition:

$$J(w) \text{ is convex} \iff \nabla_w^2 J(w) \geq 0 \tag{2.15}$$

for all w. In this case, while the function $J(w)$ will only have global minima, there can now be multiple global minima. Moreover, the convergence of the gradient-descent algorithm will now occur at the slower rate of $O(1/i)$ [32, 190].

In most problems of interest in adaptation and learning, the cost function $J(w)$ is either already strongly convex or can be made strongly

convex by means of regularization. For example, it is common in machine learning problems [37, 233] and in adaptation and estimation problems [133, 206] to incorporate regularization factors into the cost functions; these factors help ensure strong convexity automatically. For instance, the mean-square-error cost (2.6) is strongly convex whenever $R_u > 0$. If R_u happens to be singular, then the following regularized cost will be strongly convex:

$$J(w) \;\triangleq\; \frac{\rho}{2}\|w\|^2 \;+\; \mathbb{E}\,(\boldsymbol{d} - \boldsymbol{u}w)^2 \tag{2.16}$$

where $\rho > 0$ is a regularization parameter similar to (2.9).

Besides strong convexity, we also require the gradient vector of $J(w)$ to be $\delta-$Lipschitz, namely, that there exists $\delta > 0$ such that

$$\|\nabla_w J(w_2) - \nabla_w J(w_1)\| \leq \delta \, \|w_2 - w_1\| \tag{2.17}$$

for all w_1, w_2. It follows from Lemma E.3 in the appendix that for twice-differentiable costs, conditions (2.14) and (2.17) combined are equivalent to

$$0 < \nu I_M \;\leq\; \nabla_w^2 J(w) \;\leq\; \delta I_M \tag{2.18}$$

For example, it is clear that the Hessian matrices in (2.8) and (2.12) satisfy this property since

$$2\lambda_{\min}(R_u) I_M \;\leq\; \nabla_w^2 J(w) \;\leq\; 2\lambda_{\max}(R_u) I_M \tag{2.19}$$

in the first case and

$$\rho I_M \;\leq\; \nabla_w^2 J(w) \;\leq\; (\rho + \lambda_{\max}(R_h)) I_M \tag{2.20}$$

in the second case. In summary, we will be assuming the following conditions on the cost function.

Assumption 2.1 (Conditions on cost function)**.** The cost function $J(w)$ is twice-differentiable and satisfies (2.18) for some positive parameters $\nu \leq \delta$. Condition (2.18) is equivalent to requiring $J(w)$ to be $\nu-$strongly convex and for its gradient vector to be $\delta-$Lipschitz as in (2.14) and (2.17), respectively.

2.3 Optimization via Gradient Descent

There are many techniques by which optimization problems of the form (2.13) can be solved. We focus in this work on the important class of gradient descent algorithms. These algorithms require knowledge of the actual gradient vector and take the following form:

$$w_i \;=\; w_{i-1} \;-\; \mu\,\nabla_{w^{\mathsf{T}}} J(w_{i-1}), \quad i \geq 0 \tag{2.21}$$

where $i \geq 0$ is an iteration index (usually time), and $\mu > 0$ is a *constant* step-size parameter. The following result establishes that the successive iterates $\{w_i\}$ converge exponentially fast towards w^o for any step-size smaller than the threshold specified by (2.22).

Lemma 2.1 (Convergence with constant step-size: Real case)**.** Assume the cost function, $J(w)$, satisfies Assumption 2.1. If the step-size μ is chosen to satisfy

$$0 < \mu < \frac{2\nu}{\delta^2} \tag{2.22}$$

then, it holds that for any initial condition, w_{-1}, the gradient descent algorithm (2.21) generates iterates $\{w_i\}$ that converge exponentially fast to the global minimizer, w^o, i.e., it holds that

$$\|\widetilde{w}_i\|^2 \;\leq\; \alpha\,\|\widetilde{w}_{i-1}\|^2 \tag{2.23}$$

where the real scalar α satisfies $0 \leq \alpha < 1$ and is given by

$$\alpha \;=\; 1 - 2\mu\nu + \mu^2\delta^2 \tag{2.24}$$

and $\widetilde{w}_i = w^o - w_i$ denotes the error vector at iteration i.

Proof. We provide two arguments. The first derivation is perhaps more traditional, while the second derivation is based on arguments that are more convenient when we extend the results to optimization over networked agents. We start by subtracting w^o from both sides of (2.21) and use the fact that $\nabla_{w^{\mathsf{T}}} J(w^o) = 0$ to write

$$\widetilde{w}_i \;=\; \widetilde{w}_{i-1} \;+\; \mu\,[\nabla_{w^{\mathsf{T}}} J(w_{i-1}) - \nabla_{w^{\mathsf{T}}} J(w^o)] \tag{2.25}$$

Computing the squared Euclidean norms (or energies) of both sides of the above equality gives

$$
\begin{aligned}
\|\widetilde{w}_i\|^2 &= \|\widetilde{w}_{i-1}\|^2 + \mu^2 \|\nabla_{w^{\mathsf{T}}} J(w_{i-1}) - \nabla_{w^{\mathsf{T}}} J(w^o)\|^2 + \\
&\qquad 2\mu \left[\nabla_w J(w_{i-1}) - \nabla_w J(w^o)\right] \widetilde{w}_{i-1} \\
&\overset{(a)}{\leq} \|\widetilde{w}_{i-1}\|^2 + \mu^2 \left\| \left(\int_0^1 \nabla_w^2 J(w^o - t\widetilde{w}_{i-1}) dt \right) \widetilde{w}_{i-1} \right\|^2 - 2\mu\nu \|\widetilde{w}_{i-1}\|^2 \\
&\overset{(b)}{\leq} \|\widetilde{w}_{i-1}\|^2 + \mu^2 \delta^2 \|\widetilde{w}_{i-1}\|^2 - 2\mu\nu \|\widetilde{w}_{i-1}\|^2 \\
&= \alpha \|\widetilde{w}_{i-1}\|^2
\end{aligned} \tag{2.26}
$$

where step (a) uses the mean-value relation (D.9) and the strong-convexity property (C.17) from the appendices, while step (b) uses the upper bound in (2.18) on the Hessian matrix.

We next verify that condition (2.22) ensures $0 \leq \alpha < 1$. For this purpose, we refer to Figure 2.3, which plots the coefficient $\alpha(\mu)$ as a function of μ. The minimum value of $\alpha(\mu)$, which occurs at the location $\mu = \nu/\delta^2$ and is equal to $1 - \nu^2/\delta^2$, is nonnegative since $0 < \nu \leq \delta$. It is now clear from the figure that $0 \leq \alpha < 1$ for $\mu \in (0, \frac{2\nu}{\delta^2})$.

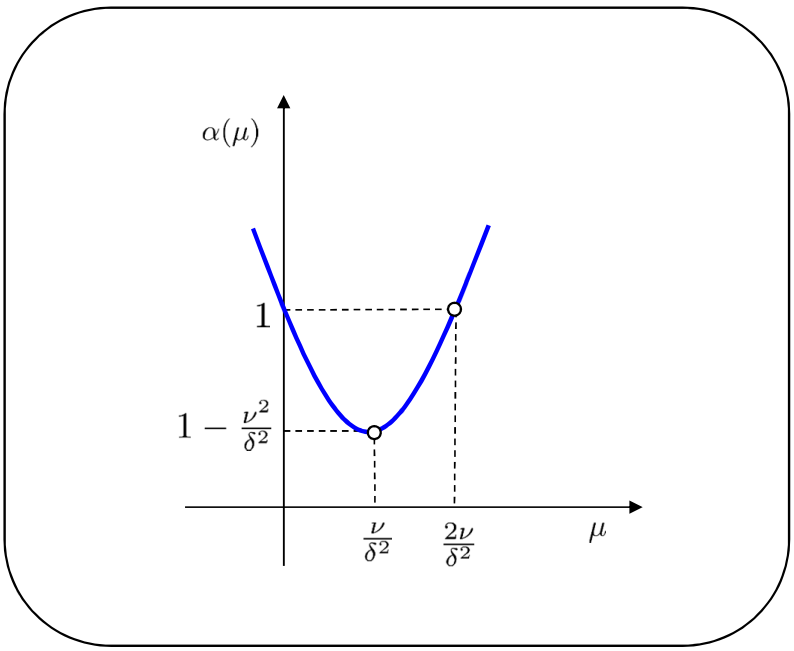

Figure 2.3: Plot of the function $\alpha(\mu) = 1 - 2\nu\mu + \mu^2\delta^2$ given by (2.24). It shows that the function $\alpha(\mu)$ assumes values below one in the range $0 < \mu < 2\nu/\delta^2$.

Alternative proof. We can arrive at the same conclusion by using an alternative argument, which may seem to be more demanding at first sight. However, it turns out to be more convenient for scenarios involving optimization by networked agents, as we are going to study in future chapters — see, e.g., the derivation in Sec. 8.4.

We again subtract w^o from both sides of (2.21) to get

$$\widetilde{w}_i \;=\; \widetilde{w}_{i-1} \;+\; \mu\,\nabla_{w^{\mathsf{T}}} J(w_{i-1}) \tag{2.27}$$

We then appeal to the mean-value relation (D.9) from the appendix to note that

$$\begin{aligned}\nabla_{w^{\mathsf{T}}} J(w_{i-1}) \;&=\; -\left(\int_0^1 \nabla_w^2\, J(w^o - t\widetilde{w}_{i-1})dt\right)\widetilde{w}_{i-1} \\ &\triangleq\; -H_{i-1}\widetilde{w}_{i-1}\end{aligned} \tag{2.28}$$

where we are introducing the *symmetric* time-variant matrix H_{i-1}, which is defined in terms of the Hessian of the cost function:

$$H_{i-1} \;\triangleq\; \int_0^1 \nabla_w^2\, J(w^o - t\widetilde{w}_{i-1})dt \tag{2.29}$$

Substituting (2.28) into (2.27), we get the alternative representation:

$$\widetilde{w}_i \;=\; (I_M - \mu H_{i-1})\widetilde{w}_{i-1} \tag{2.30}$$

Note that the matrix H_{i-1} depends on $\widetilde{w}_{i-1}$ so that the right-hand side of the above recursion actually depends on $\widetilde{w}_{i-1}$ in a nonlinear fashion. However, we can still determine a condition on μ for convergence of $\widetilde{w}_i$ to zero because we can determine a uniform bound on H_{i-1} as follows [190]. Using the sub-multiplicative property of norms, we have

$$\|\widetilde{w}_i\|^2 \;\le\; \|I_M - \mu H_{i-1}\|^2 \cdot \|\widetilde{w}_{i-1}\|^2 \tag{2.31}$$

But since $J(w)$ satisfies (2.18), we know that

$$(1-\mu\delta)I_M \;\le\; I_M - \mu H_{i-1} \;\le\; (1-\mu\nu)I_M \tag{2.32}$$

for all i. Using the fact that $I_M - \mu H_{i-1}$ is a symmetric matrix, we have that its 2−induced norm is equal to its spectral radius so that

$$\begin{aligned}\|I_M - \mu H_{i-1}\|^2 \;&=\; [\rho(I_M - \mu H_{i-1})]^2 \\ &\overset{(2.32)}{\le}\; \max\{(1-\mu\delta)^2,\; (1-\mu\nu)^2\} \\ &=\; \max\left\{1 - 2\mu\delta + \mu^2\delta^2,\; 1 - 2\mu\nu + \mu^2\nu^2\right\} \\ &\overset{(a)}{\le}\; 1 - 2\mu\nu + \mu^2\delta^2 \\ &=\; \alpha\end{aligned} \tag{2.33}$$

where we used the fact that $\delta \geq \nu$ in step (a). Combining this result with (2.31) we again conclude that (2.23) holds and, therefore, condition (2.22) on the step-size ensures $\widetilde{w}_i \to 0$ as $i \to \infty$.

Actually, the argument that led to (2.33) can be refined to conclude that convergence of $\widetilde{w}_i$ to zero occurs over the wider interval

$$\mu < 2/\delta \tag{2.34}$$

than (2.22). This is because condition (2.34) already ensures

$$\max\{(1-\mu\delta)^2,\ (1-\mu\nu)^2\} < 1 \tag{2.35}$$

We will continue with condition (2.22); it is sufficient for our purposes to know that a small enough step-size value exists that ensures convergence.

□

Example 2.3 (Optimization of mean-square-error costs)**.** Let us reconsider the quadratic cost (2.6) from Example 2.1. We know from (2.19) that $\delta = 2\lambda_{\max}(R_u)$ and $\nu = 2\lambda_{\min}(R_u)$. Furthermore, if we set the gradient vector in (2.8) to zero, we conclude that the minimizer, w^o, is given by the unique solution to the equations $R_u w^o = r_{du}$. We can alternatively determine this same minimizer in an iterative manner by using the gradient descent recursion (2.21). Indeed, if we substitute expression (2.8) for the gradient vector into (2.21), we find that the iterative algorithm reduces to

$$w_i \ = \ w_{i-1} \ + \ 2\mu\,(r_{du} - R_u w_{i-1}), \quad i \geq 0 \tag{2.36}$$

We know from condition (2.22) that the iterates $\{w_i\}$ generated by this recursion will converge to w^o at an exponential rate for any step-size $\mu < \lambda_{\min}(R_u)/\lambda^2_{\max}(R_u)$. Using condition (2.34) instead, we actually have that convergence of w_i to w^o is guaranteed over the wider range of step-size values $\mu < 1/\lambda_{\max}(R_u)$. This conclusion can also be seen from the fact that, in this case, the matrix H_{i-1} defined by (2.29) is constant and equal to $2R_u$ (i.e., it is independent of $\widetilde{w}_{i-1}$). In this way, recursion (2.30) becomes

$$\widetilde{w}_i \ = \ (I_M - 2\mu R_u)\widetilde{w}_{i-1}, \quad i \geq 0 \tag{2.37}$$

from which it is again clear that $\widetilde{w}_i$ converges to zero for all $\mu < 1/\lambda_{\max}(R_u)$.

■

2.4 Decaying Step-Size Sequences

It is also possible to employ in (2.21) iteration-dependent step-size sequences, $\mu(i) \geq 0$, instead of the constant step-size μ, and to require

$\mu(i)$ to satisfy the two conditions:

$$\sum_{i=0}^{\infty} \mu(i) = \infty, \quad \lim_{i\to\infty} \mu(i) = 0 \tag{2.38}$$

For example, sequences of the form

$$\mu(i) \;=\; \frac{\tau}{i+1}, \quad i \geq 0 \tag{2.39}$$

satisfy conditions (2.38) for any finite positive constant τ. It is well-known that, under (2.38), the gradient descent recursion, namely,

$$w_i \;=\; w_{i-1} \;-\; \mu(i)\,\nabla_{w^{\mathsf{T}}} J(w_{i-1}), \;\; i \geq 0 \tag{2.40}$$

continues to ensure the convergence of w_i towards w^o, as explained next [32, 190, 243]. However, the convergence rate will now be slower and in the order of $O(1/i^{2\nu\tau})$. That is, the convergence rate will not be geometric (or exponential) any longer. For this reason, the constant step-size implementation is preferred. Nevertheless, we will still discuss the decaying step-size case in order to prepare for our future treatment of stochastic gradient algorithms where such step-sizes are more relevant. A second issue with the use of decaying step-sizes is that conditions (2.38) force the step-size sequence to decay to zero; this feature is problematic for scenarios requiring continuous adaptation and learning from streaming data (which will be the main focus of our treatment starting from the next chapter). This is because, in many instances, it is not unusual for the location of the minimizer, w^o, to drift with time. With $\mu(i)$ decaying towards zero, the gradient descent algorithm (2.40) will stop updating and will not be able to track drifts in the solution.

Lemma 2.2 (Convergence with decaying step-size sequence: Real case)**.** Assume the cost function, $J(w)$, satisfies Assumption 2.1. If the step-size sequence $\mu(i)$ satisfies the two conditions in (2.38), then it holds that for any initial condition, w_{-1}, the gradient descent algorithm (2.40) generates iterates $\{w_i\}$ that converge to the global minimizer, w^o. Moreover, when the step-size sequence is chosen as in (2.39), then the convergence rate is in the order of $\|\widetilde{w}_i\|^2 = O(1/i^{2\nu\tau})$ for large enough i.

Proof. We first establish the convergence result for step-size sequences satisfying (2.38). The argument that led to (2.26) will similarly lead to

$$\|\widetilde{w}_i\|^2 \;\leq\; \alpha(i)\,\|\widetilde{w}_{i-1}\|^2 \tag{2.41}$$

where now $\alpha(i) = 1 - 2\nu\mu(i) + \delta^2\mu^2(i)$. We split $2\nu\mu(i)$ into the sum of two factors and write

$$\alpha(i) \;=\; 1 - \nu\mu(i) - \nu\mu(i) + \delta^2\mu^2(i) \tag{2.42}$$

Now, since $\mu(i) \to 0$, we conclude that for large enough $i > i_o$, the sequence $\mu^2(i)$ will assume smaller values than $\mu(i)$. Therefore, a large enough time index, i_o, exists such that the following two conditions are satisfied:

$$\nu\mu(i) \geq \delta^2\mu^2(i), \qquad 0 < 1 - \nu\mu(i) \leq 1, \qquad i > i_o \tag{2.43}$$

It follows that

$$\alpha(i) \;\leq\; 1 - \nu\mu(i), \qquad i > i_o \tag{2.44}$$

and, hence,

$$\|\widetilde{w}_i\|^2 \;\leq\; (1 - \nu\mu(i))\;\|\widetilde{w}_{i-1}\|^2, \qquad i > i_o \tag{2.45}$$

Iterating over i we can write (assuming a finite i_o exists for which $\|\tilde{w}_{i_o}\| \neq 0$, otherwise the algorithm would have converged) [164, 205]:

$$\lim_{i\to\infty}\left(\frac{\|\widetilde{w}_i\|^2}{\|\widetilde{w}_{i_o}\|^2}\right) \leq \prod_{i=i_o+1}^{\infty}(1 - \nu\mu(i)) \tag{2.46}$$

or, equivalently,

$$\lim_{i\to\infty}\ln\left(\frac{\|\widetilde{w}_i\|^2}{\|\widetilde{w}_{i_o}\|^2}\right) \leq \sum_{i=i_o+1}^{\infty}\ln\left(1 - \nu\mu(i)\right) \tag{2.47}$$

Now using the following easily verified property for the natural logarithm function:

$$\ln(1-y) \;\leq\; -y, \qquad \text{for all } 0 \leq y < 1 \tag{2.48}$$

and letting $y = \nu\mu(i)$, we have that

$$\ln(1 - \nu\mu(i)) \;\leq\; -\nu\mu(i), \qquad i > i_o \tag{2.49}$$

so that

$$\sum_{i=i_o+1}^{\infty}\ln(1-\nu\mu(i)) \;\leq\; -\sum_{i=i_o+1}^{\infty}\nu\mu(i) \;=\; -\nu\left(\sum_{i=i_o+1}^{\infty}\mu(i)\right) \;=\; -\infty \tag{2.50}$$

since the step-size series is assumed to be divergent in (2.38). We conclude that

$$\lim_{i\to\infty} \ln\left(\frac{\|\widetilde{w}_i\|^2}{\|\widetilde{w}_{i_o}\|^2}\right) = -\infty \tag{2.51}$$

so that $\widetilde{w}_i \to 0$ as $i \to \infty$.

We now examine the convergence rate for step-size sequences of the form (2.39). Note first that these sequences satisfy the following two conditions

$$\sum_{i=0}^{\infty} \mu(i) = \infty, \qquad \sum_{i=0}^{\infty} \mu^2(i) \;=\; \tau^2\left(\sum_{i=1}^{\infty} \frac{1}{i^2}\right) \;=\; \frac{\tau^2\pi^2}{6} < \infty \tag{2.52}$$

Again, since $\mu(i) \to 0$ and $\mu^2(i)$ decays faster than $\mu(i)$, we know that for some large enough $i > i_1$, it will hold that

$$2\nu\mu(i) \geq \delta^2\mu^2(i) \tag{2.53}$$

and, hence,

$$0 < \alpha(i) \leq 1, \quad i > i_1 \tag{2.54}$$

We can now repeat the same steps up to (2.51) using $y = 2\nu\mu(i') - \delta^2\mu^2(i')$ to conclude that

$$\begin{aligned}
\ln\left(\frac{\|\widetilde{w}_i\|^2}{\|\widetilde{w}_{i_1}\|^2}\right) &\leq \sum_{i'=i_1+1}^{i} \ln\left(1 - 2\nu\mu(i') + \delta^2\mu^2(i')\right) \\
&\leq -\sum_{i'=i_1+1}^{i} \left[2\nu\mu(i') - \delta^2\mu^2(i')\right] \\
&= -2\nu\left(\sum_{i'=i_1+1}^{i} \mu(i')\right) + \delta^2\left(\sum_{i'=i_1+1}^{i} \mu^2(i')\right) \\
&\leq -2\nu\left(\sum_{i'=i_1+1}^{i} \mu(i')\right) + \frac{\delta^2\tau^2\pi^2}{6} \\
&= -2\nu\tau\left(\sum_{i'=i_1+2}^{i+1} \frac{1}{i'}\right) + \frac{\delta^2\tau^2\pi^2}{6} \\
&\overset{(a)}{\leq} -2\nu\tau\left(\int_{i_1+2}^{i+2} \frac{1}{x}dx\right) + \frac{\delta^2\tau^2\pi^2}{6} \\
&= 2\nu\tau\ln\left(\frac{i_1+2}{i+2}\right) + \frac{\delta^2\tau^2\pi^2}{6} \\
&= \ln\left(\frac{i_1+2}{i+2}\right)^{2\nu\tau} + \frac{\delta^2\tau^2\pi^2}{6}
\end{aligned} \tag{2.55}$$

where in step (a) we used the following integral bound, which reflects the fact that the area under the curve $f(x) = 1/x$ over the interval $x \in [i_1+2, i+2]$ is upper bounded by the sum of the areas of the rectangles shown in Figure 2.4:

$$\int_{i_1+2}^{i+2} \frac{1}{x} dx \;\leq\; \sum_{i'=i_1+2}^{i+1} \frac{1}{i'} \tag{2.56}$$

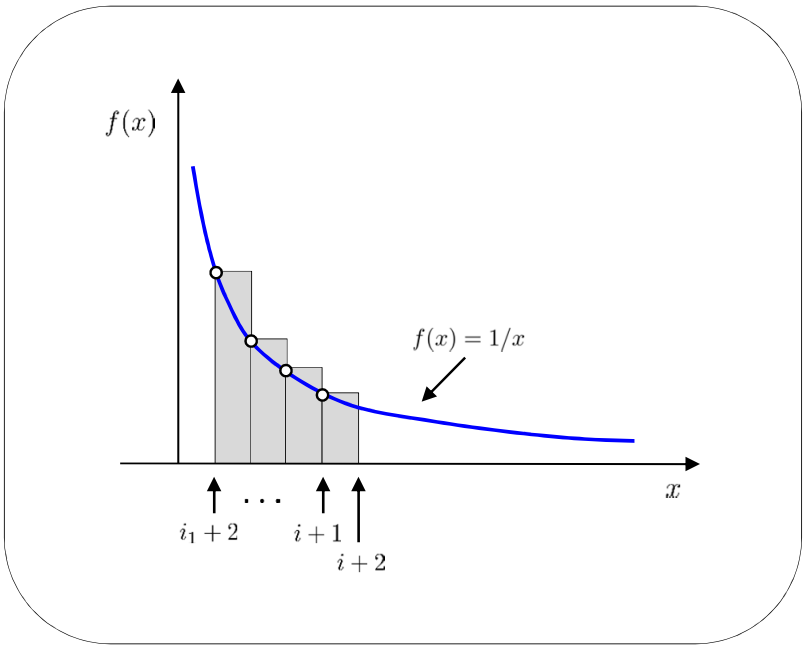

Figure 2.4: The area under the curve $f(x) = 1/x$ over the interval $x \in [i_1 + 2, i + 2]$ is upper bounded by the sum of the areas of the rectangles shown in the figure.

We therefore conclude from (2.55) that

$$\begin{aligned} \|\widetilde{w}_i\|^2 &\leq \left(e^{\left\{ \ln\left(\frac{i_1+2}{i+2}\right)^{2\nu\tau} + \frac{\delta^2\tau^2\pi^2}{6} \right\}} \right) \|\widetilde{w}_{i_1}\|^2, \quad i > i_1 \\ &= e^{\frac{\delta^2\tau^2\pi^2}{6}} \cdot \|\widetilde{w}_{i_1}\|^2 \cdot \left(\frac{i_1+2}{i+2} \right)^{2\nu\tau} \\ &= O(1/i^{2\nu\tau}) \end{aligned} \tag{2.57}$$

as claimed.

□

2.5 Optimization in the Complex Domain

We now extend the results of the previous two sections to the case in which the argument $w \in \mathbb{C}^M$ is complex-valued while $J(w) \in \mathbb{R}$ continues to be real-valued. We again focus on the case of strongly-convex functions, $J(w)$, for which the minimizer, w^o, is unique. It is explained in (C.44) in the appendix that, in the complex case, condition (2.14) is replaced by

$$J(w) \text{ is } \nu-\text{strongly convex} \iff \nabla_w^2 J(w) \geq \frac{\nu}{2} I_{2M} > 0 \tag{2.58}$$

with a factor of $\frac{1}{2}$ multiplying ν, and with I_M replaced by I_{2M} since the Hessian matrix is now $2M \times 2M$. Note that we can capture conditions (2.14) and (2.58) simultaneously in a single statement for both cases of real or complex-valued arguments by writing

$$J(w) \text{ is } \nu-\text{strongly convex} \iff \nabla_w^2 J(w) \geq \frac{\nu}{h} I_{hM} > 0 \tag{2.59}$$

where the variable h is an integer that denotes the type of the data:

$$h \triangleq \begin{cases} 1, & \text{when } w \text{ is real} \\ 2, & \text{when } w \text{ is complex} \end{cases} \tag{2.60}$$

Observe that h appears in two locations in (2.59); in the denominator of ν and in the subscript indicating the size of the identity matrix. We shall frequently employ the data-type variable, h, throughout our presentation, and especially in future chapters, in order to permit a uniform treatment of the various algorithms regardless of the type of the data.

Likewise, the Lipschitz condition (2.17) is replaced by

$$\|\nabla_w J(w_2) - \nabla_w J(w_1)\| \leq \frac{\delta}{h} \|w_2 - w_1\| \tag{2.61}$$

for all w_1, w_2, where again a factor of $h = 2$ would appear on the right-hand-side in the complex case. It follows from the result of Lemma E.7 in the appendix that for twice-differentiable costs, conditions (2.59) and (2.61) combined are equivalent to

$$0 < \frac{\nu}{h} I_{hM} \leq \nabla_w^2 J(w) \leq \frac{\delta}{h} I_{hM} \tag{2.62}$$

We therefore assume that the cost function $J(w)$ is twice-differentiable and satisfies (2.62) for some positive parameters $\nu \leq \delta$.

Example 2.4 (Complex Hessian matrices)**.** Let us reconsider the context of Example 2.1 with complex data. Let $\boldsymbol{d}$ be a scalar zero-mean random variable with variance $\sigma_d^2 = \mathbb{E}\,|\boldsymbol{d}|^2$ and let $\boldsymbol{u}$ be a $1 \times M$ zero-mean random vector with covariance matrix $R_u = \mathbb{E}\,\boldsymbol{u}^*\boldsymbol{u} > 0$. The cross-correlation vector between $\boldsymbol{d}$ and $\boldsymbol{u}$ is denoted by $r_{du} = \mathbb{E}\,\boldsymbol{d}\boldsymbol{u}^*$. The mean-square-error cost function is now defined as

$$\begin{aligned} J_k(w) &\triangleq \mathbb{E}\,|\boldsymbol{d} - \boldsymbol{u}w|^2 \\ &= \sigma_d^2 \;-\; (r_{du})^* w \;-\; w^* r_{du} \;+\; w^* R_u w \end{aligned} \tag{2.63}$$

The complex gradient vectors of $J(w)$ relative to w and w^* are given by — see Example A.3 in the appendix:

$$\nabla_w\, J(w) \;=\; (R_u w - r_{du})^*, \quad \nabla_{w^*} J(w) \;=\; R_u w - r_{du} \tag{2.64}$$

and the $2M \times 2M$ Hessian matrix of $J(w)$ can be verified to be block diagonal in this case and given by — see (B.36) in the appendix:

$$\nabla_w^2\, J(w) = \begin{bmatrix} R_u & 0 \\ 0 & R_u^{\mathsf{T}} \end{bmatrix} \tag{2.65}$$

It is instructive to compare expressions (2.64) and (2.65) with (2.8) in the real case.

■

In the complex case, the gradient descent algorithm (2.21) is replaced by

$$w_i \;=\; w_{i-1} \;-\; \mu\,\nabla_{w^*} J(w_{i-1}), \quad i \geq 0 \tag{2.66}$$

in terms of the complex gradient vector relative to w^*. Since $J(w)$ is real-valued, it holds that

$$\nabla_{w^*} J(w_{i-1}) \;=\; [\nabla_w J(w_{i-1})]^* \tag{2.67}$$

Comparing with (2.21) we see that transposition of the gradient vector is replaced by complex conjugation. The above recursion can be motivated from (2.21) as follows. We express the complex variable w in terms of its real and imaginary components as

$$w = x + jy \tag{2.68}$$

We then treat $J(w)$ as the function $J(v)$ of the $2M \times 1$ extended real variable:

$$v = \text{col}\{x, y\} \tag{2.69}$$

and consider instead the equivalent optimization problem

$$\min_{v \in \mathbb{R}^{2M}} \; J(v) \tag{2.70}$$

We already know from (2.21) that the gradient descent recursion for minimizing $J(v)$ over v, using the step-size $\mu' = \mu/2$, has the form:

$$v_i \;=\; v_{i-1} \;-\; \frac{1}{2}\mu \, \nabla_{v^{\mathsf{T}}} J(v_{i-1}), \quad i \geq 0 \tag{2.71}$$

The reason for introducing the factor of $\frac{1}{2}$ into μ' will become clear soon. We can rewrite the above recursion in terms of the components of $v_i = \text{col}\{x_i, y_i\}$ as follows:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} \;=\; \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix} \;-\; \frac{1}{2}\mu \begin{bmatrix} \nabla_{x^{\mathsf{T}}} J(x_{i-1}, y_{i-1}) \\ \nabla_{y^{\mathsf{T}}} J(x_{i-1}, y_{i-1}) \end{bmatrix} \tag{2.72}$$

where we used relation (C.29) from the appendix to express the gradient vector of $J(v)$ in terms of the gradients of the same function $J(x, y)$ relative to x and y. Now, if we multiply the second block row of (2.72) by jI_M, add both block rows, and use $w_i = x_i + jy_i$, we can rewrite (2.72) in terms of the complex variables $\{w_i, w_{i-1}\}$:

$$\begin{aligned} w_i \;&=\; w_{i-1} \;-\; \frac{1}{2}\mu \left[\nabla_{x^{\mathsf{T}}} J(x_{i-1}, y_{i-1}) \;+\; j \, \nabla_{y^{\mathsf{T}}} J(x_{i-1}, y_{i-1})\right] \\ &\overset{(C.31)}{=} \; w_{i-1} \;-\; \mu \, \nabla_{w^*} J(w_{i-1}) \end{aligned} \tag{2.73}$$

The second relation above agrees with the claimed form (2.66); it is seen that the factor of $1/2$ is used in transforming the combination of gradient vectors relative to x and y into the gradient vector relative to w. The next statement establishes the convergence of (2.66); in the statement, we employ the data-type variable, h, so that the conclusion encompasses both the real and complex-valued domains.

Lemma 2.3 (Convergence with constant step-size: Complex case)**.** Assume the cost function $J(w)$ satisfies (2.62). If the step-size μ is chosen to satisfy

$$\frac{\mu}{h} < \frac{2\nu}{\delta^2} \tag{2.74}$$

then, it holds that for any initial condition, w_{-1}, the gradient descent algorithm (2.66) generates iterates that converge exponentially fast to the global minimizer, w^o, i.e., it holds that

$$\|\widetilde{w}_i\|^2 \;\leq\; \alpha^i \, \|\widetilde{w}_{i-1}\|^2 \tag{2.75}$$

where the real scalar α satisfies $0 \leq \alpha < 1$ and is given by

$$\alpha \;=\; 1 - 2\nu \left(\frac{\mu}{h}\right) \;+\; \delta^2 \left(\frac{\mu}{h}\right)^2 \tag{2.76}$$

Proof. We are only interested in establishing the above results in the complex case, which corresponds to $h = 2$, since we already established these same conclusions for the real case in Lemma 2.1. Rather than establish the claims by working directly with recursion (2.66) in the complex domain, we instead reduce the problem to one that deals with the equivalent function $J(v)$ of the extended *real* variable $v = \mathrm{col}\{x, y\}$ and then apply the result of Lemma 2.1.

To begin with, we already know from (E.39) in the appendix that if $J(w)$ is $\nu-$strongly convex, then $J(v)$ is $\nu-$strongly convex as well. We also know from (E.22) and (E.56) in the same appendix that the gradient vector function of $J(v)$ is Lipschitz with factor δ when the gradient vector function of $J(w)$ is Lipschitz with factor $\delta/2$. We further know from (2.71)–(2.73) that a gradient descent recursion in the $w-$domain (as in (2.73)) is equivalent to a gradient descent recursion in the $v-$domain (as in (2.71)) if we use $\mu' = \mu/2$:

$$v_i \;=\; v_{i-1} \;-\; \mu' \nabla_{v^{\mathsf{T}}} J(v_{i-1}), \quad i \geq 0 \tag{2.77}$$

Lemma 2.1 then guarantees that the real-valued iterates $\{v_i\}$ will converge to v^o when $\mu' < 2\nu/\delta^2$. Consequently, the gradient descent algorithm (2.66) will converge for $\mu < 4\nu/\delta^2$, which is condition (2.74) with $h = 2$ in the complex case. We note that from the argument that led to (2.34) we can conclude that convergence actually occurs over the wider interval $\mu' < 2/\delta$ or, equivalently, $\mu/h < 2/\delta$. Either way, we find that relation (2.75) holds by noting that $\|\widetilde{w}_i\|^2 = \|\widetilde{v}_i\|^2$ and using the result from Lemma 2.1 to conclude that

$$\|\widetilde{v}_i\|^2 \;\leq\; \alpha^i \, \|\widetilde{v}_{i-1}\|^2 \tag{2.78}$$

where $\widetilde{v}_i = v^o - v_i$ and

$$\alpha = 1 - 2\mu'\nu + (\mu')^2\delta^2 \tag{2.79}$$

□

We can also study gradient descent recursions with decaying step-size sequences satisfying (2.38), namely,

$$w_i \;=\; w_{i-1} \;-\; \mu(i)\,\nabla_{w^*} J(w_{i-1}), \quad i \geq 0 \tag{2.80}$$

Lemma 2.4 (Convergence with decaying step-size: Complex case)**.** Assume the cost function $J(w)$ satisfies (2.62). If the step-size sequence $\mu(i)$ satisfies (2.38), then it holds that for any initial condition, w_{-1}, the gradient descent algorithm (2.80) generates iterates $\{w_i\}$ that converge to the global minimizer, w^o. Moreover, when the step-size sequence is chosen as in (2.39), then the convergence rate is in the order of $\|\widetilde{w}_i\|^2 = O(1/i^{(2\nu\tau/h)})$ for large enough i.

Proof. We apply Lemma 2.2 to the following recursion in the $v-$domain:

$$v_i \;=\; v_{i-1} \;-\; \mu'(i)\,\nabla_{v^\mathsf{T}} J(v_{i-1}), \quad i \geq 0 \tag{2.81}$$

where $\mu'(i) = \mu(i)/2$.

□

3

Stochastic Optimization by Single Agents

The gradient descent algorithm (2.21) of the previous chapter requires knowledge of the exact gradient vector of the cost function that is being minimized. In the context of adaptation and learning, this information is rarely available beforehand and needs to be approximated. This step is generally achieved by replacing the true gradient by an approximate gradient, thus leading to *stochastic* gradient algorithms. Important challenges and new features arise when the gradient vector is approximated. For instance, the gradient error that is caused by the approximation (and which we shall call *gradient noise*) ends up interfering with the operation of the algorithm. It therefore becomes important to assess how much degradation in performance occurs. At the same time, the stochastic approximation step infuses a powerful tracking mechanism into the operation of the gradient descent algorithm; it becomes able to track drifts in the location of the minimizer due to changes in the underlying signal statistics or models. This is because stochastic gradient implementations approximate the gradient vector from streaming data. By doing so, and by relaying on actual data realizations, the drifts in the signal models become reflected in the data and they influence the operation of the algorithm in real-time.

3.1 Adaptation and Learning

In order to illustrate the main concepts in these introductory chapters, we treat again the real case first and subsequently extend the results to the complex domain.

Thus, let $J(w) \in \mathbb{R}$ denote the real-valued cost function of a real-valued vector argument, $w \in \mathbb{R}^M$ and consider the same optimization problem (3.1):

$$w^o = \arg\min_{w} \; J(w) \tag{3.1}$$

We continue to assume that $J(w)$ is twice-differentiable and satisfies (2.18) for some positive parameters $\nu \leq \delta$, namely,

$$0 < \nu I_M \;\leq\; \nabla_w^2 J(w) \;\leq\; \delta I_M \tag{3.2}$$

Assumption 3.1 (Conditions on cost function)**.** The cost function $J(w)$ is twice-differentiable and satisfies (3.2) for some positive parameters $\nu \leq \delta$. Condition (3.2) is equivalent to requiring $J(w)$ to be $\nu-$strongly convex and for its gradient vector to be $\delta-$Lipschitz as in (2.14) and (2.17), respectively.

We mentioned in the previous chapter that it is common in adaptation and learning applications for the risk function $J(w)$ to be constructed as the expectation of some loss function, $Q(w;\boldsymbol{x})$, say,

$$J(w) = \mathbb{E}\, Q(w;\boldsymbol{x}) \tag{3.3}$$

where the expectation is evaluated over the distribution of $\boldsymbol{x}$. The traditional gradient-descent algorithm for solving (3.1) was described earlier by (2.21), and we repeat it below for ease of reference:

$$w_i \;=\; w_{i-1} \;-\; \mu \nabla_{w^{\mathsf{T}}} J(w_{i-1}), \quad i \geq 0 \tag{3.4}$$

where $i \geq 0$ is an iteration index and $\mu > 0$ is a small step-size parameter. In order to run this recursion, we need to have access to the true gradient vector, $\nabla_{w^{\mathsf{T}}} J(w_{i-1})$. This information is generally unavailable in most instances involving learning from data. For example, when cost functions are defined as the expectations of certain loss functions

as in (3.3), the statistical distribution of the data $\boldsymbol{x}$ may not be known beforehand. In that case, the exact form of $J(w)$ will not be known since the expectation of $Q(w;\boldsymbol{x})$ cannot be computed. In such situations, it is necessary to replace the true gradient vector, $\nabla_{w^\mathsf{T}}J(w_{i-1})$, by an instantaneous approximation for it, and which we shall denote by $\widehat{\nabla_{w^\mathsf{T}}J}(\boldsymbol{w}_{i-1})$. Doing so leads to the following *stochastic-gradient* recursion in lieu of (3.4):

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \mu\,\widehat{\nabla_{w^\mathsf{T}}J}(\boldsymbol{w}_{i-1}), \quad i \geq 0 \tag{3.5}$$

Note that we are using the ***boldface*** notation, $\boldsymbol{w}_i$, for the iterates in (3.5) to highlight the fact that these iterates are randomly perturbed versions of the values $\{w_i\}$ generated by the original recursion (3.4). The random perturbations arise from the use of the approximate gradient vector; different data realizations lead to different realizations for the approximate gradients. The boldface notation is therefore meant to emphasize the random nature of the iterates in (3.5).

Stochastic gradient algorithms are among the most successful iterative techniques for the solution of adaptation and learning problems by stand-alone single agents [190, 207, 243]. We will be using the term "*learning*" to refer broadly to the ability of an agent to extract information about some unknown parameter from streaming data, such as estimating the parameter itself or learning about some of its features. We will be using the term "*adaptation*" to refer broadly to the ability of the learning algorithm to track drifts in the parameter. The two attributes of learning and adaptation will be embedded simultaneously into the algorithms discussed in this work. We will also be using the term "*streaming data*" regularly because we are interested in algorithms that perform continuous learning and adaptation and that, therefore, are able to improve their performance in response to continuous streams of data arriving at the agent. This is in contrast to off-line algorithms, where the data are first aggregated before being processed for extraction of information.

We illustrate construction (3.5) by considering a scenario from classical adaptive filter theory [107, 206, 262], where the gradient vector is approximated directly from data realizations. The construction will reveal why stochastic-gradient implementations of the form (3.5), us-

ing approximate rather than exact gradient information, are naturally endowed with the ability to respond to *streaming* data.

Example 3.1 (LMS adaptation)**.** Let $\boldsymbol{d}(i)$ denote a streaming sequence of zero-mean random variables with variance $\sigma_d^2 = \mathbb{E}\,\boldsymbol{d}^2(i)$. Let $\boldsymbol{u}_i$ denote a streaming sequence of $1 \times M$ independent zero-mean random vectors with covariance matrix $R_u = \mathbb{E}\,\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i > 0$. Both processes $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ are assumed to be jointly wide-sense stationary. The cross-covariance vector between $\boldsymbol{d}(i)$ and $\boldsymbol{u}_i$ is denoted by $r_{du} = \mathbb{E}\,\boldsymbol{d}(i)\boldsymbol{u}_i^{\mathsf{T}}$. The data $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ are assumed to be related via a linear regression model of the form:

$$\boldsymbol{d}(i) \;=\; \boldsymbol{u}_i w^o \;+\; \boldsymbol{v}(i) \tag{3.6}$$

for some unknown parameter vector w^o, and where $\boldsymbol{v}(i)$ is a zero-mean white-noise process with power $\sigma_v^2 = \mathbb{E}\,\boldsymbol{v}^2(i)$ and assumed independent of $\boldsymbol{u}_j$ for all i, j. Observe that we are using parentheses to represent the time-dependency of a scalar variable, such as writing $\boldsymbol{d}(i)$, and subscripts to represent the time-dependency of a vector variable, such as writing $\boldsymbol{u}_i$. This convention will be used throughout this work. In a manner similar to Example 2.1, we again pose the problem of estimating w^o by minimizing the mean-square error cost

$$J(w) = \mathbb{E}\,(\boldsymbol{d}(i) - \boldsymbol{u}_i w)^2 \;\equiv\; \mathbb{E}\,Q(w; \boldsymbol{x}_i) \tag{3.7}$$

where the quantities $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ represent the random data $\boldsymbol{x}_i$ in the definition of the loss function, $Q(w; \boldsymbol{x}_i)$. Using (3.4), the gradient-descent recursion in this case will take the form:

$$w_i \;=\; w_{i-1} \;-\; 2\mu\,[R_u w_{i-1} - r_{du}]\,, \quad i \geq 0 \tag{3.8}$$

The main difficulty in running this recursion is that it requires knowledge of the moments $\{r_{du}, R_u\}$. This information is rarely available beforehand; the adaptive agent senses instead realizations $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ whose statistical distributions have moments $\{r_{du}, R_u\}$. The agent can therefore use these realizations to approximate the moments and the true gradient vector. There are many constructions that can be used for this purpose, with different constructions leading to different adaptive algorithms [107, 205, 206, 262]. It is sufficient to illustrate the construction by focusing on one of the most popular adaptive algorithms, which results from using the data $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ to compute *instantaneous* approximations for the unavailable moments at every time instant as follows:

$$r_{du} \;\approx\; \boldsymbol{d}(i)\boldsymbol{u}_i^{\mathsf{T}}, \qquad R_u \;\approx\; \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i \tag{3.9}$$

By doing so, the true gradient vector is approximated by:

$$\widehat{\nabla_{w^{\mathsf{T}}} J}(w) = 2\left[\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i w - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{d}(i)\right] \;=\; \nabla_{w^{\mathsf{T}}} Q(w; \boldsymbol{x}_i) \tag{3.10}$$

Observe that this construction amounts to replacing the true gradient vector, $\nabla_{w^\mathsf{T}} J(w)$, by the gradient vector of the instantaneous loss function itself (which, equivalently, amounts to dropping the expectation operator):

$$\nabla_{w^\mathsf{T}} J(w) \;=\; \nabla_{\boldsymbol{w}^\mathsf{T}} \mathbb{E}\, Q(w;\boldsymbol{x}_i) \tag{3.11}$$

$$\widehat{\nabla_{w^\mathsf{T}} J}(w) \;=\; \nabla_{\boldsymbol{w}^\mathsf{T}} Q(w;\boldsymbol{x}_i) \tag{3.12}$$

Substituting (3.10) into (3.8) leads to the well-known least-mean-squares (LMS, for short) algorithm [107, 206, 262]:

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;+\; 2\mu \boldsymbol{u}_i^\mathsf{T}[\boldsymbol{d}(i) - \boldsymbol{u}_i \boldsymbol{w}_{i-1}], \quad i \geq 0 \tag{3.13}$$

The LMS algorithm is therefore a stochastic-gradient algorithm. By relying directly on the instantaneous data $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$, the algorithm is infused with useful tracking abilities. This is because drifts in the model w^o from (3.6) will be reflected in the data $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$, which are used directly in (3.13).

■

Example 3.2 (Logistic learner)**.** Let us reconsider the setting of Example 2.2, which dealt with logistic risk functions. Let $\boldsymbol{\gamma}(i)$ be a streaming sequence of binary random variables that assume the values ± 1, and let $\boldsymbol{h}_i$ be a streaming sequence of $M \times 1$ real random (feature) vectors with $R_h = \mathbb{E}\, \boldsymbol{h}_i \boldsymbol{h}_i^\mathsf{T} > 0$. We assume the random processes $\{\boldsymbol{\gamma}(i), \boldsymbol{h}_i\}$ are wide-sense stationary. The objective is to seek the vector w that minimizes the following risk function:

$$J(w) \;\triangleq\; \frac{\rho}{2}\|w\|^2 \;+\; \mathbb{E}\left\{\ln\left(1 + e^{-\boldsymbol{\gamma}(i)\boldsymbol{h}_i^\mathsf{T} w}\right)\right\} \tag{3.14}$$

The loss function that is associated with $J(w)$ is

$$Q(w;\boldsymbol{\gamma}(i),\boldsymbol{h}_i) \;\triangleq\; \frac{\rho}{2}\|w\|^2 \;+\; \ln\left(1 + e^{-\boldsymbol{\gamma}(i)\boldsymbol{h}_i^\mathsf{T} w}\right) \;\equiv\; Q(w;\boldsymbol{x}_i) \tag{3.15}$$

and the stochastic gradient algorithm for minimizing $J(w)$ then takes the form:

$$\boldsymbol{w}_i = (1 - \mu\rho)\boldsymbol{w}_{i-1} \;+\; \mu\boldsymbol{\gamma}(i)\boldsymbol{h}_i \left(\frac{1}{1 + e^{\boldsymbol{\gamma}(i)\boldsymbol{h}_i^\mathsf{T}\boldsymbol{w}_{i-1}}}\right), \quad i \geq 0 \tag{3.16}$$

■

The idea of using sample realizations to approximate actual expectations, as was the case with steps (3.9) and (3.12), is at the core of what is known as *stochastic approximation theory*. According to [206, 243], the pioneering work in the field of stochastic approximation is that of

[200], which is a variation of a scheme developed about two decades earlier in [255]. The work by [200] dealt primarily with *scalar* weights w and was extended by [40, 217] to weight *vectors* — see [258]. During the 1950s, stochastic approximation theory did not receive much attention in the engineering community until the landmark work by [260], in which the authors developed the real form of the LMS algorithm (3.13), which has since then found remarkable success in a wide range of applications.

3.2 Gradient Noise Process

Now, the use of an approximate gradient vector in (3.5) introduces perturbations relative to the operation of the original recursion (3.4). We refer to the perturbation as *gradient noise* and define it as the difference:

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \triangleq \widehat{\nabla_{w^{\mathsf{T}}} J}(\boldsymbol{w}_{i-1}) - \nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}_{i-1}) \tag{3.17}$$

which can also be written as

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \triangleq \nabla_{w^{\mathsf{T}}} Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_i) - \nabla_{w^{\mathsf{T}}} \mathbb{E}\, Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_i) \tag{3.18}$$

for cost functions of the form (3.3) and where, as in cases (3.7) and (3.15), the $\{\boldsymbol{x}_i\}$ represent the data.

The presence of the noise perturbation, $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$, prevents the stochastic iterate, $\boldsymbol{w}_i$, from converging to the minimizer w^o when constant step-sizes are used. Some deterioration in performance occurs since the iterate $\boldsymbol{w}_i$ will instead fluctuate close to w^o in the steady-state regime. We will assess the size of these fluctuations in the next chapter. Here, we argue that they are bounded and that their mean-square-error is in the order of $O(\mu)$ — see (3.39). The next example from [66] illustrates the nature of the gradient noise process (3.17) in the context of mean-square-error adaptation.

Example 3.3 (Gradient noise)**.** It is clear from the expressions in Examples 2.3 and 3.1 that the corresponding gradient noise process is given by:

$$\begin{aligned}
\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) &= \widehat{\nabla_{w^{\mathsf{T}}} J}(\boldsymbol{w}_{i-1}) - \nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}_{i-1}) \\
&= 2\left(\boldsymbol{u}_i^{\mathsf{T}} \boldsymbol{u}_i\right) \boldsymbol{w}_{i-1} - 2\boldsymbol{u}_i^{\mathsf{T}}[\boldsymbol{u}_i w^o + \boldsymbol{v}(i)] - 2R_u \boldsymbol{w}_{i-1} + 2R_u w^o \\
&= 2(R_u - \boldsymbol{u}_i^{\mathsf{T}} \boldsymbol{u}_i)\widetilde{\boldsymbol{w}}_{i-1} - 2\boldsymbol{u}_i^{\mathsf{T}} \boldsymbol{v}(i)
\end{aligned} \tag{3.19}$$

where we introduced the error vector, $\widetilde{\boldsymbol{w}}_i = w^o - \boldsymbol{w}_i$, and used the relations $\boldsymbol{d}(i) = \boldsymbol{u}_i w^o + \boldsymbol{v}(i)$ and $R_u w^o = r_{du}$. Let the symbol $\boldsymbol{\mathcal{F}}_{i-1}$ represent the collection of all possible random events generated by the past iterates $\{\boldsymbol{w}_j\}$ up to time $j \leq i-1$. Formally, $\boldsymbol{\mathcal{F}}_{i-1}$ is the filtration generated by the random process $\boldsymbol{w}_j$ for $j \leq i-1$ (i.e., $\boldsymbol{\mathcal{F}}_{i-1}$ represents the information that is available about the random process $\boldsymbol{w}_j$ up to time $i-1$):

$$\boldsymbol{\mathcal{F}}_{i-1} \stackrel{\Delta}{=} \text{filtration}\,\{\boldsymbol{w}_{-1},\ \boldsymbol{w}_o,\ \boldsymbol{w}_1, \ldots, \boldsymbol{w}_{i-1}\} \tag{3.20}$$

It follows from the conditions on the random processes $\{\boldsymbol{u}_i, \boldsymbol{v}(i)\}$ in Example 3.1 that

$$\begin{aligned} \mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\boldsymbol{\mathcal{F}}_{i-1}\,] &= 2(R_u - \mathbb{E}\,\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i)\widetilde{\boldsymbol{w}}_{i-1} \ -\ 2\mathbb{E}\,\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}(i) \\ &= 2(R_u - R_u)\widetilde{\boldsymbol{w}}_{i-1} \ -\ 2\left(\mathbb{E}\,\boldsymbol{u}_i^{\mathsf{T}}\right)\,(\mathbb{E}\,\boldsymbol{v}(i)) \\ &= 0 \end{aligned} \tag{3.21}$$

and

$$\mathbb{E}\,\left[\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\right] \ \leq\ 4\,c\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \ +\ 4\sigma_v^2\,\mathrm{Tr}(R_u) \tag{3.22}$$

where the constant c is given by

$$c \stackrel{\Delta}{=} \mathbb{E}\,\|R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i\|^2 \tag{3.23}$$

If we take expectations of both sides of (3.22), we further conclude that

$$\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 \ \leq\ 4c\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + 4\sigma_v^2\,\mathrm{Tr}(R_u) \tag{3.24}$$

so that the variance of the gradient noise, $\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2$, is bounded by the combination of two factors. The first factor depends on the quality of the iterate, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2$, while the second factor depends on σ_v^2. Therefore, even if the adaptive agent is able to approach w^o with great fidelity so that $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2$ is small, the size of the gradient noise will still depend on σ_v^2.

■

In order to examine the convergence and performance properties of the stochastic-gradient recursion (3.5), it is necessary to introduce some assumptions on the stochastic nature of the gradient noise process (3.17), whose definition we rewrite more generally as follows for arbitrary vectors $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\boldsymbol{s}_i(\boldsymbol{w}) \stackrel{\Delta}{=} \widehat{\nabla_{w^{\mathsf{T}}} J}(\boldsymbol{w}) \ -\ \nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}) \tag{3.25}$$

The conditions that we state below are similar to conditions used earlier in the optimization literature, e.g., in [190, pp. 95–102] and [33, p. 635]; they are also motivated by the conditions we observed in the mean-square-error case in Example 3.3. Following the developments in [66, 70, 277], we assume the gradient noise process satisfies the following conditions.

Assumption 3.2 (Conditions on gradient noise)**.** It is assumed that the first and second-order conditional moments of the gradient noise process satisfy the following conditions for any $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{3.26}$$

$$\mathbb{E}\,\big[\,\|\boldsymbol{s}_i(\boldsymbol{w})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\big] \;\leq\; \bar{\beta}^2\,\|\boldsymbol{w}\|^2 \;+\; \bar{\sigma}_s^2 \tag{3.27}$$

almost surely, for some nonnegative scalars $\bar{\beta}^2$ and $\bar{\sigma}_s^2$.

Condition (3.26) ensures that the construction of the approximate gradient vector is unbiased. Moreover, using the second condition (3.27), we deduce for any $\boldsymbol{w}_{i-1} \in \boldsymbol{\mathcal{F}}_{i-1}$ that

$$\begin{aligned}
\mathbb{E}\,\Big[\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] &\leq \bar{\beta}^2\,\|\boldsymbol{w}_{i-1}\|^2 \;+\; \bar{\sigma}_s^2 \\
&\overset{\text{(a)}}{=} \bar{\beta}^2\,\|\boldsymbol{w}_{i-1} - w^o + w^o\|^2 \;+\; \bar{\sigma}_s^2 \\
&\overset{\text{(b)}}{\leq} 2\bar{\beta}^2\,\|\boldsymbol{w}_{i-1} - w^o\|^2 \;+\; 2\bar{\beta}^2\|w^o\|^2 \;+\; \bar{\sigma}_s^2 \\
&\overset{\text{(c)}}{\leq} \beta^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \sigma_s^2
\end{aligned} \tag{3.28}$$

where in step (a) we added and subtracted the global minimizer, w^o, and in step (b) we used the inequality $\|x+y\|^2 \leq 2\|x\|^2+2\|y\|^2$ for any vectors x and y, and in step (c) we introduced the nonnegative scalars:

$$\beta^2 \;\triangleq\; 2\bar{\beta}^2 \tag{3.29}$$

$$\sigma_s^2 \;\triangleq\; 2\bar{\beta}^2\|w^o\|^2 \;+\; \bar{\sigma}_s^2 \tag{3.30}$$

In other words, we conclude from conditions (3.26)–(3.27) that the following conditions also hold:

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{3.31}$$

$$\mathbb{E}\,\Big[\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] \;\leq\; \beta^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \sigma_s^2 \tag{3.32}$$

in terms of the error vector, $\widetilde{\boldsymbol{w}}_{i-1} = w^o - \boldsymbol{w}_{i-1}$, and for some nonnegative scalars $\beta^2 \geq 0$ and $\sigma_s^2 \geq 0$. We shall use these conditions more frequently in lieu of (3.26)–(3.27). We could have required these conditions directly in the statement of Assumption 3.2. We instead opted to state conditions (3.26)–(3.27) in that manner, in terms of a generic $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$ rather than $\widetilde{\boldsymbol{w}}_{i-1}$, so that the upper bound in (3.27) is independent of the unknown w^o.

By further taking expectations of the relations (3.31)–(3.32), we conclude that the gradient noise process also satisfies:

$$\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) = 0 \tag{3.33}$$

$$\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 \leq \beta^2\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \sigma_s^2 \tag{3.34}$$

It is straightforward to verify that the gradient noise process (3.19) in the mean-square-error case satisfies conditions (3.31)–(3.32). Note in particular from (3.24) that we can make the identifications

$$\sigma_s^2 \to 4\sigma_v^2\,\mathrm{Tr}(R_u), \quad \beta^2 \to 4c \tag{3.35}$$

3.3 Stability of Second-Order Error Moment

We can now examine the convergence of the stochastic-gradient recursion (3.5) in the mean-square-error sense. Result (3.39) below is stated in terms of the *limit superior* of the error variance sequence, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$. We recall that the limit superior of a sequence essentially corresponds to the smallest upper bound for the limiting behavior of that sequence; this concept is particularly useful when the sequence is not necessarily convergent but tends towards a small bounded region [89, 144, 202]. One such situation is illustrated schematically in Figure 3.1 for the sequence $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$. If the sequence happens to be convergent, then the limit superior will coincide with its regular limiting value.

Lemma 3.1 (Mean-square-error stability: Real case)**.** Assume the conditions under Assumptions 3.1 and 3.2 on the cost function and the gradient noise process hold, and consider the nonnegative scalars $\{\beta^2, \sigma_s^2\}$ defined by (3.29)–(3.30). For any step-size value, μ, satisfying:

$$\mu \;<\; \frac{2\nu}{\delta^2+\beta^2} \tag{3.36}$$

it holds that $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ converges exponentially (i.e., at a geometric rate) according to the recursion

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;\le\; \alpha\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \mu^2\sigma_s^2 \tag{3.37}$$

where the scalar α satisfies $0 \le \alpha < 1$ and is given by

$$\alpha \;=\; 1 - 2\nu\mu + (\delta^2+\beta^2)\mu^2 \tag{3.38}$$

It follows from (3.37) that, for sufficiently small step-sizes:

$$\limsup_{i\to\infty}\; \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O(\mu) \tag{3.39}$$

Proof. While the result can be established in other ways, we follow the alternative route suggested in the proof of the earlier Lemma 2.1 since this argument is more convenient for extensions to the case of networked agents [66, 69, 70, 277]. We subtract w^o from both sides of (3.5) and use (3.17) to get

$$\widetilde{\boldsymbol{w}}_i \;=\; \widetilde{\boldsymbol{w}}_{i-1} \;+\; \mu\,\nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}_{i-1}) \;+\; \mu\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \tag{3.40}$$

We now appeal to the mean-value relation (D.9) from the appendix to write [190]:

$$\begin{aligned}\nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}_{i-1}) \;&=\; -\left(\int_0^1 \nabla_w^2\, J(w^o - t\widetilde{\boldsymbol{w}}_{i-1})dt\right)\widetilde{\boldsymbol{w}}_{i-1}\\ &\triangleq\; -\boldsymbol{H}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}\end{aligned} \tag{3.41}$$

where we are introducing the *symmetric* and *random* time-variant matrix $\boldsymbol{H}_{i-1}$ to represent the integral expression. Substituting into (3.40), we get

$$\widetilde{\boldsymbol{w}}_i \;=\; (I_M - \mu\boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1} \;+\; \mu\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \tag{3.42}$$

so that

$$\begin{aligned}\mathbb{E}\,\left[\|\widetilde{\boldsymbol{w}}_i\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] \;&\le\; \|I_M - \mu\boldsymbol{H}_{i-1}\|^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\\ &\qquad \mu^2\,\mathbb{E}\,\left[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right]\\ &\overset{(3.32)}{\le}\; \|I_M - \mu\boldsymbol{H}_{i-1}\|^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\\ &\qquad \mu^2\,\left(\beta^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \sigma_s^2\right)\end{aligned} \tag{3.43}$$

Using an argument similar to (2.33) we have

$$\begin{aligned}\|I_M - \mu \boldsymbol{H}_{i-1}\|^2 &= [\rho(I_M - \mu \boldsymbol{H}_{i-1})]^2 \\ &\leq \max\{(1-\mu\delta)^2,\ (1-\mu\nu)^2\} \\ &\leq 1 - 2\mu\nu + \mu^2\delta^2 \end{aligned} \tag{3.44}$$

since $\nu \leq \delta$. Substituting into (3.43) and using the definition (3.38) we obtain

$$\mathbb{E}\left[\|\widetilde{\boldsymbol{w}}_i\|^2 \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \leq \alpha\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \mu^2\sigma_s^2 \tag{3.45}$$

Taking expectations of both sides of this inequality we arrive at (3.37). The bound (3.36) on the step-size ensures that $0 \leq \alpha < 1$. Iterating recursion (3.37) gives

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \alpha^{i+1}\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{-1}\|^2 + \frac{\mu^2\sigma_s^2}{1-\alpha} \tag{3.46}$$

which proves that $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ converges exponentially to a region that is upper bounded by

$$\limsup_{i\to\infty}\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \frac{\mu^2\sigma_s^2}{1-\alpha} = \frac{\mu\sigma_s^2}{2\nu - \mu(\delta^2+\beta^2)} \tag{3.47}$$

It is easy to check that the upper bound does not exceed $\mu\sigma_s^2/\nu$ for any step-size $\mu < \nu/(\delta^2+\beta^2)$. We conclude that (3.39) holds for sufficiently small step-sizes.

□

Observe that we can rewrite (3.37) in the equivalent form

$$\left(\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 - \frac{\mu^2\sigma_s^2}{1-\alpha}\right) \leq \alpha\left(\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 - \frac{\mu^2\sigma_s^2}{1-\alpha}\right) \tag{3.48}$$

where the steady-state bound is subtracted from both sides. It is clear from this representation that α relates to the rate of decay of the mean-square-error towards its steady-state bound — see Figure 3.1.

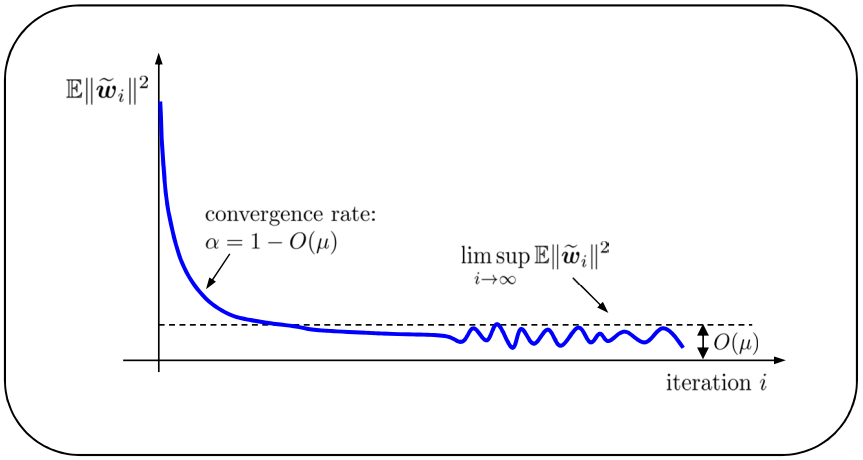

Figure 3.1: Exponential decay of the mean-square error described by (3.37) to a level that is bounded by $O(\mu)$ and at a rate that is in the order of $1 - O(\mu)$.

3.4 Stability of Fourth-Order Error Moment

We can also examine the stability of the fourth-order moment of the error vector by showing that the limit superior of $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^4$ tends asymptotically to a region that is bounded by $O(\mu^2)$. The main motivation for establishing this result, in addition to the stability of the second-order moment already established by (3.39), is that these results will be used in the next chapter to derive expressions that quantify the performance of stochastic gradient algorithms to first-order in the step-size parameter.

To establish the convergence of the fourth-order moment, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^4$, to a bounded region, we need to replace Assumption 3.2 by the following condition on the fourth-order moment of the gradient noise process [71, 278].

Assumption 3.3 (Conditions on gradient noise)**.** It is assumed that the first and fourth-order conditional moments of the gradient noise process satisfy the following conditions for any iterates $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{3.49}$$

$$\mathbb{E}\,\big[\,\|\boldsymbol{s}_i(\boldsymbol{w})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\big] \;\leq\; \bar{\beta}^4\,\|\boldsymbol{w}\|^4 \;+\; \bar{\sigma}_s^4 \tag{3.50}$$

almost surely, for some nonnegative coefficients $\bar{\sigma}_s^4$ and $\bar{\beta}^4$.

It is straightforward to check that if the above condition on the fourth-order moment holds, then a condition similar to (3.27) on the second-order moment will also hold (while the reverse direction is not necessarily true). Indeed, note that

$$\mathbb{E}\,\Big[\,\|\boldsymbol{s}_i(\boldsymbol{w})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] \;\leq\; \Big(\bar{\beta}^2\,\|\boldsymbol{w}\|^2 \;+\; \bar{\sigma}_s^2\Big)^2 \tag{3.51}$$

so that using the property that $(\mathbb{E}\,\boldsymbol{a})^2 \leq \mathbb{E}\,\boldsymbol{a}^2$ for any real random variable $\boldsymbol{a}$, we conclude that

$$\mathbb{E}\,\Big[\,\|\boldsymbol{s}_i(\boldsymbol{w})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] \;\leq\; \bar{\beta}^2\,\|\boldsymbol{w}\|^2 \;+\; \bar{\sigma}_s^2 \tag{3.52}$$

Therefore, the conditions in Assumption 3.3 continue to ensure the mean-square stability of the stochastic-gradient algorithm, as already established by Lemma 3.1.

Now, for any two vectors a and b, it holds that

$$\begin{aligned}
\|a+b\|^4 \;&=\; \left\|\frac{1}{2}\cdot 2a \;+\; \frac{1}{2}\cdot 2b\right\|^4 \\
&\overset{\text{(a)}}{\leq}\; \frac{1}{2}\|2a\|^4 \;+\; \frac{1}{2}\|2b\|^4 \\
&\leq\; 8\|a\|^4 \;+\; 8\|b\|^4
\end{aligned} \tag{3.53}$$

where in step (a) we called upon Jensen's inequality (F.26) from the appendix and applied it to the convex function $f(x) = \|x\|^4$. Using (3.53), it follows from condition (3.50) that the gradient noise process itself satisfies:

$$\begin{aligned}
\mathbb{E}\,\Big[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\Big] \;&\leq\; \bar{\beta}^4\,\|\boldsymbol{w}_{i-1}\|^4 \;+\; \bar{\sigma}_s^4 \\
&=\; \bar{\beta}^4\,\|\boldsymbol{w}_{i-1} - w^o + w^o\|^4 \;+\; \bar{\sigma}_s^4 \\
&\leq\; 8\bar{\beta}^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; 8\bar{\beta}^4\|w^o\|^4 \;+\; \bar{\sigma}_s^4
\end{aligned} \tag{3.54}$$

so that the following conditions also hold:

$$\mathbb{E}\,[\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}] \;=\; 0 \tag{3.55}$$

$$\mathbb{E}\left[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] \;\leq\; \beta_4^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 + \sigma_{s4}^4 \tag{3.56}$$

where we introduced the non-negative parameters:

$$\beta_4^4 \;\triangleq\; 8\bar{\beta}^4 \tag{3.57}$$

$$\sigma_{s4}^4 \;\triangleq\; 8\bar{\beta}^4\|w^o\|^4 \;+\; \bar{\sigma}_s^4 \tag{3.58}$$

We shall use conditions (3.55)–(3.56) more frequently in lieu of (3.49)–(3.50). By taking expectations of (3.55)–(3.56) we obtain:

$$\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;=\; 0 \tag{3.59}$$

$$\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4 \;\leq\; \beta_4^4\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 + \sigma_{s4}^4 \tag{3.60}$$

The following example illustrates that the mean-square-error cost considered earlier in Examples 3.1 and 3.2 satisfies the conditions of Assumption 3.3.

Example 3.4 (Mean-square error costs)**.** Let us consider the same scenario from Example 3.3 where we determined in (3.19) that the gradient noise process is given by

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;=\; 2(R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i)\widetilde{\boldsymbol{w}}_{i-1} \;-\; 2\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}(i) \tag{3.61}$$

It follows that

$$\begin{aligned}\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4 &\overset{(3.53)}{\leq} 8\|2(R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i)\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; 8\|2\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}(i)\|^4 \\ &\leq 128\|R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i\|^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; 128\|\boldsymbol{u}_i\|^4\,\|\boldsymbol{v}(i)\|^4\end{aligned} \tag{3.62}$$

From the conditions on the random processes $\{\boldsymbol{u}_i, \boldsymbol{v}(i)\}$ in Example 3.1, and assuming further that the fourth-order moments of $\{\boldsymbol{v}(i), \boldsymbol{u}_i\}$ are bounded and independent of i, we get

$$\begin{aligned}\mathbb{E}\,\left[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] &\leq 128\left(\mathbb{E}\,\|R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i\|^4\right)\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+ \\ &\qquad 128\left(\mathbb{E}\,\|\boldsymbol{u}_i\|^4\right)\left(\mathbb{E}\,\|\boldsymbol{v}(i)\|^4\right) \\ &\triangleq \beta_4^4\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; \sigma_{s4}^4\end{aligned} \tag{3.63}$$

which is of the same form as (3.60) with

$$\beta_4^4 \triangleq 128\left(\mathbb{E}\,\|R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i\|^4\right) \tag{3.64}$$

$$\sigma_{s4}^4 \triangleq 128\left(\mathbb{E}\,\|\boldsymbol{u}_i\|^4\right)\left(\mathbb{E}\,\|\boldsymbol{v}(i)\|^4\right) \tag{3.65}$$

■

In a manner similar to Lemma 3.1 we can now argue that the evolution of the fourth-order moment of the weight-error vector is also stable [71, 278].

Lemma 3.2 (Stability of fourth-order moment: Real case)**.** Assume the conditions under Assumptions 3.1 and 3.3 on the cost function and the gradient noise process hold. Then, for sufficiently small step-sizes, it holds that

$$\limsup_{i\to\infty}\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 = O(\mu) \tag{3.66}$$

$$\limsup_{i\to\infty}\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^4 = O(\mu^2) \tag{3.67}$$

Proof. We only need to establish (3.67) since (3.66) was established earlier in Lemma 3.1. Following an argument similar to [278], we refer to the error recursion (3.42):

$$\widetilde{\boldsymbol{w}}_i = (I_M - \mu\boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1} + \mu\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \tag{3.68}$$

Using the fact that, for any vectors a and b,

$$\|a+b\|^4 = \|a\|^4 + \|b\|^4 + 2\|a\|^2\,\|b\|^2 + 4(a^{\mathsf{T}}b)^2 + 4\|b\|^2\,a^{\mathsf{T}}b + 4\|a\|^2\,a^{\mathsf{T}}b \tag{3.69}$$

we can equate the fourth-order powers of both sides of (3.68) to get

$$\begin{aligned}\|\widetilde{\boldsymbol{w}}_i\|^4 = {} & \|(I_M - \mu\boldsymbol{H}_{i-1})\,\widetilde{\boldsymbol{w}}_{i-1}\|^4 + \mu^4\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4 + \\ & 2\mu^2\|(I_M - \mu\boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1}\|^2\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 + \\ & 4\mu^2\left[\widetilde{\boldsymbol{w}}_{i-1}^{\mathsf{T}}(I_M - \mu\boldsymbol{H}_{i-1})\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\right]^2 + \\ & 4\mu^2\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\left[\widetilde{\boldsymbol{w}}_{i-1}^{\mathsf{T}}(I_M - \mu\boldsymbol{H}_{i-1})\mu\boldsymbol{s}(\boldsymbol{w}_{i-1})\right] + \\ & 4\|(I_M - \mu\boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1}\|^2\left[\widetilde{\boldsymbol{w}}_{i-1}^{\mathsf{T}}(I_M - \mu\boldsymbol{H}_{i-1})\mu\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\right]\end{aligned} \tag{3.70}$$

Applying the Cauchy Schwarz's inequality $(a^{\mathsf{T}}b)^2 \leq \|a\|^2\,\|b\|^2$ to the third term on the right-hand side, and using the sub-multiplicative property of norms, we get

$$\begin{aligned}\|\widetilde{\boldsymbol{w}}_i\|^4 \;\leq\; & \|I_M-\mu\boldsymbol{H}_{i-1}\|^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; \mu^4\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4 + \\ & 6\mu^2\|(I_M-\mu\boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1}\|^2\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 + \\ & 4\mu^2\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\,\left[\widetilde{\boldsymbol{w}}_{i-1}^{\mathsf{T}}(I_M-\mu\boldsymbol{H}_{i-1})\mu\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\right] + \\ & 4\|(I_M-\mu\boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1}\|^2\,\left[\widetilde{\boldsymbol{w}}_{i-1}^{\mathsf{T}}(I_M-\mu\boldsymbol{H}_{i-1})\mu\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\right]\end{aligned} \tag{3.71}$$

Applying further the inequality $2a^{\mathsf{T}}b \leq \|a\|^2+\|b\|^2$ to the rightmost factor in the third line, and using again the sub-multiplicative property of norms, we get

$$\begin{aligned}\|\widetilde{\boldsymbol{w}}_i\|^4 \;\leq\; & \|I_M-\mu\boldsymbol{H}_{i-1}\|^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; 3\mu^4\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4 + \\ & 8\mu^2\|I_M-\mu\boldsymbol{H}_{i-1}\|^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 + \\ & 4\|(I_M-\mu\boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1}\|^2\,\left[\widetilde{\boldsymbol{w}}_{i-1}^{\mathsf{T}}(I_M-\mu\boldsymbol{H}_{i-1})\mu\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\right]\end{aligned} \tag{3.72}$$

Conditioning both sides of (3.72) on $\boldsymbol{\mathcal{F}}_{i-1}$ and using (3.55) and (3.56), we obtain

$$\begin{aligned}\mathbb{E}\,\left[\|\widetilde{\boldsymbol{w}}_i\|^4|\boldsymbol{\mathcal{F}}_{i-1}\right] \;\leq\; & \|I_M-\mu\boldsymbol{H}_{i-1}\|^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+ \\ & 3\mu^4(\beta_4^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4+\sigma_{s4}^4) + \\ & 8\mu^2\,\|I_M-\mu\boldsymbol{H}_{i-1}\|^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\,(\beta^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\sigma_s^2)\end{aligned} \tag{3.73}$$

where the expectation of the last term on the right-hand side of (3.72) is zero since $\mathbb{E}\,[\boldsymbol{s}_i(\boldsymbol{w}_{i-1})|\boldsymbol{\mathcal{F}}_{i-1}]=0$. Using an argument similar to (3.44) we have

$$\begin{aligned}\|I_M-\mu\boldsymbol{H}_{i-1}\|^2 \;&\leq\; 1-2\mu\nu+\mu^2\delta^2 \\ &<\; 1+\mu^2\delta^2\end{aligned} \tag{3.74}$$

and

$$\begin{aligned}\|I_M-\mu\boldsymbol{H}_{i-1}\|^4 \;&\leq\; (1-2\mu\nu+\mu^2\delta^2)^2 \\ &=\; 1-4\mu\nu+2\mu^2(2\nu^2+\delta^2)+\mu^4\delta^4-4\mu^3\nu\delta^2 \\ &<\; 1-4\mu\nu+2\mu^2(2\nu^2+\delta^2)+\mu^4\delta^4\end{aligned} \tag{3.75}$$

Substituting these bounds into (3.73), taking expectations of both sides again to eliminate the conditioning on $\boldsymbol{\mathcal{F}}_{i-1}$, and grouping terms we get

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^4 \;\leq\; (1-a_1)\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; a_2\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; a_3 \tag{3.76}$$

where the constants $\{a_1, a_2, a_3\}$ are defined by

$$\begin{aligned} a_1 &= 4\mu\nu - 2\mu^2(2\nu^2+\delta^2+4\beta^2) - \mu^4(\delta^4+8\beta^2\delta^2+3\beta_4^4) \\ &= O(\mu) \qquad (3.77) \\ a_2 &= 8\mu^2(1+\mu^2\delta^2)\sigma_s^2 \;=\; O(\mu^2) \qquad (3.78) \\ a_3 &= 3\mu^4\sigma_{s4}^4 \;=\; O(\mu^4) \qquad (3.79) \end{aligned}$$

We can combine (3.76) and the earlier mean-square-error inequality (3.37) into a single linear recursive inequality as follows:

$$\begin{bmatrix} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^4 \end{bmatrix} \preceq \begin{bmatrix} \alpha & 0 \\ a_2 & (1-a_1) \end{bmatrix} \begin{bmatrix} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \end{bmatrix} + \begin{bmatrix} \mu^2\sigma_s^2 \\ a_3 \end{bmatrix} \qquad (3.80)$$

where the notation $a \preceq b$ means that each entry of vector a is smaller than or equal to the corresponding entry in vector b. We already know from (3.36) that for $\mu < 2\nu/(\delta^2+\beta^2)$, it will hold that $0 \leq \alpha < 1$ so that the mean-square error, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, converges asymptotically to a region bounded by $O(\mu)$. We can therefore ensure the convergence of recursion (3.80) by showing that a small enough step-size can be chosen to further enforce $|1-a_1| < 1$ or, equivalently, $0 < a_1 < 2$. Since we know from (3.77) that $a_1 < 4\mu\nu$, then selecting μ according to the following three conditions is sufficient to meet the requirement $0 < a_1 < 2$ (these conditions combined guarantee $\mu\nu < a_1 < 2$):

$$\begin{aligned} 4\mu\nu &< 2 \qquad (3.81) \\ \mu^4(\delta^4+8\beta^2\delta^2+3\beta_4^4) &< \mu^2(2\nu^2+\delta^2+4\beta^2) \qquad (3.82) \\ \mu^2(2\nu^2+\delta^2+4\beta^2) &< \mu\nu \qquad (3.83) \end{aligned}$$

or, since $\delta \geq \nu$,

$$\begin{aligned} \mu &< 1/2\delta \qquad (3.84) \\ \mu &< \left(\frac{2\nu^2+\delta^2+4\beta^2}{\delta^4+8\beta^2\delta^2+3\beta_4^4}\right)^{1/2} \qquad (3.85) \\ \mu &< \frac{\nu}{2\nu^2+\delta^2+4\beta^2} \qquad (3.86) \end{aligned}$$

Since the bounds on the right-hand side are positive constants and independent of μ, it is clear that a sufficiently small μ exists that meets all three conditions and leads to $|1-a_1| < 1$. For example, the smallest bound among the above three bounds determines an upper limit, μ_o, such that for all $\mu < \mu_o$ we get $0 < a_1 < 2$:

$$\mu_o \;=\; \min\left\{\frac{1}{2\delta},\; \frac{\nu}{2\nu^2+\delta^2+4\beta^2},\; \left(\frac{2\nu^2+\delta^2+4\beta^2}{\delta^4+8\beta^2\delta^2+3\beta_4^4}\right)^{1/2}\right\} \qquad (3.87)$$

It is clear that

$$\frac{\nu}{2\nu^2+\delta^2+4\beta^2} \;<\; \frac{\nu}{\delta^2+\beta^2} \tag{3.88}$$

Therefore, any $\mu < \mu_o$ also satisfies $\mu < \nu/(\delta^2+\beta^2)$ and $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ will be mean-square stable according to (3.36), i.e.,

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;\leq\; b\mu \tag{3.89}$$

for some constant $b > 0$. Computing the limit superior of both sides of (3.76) then gives:

$$\begin{aligned}
\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^4 &\leq \frac{a_2 b\mu + a_3}{a_1} \\
&\overset{\text{(a)}}{\leq} \frac{8\mu^2(1+\mu^2\delta^2)\sigma_s^2 b\mu \;+\; 3\mu^4\sigma_{s4}^4}{\mu\nu} \\
&\leq \left(\frac{8b\sigma_s^2}{\nu}\right)\mu^2 \;+\; \left(\frac{3\sigma_{s4}^4}{\nu}\right)\mu^3 \;+\; \left(\frac{8b\sigma_s^2\delta^2}{\nu}\right)\mu^4 \\
&\overset{\text{(b)}}{\leq} \left(\frac{8b\sigma_s^2}{\nu}\right)\mu^2 + \left(\frac{3\sigma_{s4}^4}{2\nu^2}\right)\mu^2 + \left(\frac{2b\sigma_s^2\delta^2}{\nu^3}\right)\mu^2 \\
&= O(\mu^2)
\end{aligned} \tag{3.90}$$

where step (a) is because $a_1 > \mu\nu$ and step (b) is because $\mu < 1/2\nu$.

□

3.5 Decaying Step-Size Sequences

If desired, it is also possible to employ iteration-dependent step-size sequences in (3.5) instead of the constant step-size μ, and to require $\mu(i) > 0$ to satisfy either of the following two sets of conditions:

$$\sum_{i=0}^{\infty} \mu(i) = \infty, \qquad \lim_{i\to\infty} \mu(i) = 0 \tag{3.91}$$

or

$$\sum_{i=0}^{\infty} \mu(i) = \infty, \qquad \sum_{i=0}^{\infty} \mu^2(i) < \infty \tag{3.92}$$

The first set of conditions is the same one we encountered before in (2.38). The second set of conditions is stronger: if a sequence $\mu(i)$ satisfies (3.92) then it also satisfies (3.91). In either case, recursion (3.5)

would be replaced by

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \mu(i)\,\widehat{\nabla_{w^{\mathsf{T}}} J}(\boldsymbol{w}_{i-1}), \quad i \geq 0 \tag{3.93}$$

It is well-known [32, 190, 243] that the iterate $\boldsymbol{w}_i$ converges towards w^o in the mean-square sense under (3.91), i.e.,

$$\lim_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 = 0 \qquad \text{(under (3.91))} \tag{3.94}$$

and it converges to w^o almost surely, i.e., with probability one, under (3.92):

$$\text{Prob}\left(\lim_{i\to\infty} \boldsymbol{w}_i = w^o\right) \;=\; 1 \quad \text{(under (3.92))} \tag{3.95}$$

However, as already noted before, conditions (3.91)–(3.92) force the step-size sequence to decay to zero, which is problematic for applications requiring continuous adaptation from streaming data.

Lemma 3.3 (Almost-sure convergence: Real case)**.** Assume the conditions under Assumptions 3.1 and 3.2 on the cost function and the gradient noise process hold. Then, the following convergence properties hold for (3.93):

(a) If the step-size sequence $\mu(i)$ satisfies (3.92), then $\boldsymbol{w}_i$ converges almost surely to w^o, written as $\boldsymbol{w}_i \to w^o$ a.s.

(b) If the step-size sequence $\mu(i)$ satisfies (3.91), then $\boldsymbol{w}_i$ converges in the mean-square-error sense to w^o, i.e., $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \to 0$.

Proof. We again subtract w^o from both sides of (3.93) to get

$$\widetilde{\boldsymbol{w}}_i \;=\; \widetilde{\boldsymbol{w}}_{i-1} \;+\; \mu(i)\,\nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}_{i-1}) \;+\; \mu(i)\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \tag{3.96}$$

We then use the mean-value relation (D.7) from the appendix to note that

$$\nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}_{i-1}) \;=\; \underbrace{\left(\int_0^1 \nabla_w^2\, J(w^o - t\widetilde{\boldsymbol{w}}_{i-1})dt\right)}_{\triangleq\, \boldsymbol{H}_{i-1}} \widetilde{\boldsymbol{w}}_{i-1} \tag{3.97}$$

where we are introducing the *symmetric* and *random* time-variant matrix $\boldsymbol{H}_{i-1}$, which is defined in terms of the Hessian of the cost function; note that this matrix depends on the random error vector $\widetilde{\boldsymbol{w}}_{i-1}$. Substituting the above relation into (3.96), we get the recursion

$$\widetilde{\boldsymbol{w}}_i \;=\; (I_M - \mu(i)\boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1} \;+\; \mu(i)\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \tag{3.98}$$

It then follows that

$$\begin{aligned}\mathbb{E}\left[\|\widetilde{\boldsymbol{w}}_i\|^2 \mid \boldsymbol{\mathcal{F}}_{i-1}\right] &\leq \|I_M - \mu(i)\boldsymbol{H}_{i-1}\|^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \\ &\quad \mu^2(i)\,\mathbb{E}\left[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2|\boldsymbol{\mathcal{F}}_{i-1}\right] \\ &\overset{\text{(a)}}{\leq} (1-2\mu(i)\nu+\delta^2\mu^2(i))\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \\ &\quad \beta^2\mu^2(i)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \mu^2(i)\sigma_s^2 \end{aligned} \tag{3.99}$$

where step (a) uses an argument similar to (3.44). Therefore, it holds that:

$$\mathbb{E}\left[\,\|\widetilde{\boldsymbol{w}}_i\|^2 \mid \boldsymbol{\mathcal{F}}_{i-1}\,\right] \;\leq\; \alpha(i)\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \mu^2(i)\sigma_s^2 \tag{3.100}$$

where

$$\alpha(i) \;\triangleq\; 1-2\nu\mu(i)+(\delta^2+\beta^2)\mu^2(i) \tag{3.101}$$

Now note that we can split the term $2\nu\mu(i)$ in the above expression for $\alpha(i)$ into the sum of two terms and write

$$\alpha(i) \;=\; 1-\nu\mu(i)-\nu\mu(i)+(\delta^2+\beta^2)\mu^2(i) \tag{3.102}$$

And since $\mu(i)\to 0$, we conclude that for large enough $i > i_o$, the sequence $\mu^2(i)$ will assume smaller values than $\mu(i)$. Therefore, a large enough time index, i_o, exists such that the following two conditions are satisfied:

$$\nu\mu(i) \geq (\delta^2+\beta^2)\mu^2(i), \qquad 0 \leq \nu\mu(i) < 1, \qquad i > i_o \tag{3.103}$$

Consequently,

$$\alpha(i) \;\leq\; 1-\nu\mu(i), \qquad i > i_o \tag{3.104}$$

Then, inequalities (3.100) and (3.104) imply that

$$\mathbb{E}\left[\,\|\widetilde{\boldsymbol{w}}_i\|^2 \mid \boldsymbol{\mathcal{F}}_{i-1}\,\right] \;\leq\; (1-\nu\mu(i))\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \mu^2(i)\sigma_s^2, \quad i > i_o \tag{3.105}$$

For convenience of notation, let

$$\boldsymbol{u}(i+1) \;\triangleq\; \|\widetilde{\boldsymbol{w}}_i\|^2 \tag{3.106}$$

Then, inequality (3.105) implies that

$$\mathbb{E}\left[\boldsymbol{u}(i+1)|\,\boldsymbol{u}(0),\boldsymbol{u}(1),\ldots,\boldsymbol{u}(i)\,\right] \;\leq\; (1-\nu\mu(i))\;\boldsymbol{u}(i) \;+\; \mu^2(i)\sigma_s^2, \;\; i > i_o \tag{3.107}$$

We now call upon the useful result (F.53) from the appendix and make the identifications

$$a(i) = \nu\mu(i), \quad b(i) = \mu^2(i)\sigma_s^2 \tag{3.108}$$

These sequences satisfy conditions (F.54) in the appendix in view of assumption (3.92) on the step-size sequence and the second condition in (3.103). We then conclude that $\boldsymbol{u}(i) \to 0$ almost surely and, hence, $\boldsymbol{w}_i \to w^o$ almost surely.

Finally, taking expectations of both sides of (3.107) leads to

$$\mathbb{E}\,\boldsymbol{u}(i+1) \;\leq\; (1-\nu\mu(i))\,\mathbb{E}\,\boldsymbol{u}(i) \;+\; \mu^2(i)\sigma_s^2, \quad i > i_o \tag{3.109}$$

with the expectation operator appearing on both sides of the inequality. Then, we conclude from result (F.49) in the appendix, under conditions (3.91), that $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \to 0$ so that $\boldsymbol{w}_i$ converges to w^o in the mean-square-error sense.

□

We can be more specific and quantify the rate at which the variance $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ converges towards zero for step-size sequences of the form:

$$\mu(i) = \frac{\tau}{i+1}, \quad \xi > 0 \tag{3.110}$$

which satisfy both conditions (3.91) and (3.92). In contrast to the result of Lemma 2.2 on the convergence rate of gradient descent algorithms, which was seen to be in the order of $O(1/i^{2\nu\tau})$, the next statement indicates that now three rates of convergence are possible depending on how $\nu\tau$ compares to the value one.

Lemma 3.4 (Rates of convergence for a decaying step-size)**.** Assume the conditions under Assumptions 3.1 and 3.2 on the cost function and the gradient noise process hold. Assume further that the step-size sequence is selected according to (3.110). Then, three convergence rates are possible depending on how the factor $\nu\tau$ compares to the value one. Specifically, for large enough i, it holds that:

$$\begin{cases} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;\leq\; \left(\frac{\tau^2\sigma_s^2}{\nu\tau-1}\right)\frac{1}{i} \;+\; o\left(\frac{1}{i}\right), & \nu\tau > 1 \\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O\left(\frac{\log i}{i}\right), & \nu\tau = 1 \\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O\left(\frac{1}{i^{\nu\tau}}\right), & \nu\tau < 1 \end{cases} \tag{3.111}$$

The fastest convergence rate occurs when $\nu\tau > 1$ (i.e., for large enough τ) and is in the order of $O(1/i)$.

Proof. We use (3.109) and the assumed form for $\mu(i)$ in (3.110) to write

$$\mathbb{E}\,\boldsymbol{u}(i+1) \;\leq\; \left(1 - \frac{\nu\tau}{i+1}\right)\mathbb{E}\,\boldsymbol{u}(i) \;+\; \frac{\tau^2\sigma_s^2}{(i+1)^2}, \quad i > i_o \tag{3.112}$$

This recursion has the same form as recursion (F.49) in the appendix with the identifications

$$a(i) = \frac{\nu\tau}{i+1}, \quad b(i) = \frac{\tau^2\sigma_s^2}{(i+1)^2}, \quad p = 1 \tag{3.113}$$

The above rates of convergence then follow from the statement in part (b) of Lemma F.5 in the appendix.

□

3.6 Optimization in the Complex Domain

We now extend the previous results to the case in which the argument $w \in \mathbb{C}^M$ is complex-valued. As was explained earlier in Sec. 2.5, the strongly-convex function, $J(w) \in \mathbb{R}$, is required to satisfy condition (2.62), namely,

$$0 < \frac{\nu}{h} I_{hM} \;\le\; \nabla_w^2 J(w) \;\le\; \frac{\delta}{h} I_{hM} \tag{3.114}$$

in terms of the data-type variable

$$h \;\triangleq\; \begin{cases} 1, & \text{when } w \text{ is real} \\ 2, & \text{when } w \text{ is complex} \end{cases} \tag{3.115}$$

Condition (3.114) captures the requirements that $J(w)$ is twice-differentiable, $\nu-$strongly convex, and has a $\delta-$Lipschitz gradient vector function. The condition is also applicable to both cases of real and complex data. In this section, we are interested in the case $h = 2$ corresponding to complex data. The previous sections studied the case $h = 1$.

In the complex domain, the stochastic gradient recursions (3.4) and (3.93) are replaced by

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \mu\,\widehat{\nabla_{w^*} J}(\boldsymbol{w}_{i-1}), \quad i \ge 0 \tag{3.116}$$

and

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \mu(i)\,\widehat{\nabla_{w^*} J}(\boldsymbol{w}_{i-1}), \quad i \ge 0 \tag{3.117}$$

respectively, where the second form employs an iteration-dependent step-size sequence. Comparing with (3.4) and (3.93) we see that transposition of the approximate gradient vector is replaced by complex

conjugation. We again denote the approximation error by the gradient noise model:

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;\triangleq\; \widehat{\nabla_{w^*} J}(\boldsymbol{w}_{i-1}) \;-\; \nabla_{w^*} J(\boldsymbol{w}_{i-1}) \tag{3.118}$$

This noise process is now complex-valued.

Example 3.5 (LMS adaptation in the complex domain)**.** We extend the formulation of Examples 3.1 and 3.3 to the complex case. Thus, let $\boldsymbol{d}(i)$ denote a streaming sequence of zero-mean (now complex-valued) random variables with variance $\sigma_d^2 = \mathbb{E}\,|\boldsymbol{d}(i)|^2$. Let $\boldsymbol{u}_i$ denote a streaming sequence of $1 \times M$ independent zero-mean (now complex-valued) random vectors with covariance matrix $R_u = \mathbb{E}\,\boldsymbol{u}_i^*\boldsymbol{u}_i > 0$. Both processes $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ are assumed to be jointly wide-sense stationary. The cross-covariance vector between $\boldsymbol{d}(i)$ and $\boldsymbol{u}_i$ is denoted by $r_{du} = \mathbb{E}\,\boldsymbol{d}(i)\boldsymbol{u}_i^*$. The data $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ are assumed to be related via the same linear regression model

$$\boldsymbol{d}(i) \;=\; \boldsymbol{u}_i w^o \;+\; \boldsymbol{v}(i) \tag{3.119}$$

for some unknown parameter vector w^o, and where $\boldsymbol{v}(i)$ is a zero-mean white-noise process with power $\sigma_v^2 = \mathbb{E}\,|\boldsymbol{v}(i)|^2$ and assumed independent of $\boldsymbol{u}_j$ for all i, j. In a manner similar to Example 2.1, we again pose the problem of estimating w^o by minimizing the mean-square error cost

$$\begin{aligned} J(w) &= \mathbb{E}\,|\boldsymbol{d}(i) - \boldsymbol{u}_i w|^2 \\ &= \sigma_d^2 - r_{du}^* w - w^* r_{du} + w^* R_u w \\ &\equiv \mathbb{E}\,Q(w; \boldsymbol{x}_i) \end{aligned} \tag{3.120}$$

where the quantities $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ represent the random data $\boldsymbol{x}_i$ in the definition of $Q(w; \boldsymbol{x}_i)$. Using (2.66), the gradient-descent recursion in this case will take the form:

$$w_i \;=\; w_{i-1} \;-\; \mu\,[R_u w_{i-1} - r_{du}]\,, \quad i \geq 0 \tag{3.121}$$

Observe that the factor of 2 that used to appear multiplying μ in (3.8) in the real case is not needed here since now

$$\nabla_{w^*} J(w_{i-1}) = R_u w_{i-1} - r_{du} \tag{3.122}$$

Again, the main difficulty in running (3.121) is that it requires knowledge of the moments $\{r_{du}, R_u\}$. Using the *instantaneous* approximations:

$$r_{du} \;\approx\; \boldsymbol{d}(i)\boldsymbol{u}_i^*, \qquad R_u \;\approx\; \boldsymbol{u}_i^*\boldsymbol{u}_i \tag{3.123}$$

we can replace the true gradient vector by the approximation:

$$\widehat{\nabla_{w^*} J}(w) = [\boldsymbol{u}_i^* \boldsymbol{u}_i w - \boldsymbol{u}_i^* \boldsymbol{d}(i)] \;=\; \nabla_{w^*} Q(w; \boldsymbol{x}_i) \tag{3.124}$$

Substituting (3.124) into (3.121) leads to the complex form of the least-mean-squares (LMS) algorithm [107, 206, 262]:

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;+\; \mu \boldsymbol{u}_i^* [\boldsymbol{d}(i) - \boldsymbol{u}_i \boldsymbol{w}_{i-1}], \quad i \geq 0 \tag{3.125}$$

It can be verified from the construction of the approximate gradient vector that the corresponding gradient noise process is now given by

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;=\; (R_u - \boldsymbol{u}_i^* \boldsymbol{u}_i) \widetilde{\boldsymbol{w}}_{i-1} \;-\; \boldsymbol{u}_i^* \boldsymbol{v}(i) \tag{3.126}$$

in terms of $\widetilde{\boldsymbol{w}}_i = w^o - \boldsymbol{w}_i$. If we again let $\boldsymbol{\mathcal{F}}_{i-1}$ represent filtration generated by the random process $\boldsymbol{w}_j$ for $j \leq i-1$, we readily obtain that

$$\mathbb{E}\,[\, \boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \,|\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{3.127}$$

$$\mathbb{E}\,\big[\, \|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 \,\big|\, \boldsymbol{\mathcal{F}}_{i-1}\,\big] \;\leq\; c\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \sigma_v^2\,\mathrm{Tr}(R_u) \tag{3.128}$$

where the constant c is given by

$$c \;\triangleq\; \mathbb{E}\,\|R_u - \boldsymbol{u}_i^* \boldsymbol{u}_i\|^2 \tag{3.129}$$

If we take expectations of both sides of (3.128), we further conclude that

$$\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 \;\leq\; c\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \sigma_v^2\,\mathrm{Tr}(R_u) \tag{3.130}$$

so that the variance of the gradient noise, $\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2$, is again bounded by the combination of two factors. The first factor depends on the quality of the iterate, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2$, while the second factor depends on σ_v^2.

■

In a manner similar to Assumption 3.2, we assume the gradient noise process satisfies the following conditions. The statement below is applicable to both cases of real and complex data through the use of the data-type variable: $h = 1$ for real data and $h = 2$ for complex data.

Assumption 3.4 (Conditions on gradient noise: Complex case)**.** It is assumed that the first and second-order conditional moments of the gradient noise process satisfy the following conditions for any $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\mathbb{E}\,[\, \boldsymbol{s}_i(\boldsymbol{w}) \,|\, \boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{3.131}$$

$$\mathbb{E}\,\big[\, \|\boldsymbol{s}_i(\boldsymbol{w})\|^2 \,|\, \boldsymbol{\mathcal{F}}_{i-1}\,\big] \;\leq\; \big(\bar{\beta}/h\big)^2\,\|\boldsymbol{w}\|^2 \;+\; \bar{\sigma}_s^2 \tag{3.132}$$

almost surely, for some nonnegative scalars $\bar{\beta}^2$ and $\bar{\sigma}_s^2$.

In a manner similar to the derivation of (3.31)–(3.32) in the real case, we can again verify that the above two conditions lead to the following forms, which we shall use frequently:

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{3.133}$$

$$\mathbb{E}\left[\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\right] \;\leq\; (\beta/h)^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \sigma_s^2 \tag{3.134}$$

and where the scalars $\{\beta^2, \sigma_s^2\}$ are defined by

$$\beta^2 \;\triangleq\; 2\bar{\beta}^2 \tag{3.135}$$

$$\sigma_s^2 \;\triangleq\; 2(\bar{\beta}/h)^2\|w^o\|^2 \;+\; \bar{\sigma}_s^2 \tag{3.136}$$

By taking expectations of (3.133)–(3.134), we conclude that the gradient noise process also satisfies:

$$\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;=\; 0 \tag{3.137}$$

$$\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 \;\leq\; (\beta/h)^2\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \sigma_s^2 \tag{3.138}$$

It is straightforward to verify from Example 3.5 that the gradient noise process in the mean-square-error case satisfies conditions (3.133)–(3.134). Note in particular from (3.130) that we can make the identifications

$$\sigma_s^2 \to \sigma_v^2\,\mathrm{Tr}(R_u), \quad \beta^2 \to 4c \tag{3.139}$$

Stability of Second-Order Error Moment

The next statement extends Lemma 3.1 to the complex case and ascertains the mean-square-error stability of recursion (3.116).

Lemma 3.5 (Mean-square-error stability: Complex case)**.** Assume the cost function $J(w)$ satisfies (3.114) and the gradient noise process satisfies the conditions in Assumption 3.4, and consider the nonnegative scalars $\{\beta^2, \sigma_s^2\}$ defined by (3.135)–(3.136). If the step-size parameter is chosen to satisfy

$$\frac{\mu}{h} < \frac{2\nu}{\delta^2 + \beta^2} \tag{3.140}$$

Then, it holds that for any initial condition, $\boldsymbol{w}_{-1}$, the mean-square error, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, converges exponentially (i.e., at a geometric rate) according to the recursion:

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;\leq\; \alpha\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \mu^2\sigma_s^2 \tag{3.141}$$

where

$$\alpha \;=\; 1 - 2\nu\left(\frac{\mu}{h}\right) \;+\; (\delta^2+\beta^2)\left(\frac{\mu}{h}\right)^2 \tag{3.142}$$

It follows from (3.141) that, for sufficiently small step-sizes:

$$\limsup_{i\to\infty}\; \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O(\mu) \tag{3.143}$$

Proof. We apply the result of Lemma 3.1 to the $v-$domain recursion:

$$\boldsymbol{v}_i \;=\; \boldsymbol{v}_{i-1} \;-\; \mu'\,\widehat{\nabla_{v^{\mathsf{T}}} J}(\boldsymbol{v}_{i-1}) \tag{3.144}$$

where $\mu' = \mu/2$ and $\boldsymbol{v}_i = \mathrm{col}\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ in terms of the real and imaginary parts of $\boldsymbol{w}_i = \boldsymbol{x}_i + j\boldsymbol{y}_i$. We already know from (E.39) in the appendix that $J(v)$ is $\nu-$strongly convex since $J(w)$ is $\nu-$strongly convex. We also know from from (E.22) and (E.56) in the same appendix that the gradient vector function of $J(v)$ is $\delta-$Lipschitz. Therefore, the equivalent function $J(v)$, defined in terms of the real-valued argument v, satisfies the conditions stated in Lemma 3.1. All that remains to check is to identify the nature of the gradient noise associated with the modified recursion (3.144) and to verify that this noise satisfies conditions of the same form required by Assumption 3.2. Let us denote the gradient noise of the above recursion in the $v-$domain by

$$\boldsymbol{t}_i(\boldsymbol{v}_{i-1}) \;\triangleq\; \widehat{\nabla_{v^{\mathsf{T}}} J}(\boldsymbol{v}_{i-1}) \;-\; \nabla_{v^{\mathsf{T}}} J(\boldsymbol{v}_{i-1}) \tag{3.145}$$

We now express $\boldsymbol{t}_i(\cdot)$ in terms of the original gradient noise $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$ from the $w-$domain given by (3.118). To begin with, recursion (3.144) is equivalent to

$$\boldsymbol{v}_i \;=\; \boldsymbol{v}_{i-1} \;-\; \frac{\mu}{2}\,\nabla_{v^{\mathsf{T}}}\, J(\boldsymbol{v}_{i-1}) \;-\; \frac{\mu}{2}\,\boldsymbol{t}_i(\boldsymbol{v}_{i-1}) \tag{3.146}$$

Multiplying (3.146) from the left by the matrix D from (B.27) in the appendix and using (C.32), we can transform the above recursion into the following form in terms of the original variables $\boldsymbol{w}_i$:

$$\begin{bmatrix} \boldsymbol{w}_i \\ (\boldsymbol{w}_i^*)^{\mathsf{T}} \end{bmatrix} \;=\; \begin{bmatrix} \boldsymbol{w}_{i-1} \\ (\boldsymbol{w}_{i-1}^*)^{\mathsf{T}} \end{bmatrix} \;-\; \mu \begin{bmatrix} \nabla_{w^*} J(\boldsymbol{w}_{i-1}) \\ \nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}_{i-1}) \end{bmatrix} \;-\; \frac{\mu}{2}\, D\,\boldsymbol{t}_i(\boldsymbol{v}_{i-1}) \tag{3.147}$$

If we instead start from (3.117), then we would obtain

$$\begin{bmatrix} \boldsymbol{w}_i \\ (\boldsymbol{w}_i^*)^{\mathsf{T}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_{i-1} \\ (\boldsymbol{w}_{i-1}^*)^{\mathsf{T}} \end{bmatrix} - \mu \begin{bmatrix} \nabla_{w^*} J(\boldsymbol{w}_{i-1}) \\ \nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}_{i-1}) \end{bmatrix} - \mu \begin{bmatrix} \boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \\ (\boldsymbol{s}_i^*(\boldsymbol{w}_{i-1}))^{\mathsf{T}} \end{bmatrix} \tag{3.148}$$

Comparing (3.147) and (3.148) we conclude that the processes $\boldsymbol{t}_i(\cdot)$ and $\boldsymbol{s}_i(\cdot)$ are related as follows:

$$\frac{1}{2}\, D\,\boldsymbol{t}_i(\boldsymbol{v}_{i-1}) \;=\; \begin{bmatrix} \boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \\ (\boldsymbol{s}_i^*(\boldsymbol{w}_{i-1}))^{\mathsf{T}} \end{bmatrix} \tag{3.149}$$

from which, using the fact that $D^*D = 2I_{2M}$ from (B.28) in the appendix, we can solve for $\boldsymbol{t}_i(\boldsymbol{v}_{i-1})$ and find that

$$\boldsymbol{t}_i(\boldsymbol{v}_{i-1}) \;=\; 2\left[\begin{array}{c}\boldsymbol{s}_{R,i}(\boldsymbol{w}_{i-1})\\ \boldsymbol{s}_{I,i}(\boldsymbol{w}_{i-1})\end{array}\right] \tag{3.150}$$

in terms of the real and imaginary parts of the gradient noise vector:

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;\stackrel{\Delta}{=}\; \boldsymbol{s}_{R,i}(\boldsymbol{w}_{i-1}) \;+\; j\boldsymbol{s}_{I,i}(\boldsymbol{w}_{i-1}) \tag{3.151}$$

Now since $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$ satisfies conditions (3.133)–(3.134), it follows that

$$\mathbb{E}\,[\,\boldsymbol{t}_i(\boldsymbol{v}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] = 0 \tag{3.152}$$

and

$$\begin{aligned}\mathbb{E}\,[\,\|\boldsymbol{t}_i(\boldsymbol{v}_{i-1})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] &\stackrel{(3.150)}{=} 4\,\mathbb{E}\,[\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,]\\ &\stackrel{(3.138)}{\leq} 4\left(\frac{\beta}{h}\right)^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; 4\,\sigma_s^2\\ &= \beta^2\,\|\widetilde{\boldsymbol{w}}_{i-1})\|^2 \;+\; 4\,\sigma_s^2\end{aligned} \tag{3.153}$$

where we used $h = 2$ for complex data. Therefore, the gradient noise process $\boldsymbol{t}_i(\boldsymbol{v}_{i-1})$ satisfies conditions similar to (3.34) and the result of Lemma 3.1 is then immediately applicable to the $v-$domain recursion (3.144). Specifically, we know from the statement of that lemma that the stochastic gradient recursion (3.146) converges in the mean-square sense when $\mu' < 2\nu/(\delta^2+\beta^2)$, which is equivalent to (3.140). Moreover, from (3.37) we get

$$\begin{aligned}\mathbb{E}\,\|\widetilde{\boldsymbol{v}}_i\|^2 &\leq \alpha\,\mathbb{E}\,\|\widetilde{\boldsymbol{v}}_{i-1}\|^2 \;+\; (\mu')^2\,(4\sigma_s^2)\\ &= \alpha\,\mathbb{E}\,\|\widetilde{\boldsymbol{v}}_{i-1}\|^2 \;+\; \mu^2\sigma_s^2\end{aligned} \tag{3.154}$$

where

$$\begin{aligned}\alpha &= 1 - 2\nu\mu' + (\mu')^2(\delta^2+\beta^2)\\ &= 1 - \nu\mu + \frac{\mu^2}{4}(\delta^2+\beta^2)\end{aligned} \tag{3.155}$$

and, therefore, from (3.154):

$$\limsup_{i\to\infty}\;\mathbb{E}\,\|\widetilde{\boldsymbol{v}}_i\|^2 \;\leq\; \frac{\mu\sigma_s^2}{\nu - \frac{\mu}{4}(\delta^2+\beta^2)} \tag{3.156}$$

It is easy to check that the upper bound does not exceed $2\mu\sigma_s^2/\nu$ for any μ satisfying $\mu < 2\nu(\delta^2+\beta^2)$. We conclude that (3.143) holds for sufficiently small step-sizes.

□

Stability of Fourth-Order Error Moment

We can similarly extend the conclusion of Lemma 3.2 to the complex domain. For that purpose, and in a manner similar to Assumption 3.3, we assume the gradient noise process satisfies the following conditions.

Assumption 3.5 (Conditions on gradient noise: Complex case)**.** It is assumed that the first and fourth-order conditional moments of the gradient noise process satisfy the following conditions for any iterates $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{3.157}$$

$$\mathbb{E}\,\big[\,\|\boldsymbol{s}_i(\boldsymbol{w})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\big] \;\leq\; \big(\bar{\beta}/h\big)^4\,\|\boldsymbol{w}\|^4 \;+\; \bar{\sigma}_s^4 \tag{3.158}$$

almost surely, for some nonnegative coefficients $\bar{\sigma}_s^4$ and $\bar{\beta}^4$.

In a manner similar to the derivation of (3.55)–(3.56) in the real case, we can again verify that the above two conditions lead to the following forms:

$$\mathbb{E}\,[\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}] \;=\; 0 \tag{3.159}$$

$$\mathbb{E}\,\Big[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\Big] \;\leq\; \beta_4^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 + \sigma_{s4}^4 \tag{3.160}$$

in terms of the nonnegative parameters:

$$\beta_4^4 \;\triangleq\; 8\bar{\beta}^4 \tag{3.161}$$

$$\sigma_{s4}^4 \;\triangleq\; 8(\bar{\beta}/h)^4\|w^o\|^4 \;+\; \bar{\sigma}_s^4 \tag{3.162}$$

By taking expectations of (3.159)–(3.160) we obtain:

$$\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;=\; 0 \tag{3.163}$$

$$\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4 \;\leq\; (\beta_4/h)^4\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 + \sigma_{s4}^4 \tag{3.164}$$

Lemma 3.6 (Stability of fourth-order moment: Complex case)**.** Assume the conditions under Assumptions 3.1 and 3.5 on the cost function and the gradient noise process hold. Then, for sufficiently small step-sizes, it again holds that

$$\limsup_{i\to\infty}\;\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O(\mu) \tag{3.165}$$

$$\limsup_{i\to\infty}\;\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^4 \;=\; O(\mu^2) \tag{3.166}$$

Proof. We apply Lemma 3.2 to the $v-$domain recursion

$$\boldsymbol{v}_i \;=\; \boldsymbol{v}_{i-1} \;-\; \mu'\,\widehat{\nabla_{v^{\mathsf{T}}}\,J}(\boldsymbol{v}_{i-1}) \tag{3.167}$$

where $\mu' = \mu/2$ after noting that the gradient noise process $\boldsymbol{t}_i(\boldsymbol{v}_{i-1})$ satisfies a fourth-order condition of the same form as (3.60) since

$$\begin{aligned}
\mathbb{E}\,\big[\,\|\boldsymbol{t}_i(\boldsymbol{v}_{i-1})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\big] \;&=\; \mathbb{E}\,\Big[\,\big(\|\boldsymbol{t}_i(\boldsymbol{v}_{i-1})\|^2\big)^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] \\
&\overset{(3.150)}{=}\; \mathbb{E}\,\Big[\,\big(4\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\big)^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] \\
&=\; 16\mathbb{E}\,\big[\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\big] \\
&\overset{(3.164)}{\leq}\; \beta_4^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; 16\,\sigma_{s4}^4
\end{aligned} \tag{3.168}$$

using $h = 2$.

□

Decaying Step-Sizes

We now examine the convergence of the iterates $\{\boldsymbol{w}_i\}$ generated by (3.117) towards the minimizer, w^o. The lemmas that follow extend the results from the real case to the complex case with some minimal differences.

Lemma 3.7 (Almost-sure convergence: Complex case)**.** Assume the cost function $J(w)$ satisfies (3.114) and the gradient noise process satisfies the conditions in Assumption 3.4. Then, the following convergence properties hold for (3.117):

(a) If the step-size sequence $\mu(i)$ satisfies (3.92), then $\boldsymbol{w}_i$ converges almost surely to w^o, written as $\boldsymbol{w}_i \to w^o$ a.s.
(b) If the step-size sequence $\mu(i)$ satisfies (3.91), then $\boldsymbol{w}_i$ converges in the mean-square-error sense to w^o, i.e., $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \to 0$.

Proof. We apply the result of Lemma 3.3 to the $v-$domain recursion:

$$\boldsymbol{v}_i \;=\; \boldsymbol{v}_{i-1} \;-\; \mu'(i)\,\widehat{\nabla_{v^{\mathsf{T}}}\,J}(\boldsymbol{v}_{i-1}) \tag{3.169}$$

where $\mu'(i) = \mu(i)/2$.

□

Lemma 3.8 (Rates of convergence for a decaying step-size)**.** Assume the cost function $J(w)$ satisfies (3.114) and the gradient noise process satisfies the conditions in Assumption 3.4. Assume further that the step-size sequence is selected according to (3.110). Then, three convergence rates are possible depending on how the factor $\nu\tau/h$ compares to the value one. Specifically, for large enough i, it holds that:

$$\begin{cases} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;\leq\; \left(\frac{\tau^2\sigma_s^2}{\nu\tau/h-1}\right)\frac{1}{i} \;+\; o\left(\frac{1}{i}\right), & \nu\tau/h > 1 \\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O\left(\frac{\log i}{i}\right), & \nu\tau/h = 1 \\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O\left(\frac{1}{i^{\nu\tau/h}}\right), & \nu\tau/h < 1 \end{cases} \tag{3.170}$$

where $h = 2$ for complex data and $h = 1$ for real data. The fastest convergence rate occurs when $\nu\tau/h > 1$ (i.e., for large enough τ) and is in the order of $O(1/i)$.

Proof. Apply the result of Lemma 3.4 to (3.169) noting that

$$\mu'(i) = \frac{\tau/2}{i+1} \tag{3.171}$$

so that τ is replaced by $\tau/2$ and, from (3.153), σ_s^2 is replaced by $4\sigma_s^2$.

□

4

Performance of Single Agents

We established in Lemmas 3.3 and 3.7, for both cases of real and complex data, that the use of a stochastic-gradient algorithm with a decaying step-size sequence of the form $\mu(i) = \tau/(i+1)$ guarantees the almost sure convergence of the iterate $\boldsymbol{w}_i$ to w^o. However, the largest rate of convergence that is attainable under this construction is in the order of $O(1/i)$, namely, for large enough i it holds that

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O(1/i) \tag{4.1}$$

On the other hand, when a constant step-size, μ, is used, we established in Lemmas 3.1 and 3.5 that the stochastic-gradient algorithm is mean-square stable in the sense that the error variance enters a bounded region whose size is in the order of $O(\mu)$, namely, for large enough i it now holds that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O(\mu) \tag{4.2}$$

More interestingly, we showed that convergence towards this bounded region occurs at a faster *geometric* rate and is in the order of $O(\alpha^i)$ for some $0 \leq \alpha < 1$. In other words, although some degradation in steady-state performance occurs, the convergence rate is nevertheless exponential. In this chapter, we will assess the size of the fluctuations

of $\boldsymbol{w}_i$ around w^o in steady-state, as $i \to \infty$, for both cases of real and complex data. More specifically, we will determine an expression for the mean-square-deviation (MSD) of the stochastic gradient algorithm in the slow adaptation regime when μ is sufficiently small. The MSD is a useful metric that measures the size of the error variance, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, in steady-state after sufficient iterations have elapsed. We will motivate and define the MSD further ahead in relation (4.94) and subsequently determine closed-form expressions for it in (4.100) for real data and in (4.170) for complex data. In the mean time, we introduce the following terminology for ease of reference here and elsewhere in this work.

Definition 4.1 (Operating regimes)**.** The term "steady-state regime" will refer to the operation of the stochastic-gradient implementation after sufficient iterations have elapsed, i.e., as $i \to \infty$. Likewise, the term "slow adaptation regime" will refer to the operation of the stochastic-gradient implementation with a sufficiently small step-size, i.e., as $\mu \to 0$.

If we now examine expressions (4.100) and (4.170) for the MSD, we observe that it will turn out to be proportional to the step-size parameter, i.e., it is small and in the order of $O(\mu)$, as expected from (4.2). This means that adaptation with small constant step-sizes can still lead to reliable performance at an exponential convergence rate even in the presence of gradient noise, which is a reassuring result. We are also able to conclude that adaptation with constant step-sizes is useful even for stationary environments when w^o remains fixed. This is because it is generally sufficient in practice to attain an iterate $\boldsymbol{w}_i$ within some fidelity (or confidence) level from w^o in a *finite* number of iterations. As long as the MSD level is satisfactory, a stochastic-gradient algorithm will be able to attain satisfactory fidelity within a reasonable time frame. In comparison, although diminishing step-sizes ensure almost-sure convergence of $\boldsymbol{w}_i$ to w^o, they nevertheless disable tracking and can only guarantee slower than geometric convergence rates (see also [32, 190]).

We shall derive the closed-form expressions for the MSD metric, and for a related excess-risk (ER) metric, starting from Sec. 4.5. This

is an important task to pursue for various reasons [207]. First, once performance expressions are available, it becomes possible to carry out meaningful comparisons among different configurations for adaptation and learning (such as non-cooperative, centralized, and distributed implementations). Second, it also becomes possible to quantify how performance depends on the algorithm and system parameters (such as step-size, network topology, and cooperation policy); these parameters can then be optimized for enhanced performance. And third, the mean-square-error expressions define confidence levels about how well the iterate $\boldsymbol{w}_i$ approaches the global minimum, w^o. In the sequel, we shall derive an expression for the MSD by following the energy conservation technique of [6, 205, 206, 269]. For that purpose, we need to introduce an additional smoothness condition on the cost function and the gradient noise, as explained next.

4.1 Conditions on Risk Function and Noise

We consider the case of real arguments first. Thus, let $J(w) \in \mathbb{R}$ denote the real-valued cost function of a real-valued vector argument, $w \in \mathbb{R}^M$ and consider the same optimization problem (3.1):

$$w^o = \arg\min_{w} \; J(w) \tag{4.3}$$

We continue to assume that $J(w)$ is twice-differentiable and satisfies (3.2) for some positive parameters $\nu \leq \delta$, namely,

$$0 < \nu I_M \;\leq\; \nabla_w^2 J(w) \;\leq\; \delta I_M \tag{4.4}$$

Assumption 4.1 (Conditions on cost function)**.** The cost function $J(w)$ is twice-differentiable and satisfies (4.4) for some positive parameters $\nu \leq \delta$. Condition (4.4) is equivalent to requiring $J(w)$ to be $\nu-$strongly convex and for its gradient vector to be $\delta-$Lipschitz as in (2.14) and (2.17), respectively.

We established in the previous chapter the mean-square-error stability of the following stochastic-gradient recursion for seeking the minimizer

w^o in the real data case:

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \mu\,\widehat{\nabla_{w^{\mathsf{T}}} J}(\boldsymbol{w}_{i-1}), \quad i \geq 0 \tag{4.5}$$

The analysis relied on the conditions in Assumption 3.2 on the gradient noise process, $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$, which we repeat here for ease of reference. Recall from (3.25) that

$$\boldsymbol{s}_i(\boldsymbol{w}) \;\triangleq\; \widehat{\nabla_{w^{\mathsf{T}}} J}(\boldsymbol{w}) \;-\; \nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}) \tag{4.6}$$

Assumption 4.2 (Conditions on gradient noise)**.** It is assumed that the first and second-order conditional moments of the gradient noise process satisfy the following conditions for any $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{4.7}$$

$$\mathbb{E}\,\big[\,\|\boldsymbol{s}_i(\boldsymbol{w})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\big] \;\leq\; \bar{\beta}^2\,\|\boldsymbol{w}\|^2 \;+\; \bar{\sigma}_s^2 \tag{4.8}$$

almost surely, for some nonnegative scalars $\bar{\beta}^2$ and $\bar{\sigma}_s^2$. These conditions were shown in (3.31)–(3.32) to imply that the gradient noise process satisfies for any $\boldsymbol{w}_{i-1} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{4.9}$$

$$\mathbb{E}\,\big[\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\big] \;\leq\; \beta^2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \sigma_s^2 \tag{4.10}$$

almost surely, for some nonnegative scalars β^2 and σ_s^2, and where $\widetilde{\boldsymbol{w}}_{i-1} = w^o - \boldsymbol{w}_{i-1}$.

Now, in order to pursue a closed form expression for the MSD of the algorithm, we need to introduce two smoothness conditions: one condition is on the cost function and the other condition is on the covariance matrix of the gradient noise process.

For any $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$, we let

$$R_{s,i}(\boldsymbol{w}) \;\triangleq\; \mathbb{E}\,\Big[\,\boldsymbol{s}_i(\boldsymbol{w})\boldsymbol{s}_i^{\mathsf{T}}(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] \tag{4.11}$$

denote the conditional second-order moment of the gradient noise process, which generally depends on i because the statistical distribution of $\boldsymbol{s}_i(\boldsymbol{w})$ can be iteration-dependent. Note that $R_{s,i}(\boldsymbol{w})$ is a random

quantity since it depends on the random iterate $\boldsymbol{w}$. We assume that, in the limit, this covariance matrix tends to a constant value when evaluated at w^o and we denote the limit by

$$R_s \triangleq \lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{s}_i(w^o)\boldsymbol{s}_i^{\mathsf{T}}(w^o) \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \tag{4.12}$$

We sometimes refer to the term $\boldsymbol{s}_i(w^o)$ as the *absolute* noise component.

Example 4.1 (Gradient noise for mean-square-error costs)**.** Let us reconsider the scenario studied in Example 3.3, which dealt with mean-square-error costs of the form $J(w) = \mathbb{E}\,(\boldsymbol{d}(i) - \boldsymbol{u}_i w)^2$. From expression (3.19) we know that

$$\boldsymbol{s}_i(w^o) = -2\boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{v}(i) \tag{4.13}$$

$$R_s = 4\sigma_v^2 R_u \;\equiv\; R_{s,i}(w^o), \quad \text{for all } i \tag{4.14}$$

Moreover, from expression (3.19) for $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$, and from the conditions on the random processes $\{\boldsymbol{u}_i, \boldsymbol{v}(i)\}$ in Example 3.1, we have that

$$R_{s,i}(\boldsymbol{w}_{i-1}) \;=\; 4\mathbb{E}\left\{(R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i)\widetilde{\boldsymbol{w}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^{\mathsf{T}}(R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i)\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right\} \;+\; 4\sigma_v^2 R_u \tag{4.15}$$

Now since $\boldsymbol{u}_i$ and $\widetilde{\boldsymbol{w}}_{i-1}$ are independent of each other:

$$\begin{aligned}
&\|R_{s,i}(\boldsymbol{w}_{i-1}) - R_{s,i}(w^o)\| \\
&\quad= 4\left\|\mathbb{E}\left\{(R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i)\widetilde{\boldsymbol{w}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^{\mathsf{T}}(R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i)\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right\}\right\| \\
&\quad\le 4\mathbb{E}\left\{\left\|(R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i)\widetilde{\boldsymbol{w}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^{\mathsf{T}}(R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i)\right\|\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right\} \\
&\quad\le 4\mathbb{E}\left\{\left\|R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i\right\|^2 \|\widetilde{\boldsymbol{w}}_{i-1}\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right\} \\
&\quad= 4\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\left(\mathbb{E}\left\|R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i\right\|^2\right) \\
&\quad= 4c\|\widetilde{\boldsymbol{w}}_{i-1}\|^2
\end{aligned} \tag{4.16}$$

with the constant $c = \mathbb{E}\,\|R_u - \boldsymbol{u}_i^{\mathsf{T}}\boldsymbol{u}_i\|^2$. It follows that, by taking expectations,

$$\mathbb{E}\,\|R_{s,i}(\boldsymbol{w}_{i-1}) - R_{s,i}(w^o)\| \;\le\; 4c\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \tag{4.17}$$

■

The two smoothness conditions that are needed for the subsequent evaluation of the MSD performance of the stochastic-gradient algorithm (4.5) are the following [66, 71, 278].

Assumption 4.3 (Smoothness conditions)**.** It is assumed that the Hessian matrix of the cost function, $J(w)$, and the noise covariance matrix defined by (4.11) are locally Lipschitz continuous in a small neighborhood around $w = w^o$ in the following manner:

$$\begin{aligned} \left\|\nabla_w^2 J(w^o + \Delta w) - \nabla_w^2 J(w^o)\right\| &\leq \kappa_1 \|\Delta w\| \quad (4.18) \\ \|R_{s,i}(w^o + \Delta w) - R_{s,i}(w^o)\| &\leq \kappa_2 \|\Delta w\|^\gamma \quad (4.19) \end{aligned}$$

for small perturbations $\|\Delta w\| \leq \epsilon$ and for some constants $\kappa_1 \geq 0$, $\kappa_2 \geq 0$, and exponent $0 < \gamma \leq 4$.

Observe from (4.17) that for mean-square-error costs, the Lipschitz condition (4.19) is satisfied with $\gamma = 2$. Likewise, for mean-square-error costs, the first condition (4.18) is automatically satisfied since the Hessian matrices of quadratic costs are constant and independent of w.

Although conditions (4.18)–(4.19) are required to hold only locally in the proximity of $w = w^o$, they actually turn out to imply that similar bounds hold more globally. For example, using result (E.30) from the appendix, it can be verified that condition (4.18) translates into a global Lipschitz property relative to the minimizer w^o, i.e., it will also hold that [278]:

$$\|\nabla_w^2 J(w) - \nabla_w^2 J(w^o)\| \leq \kappa_1' \|w - w^o\| \quad (4.20)$$

for all w and for some constant $\kappa_1' \geq 0$.

A similar conclusion follows from (4.19). To see that, let us consider any $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$ such that $\|w^o - \boldsymbol{w}\| > \epsilon$. This condition corresponds to a situation where the perturbation $\Delta\boldsymbol{w}$ in (4.19) lies outside the disc of radius ϵ. Nevertheless, we can still argue that an upper bound similar to (4.19) continues to hold, albeit with some adjustment [71] — see expression (4.24). To arrive at this expression, we start by using the triangle inequality of norms to note that

$$\|R_{s,i}(\boldsymbol{w}) - R_{s,i}(w^o)\| \leq \|R_{s,i}(\boldsymbol{w})\| + \|R_{s,i}(w^o)\| \quad (4.21)$$

Using the property that $\|A\| \leq \mathrm{Tr}(A)$ for any symmetric nonnegative-definite matrix A (since the trace is the sum of the eigenvalues of the matrix and the 2−induced norm is its largest eigenvalue), we can bound each term on the right-hand side of (4.21) as follows:

$$\begin{aligned}
\|R_{s,i}(\boldsymbol{w})\| &\leq \mathrm{Tr}\left[R_{s,i}(\boldsymbol{w})\right] \\
&= \mathrm{Tr}\left[\mathbb{E}\left\{\boldsymbol{s}_i(\boldsymbol{w})\boldsymbol{s}_i^{\mathsf{T}}(\boldsymbol{w}) \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right\}\right] \\
&= \mathbb{E}\left\{\mathrm{Tr}\left(\boldsymbol{s}_i(\boldsymbol{w})\boldsymbol{s}_i^{\mathsf{T}}(\boldsymbol{w})\right) \;|\; \boldsymbol{\mathcal{F}}_{i-1}\right\} \\
&= \mathbb{E}\left[\|\boldsymbol{s}_i(\boldsymbol{w})\|^2 \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \\
&\overset{(4.10)}{\leq} \beta^2\|w^o - \boldsymbol{w}\|^2 + \sigma_s^2
\end{aligned} \tag{4.22}$$

By setting $\boldsymbol{w} = w^o$ we also conclude that $\|R_{s,i}(w^o)\| \leq \sigma_s^2$. Substituting into (4.21) we get

$$\begin{aligned}
\|R_{s,i}(\boldsymbol{w}) - R_{s,i}(w^o)\| &\leq \beta^2\|w^o - \boldsymbol{w}\|^2 + 2\sigma_s^2 \\
&\overset{\text{(a)}}{\leq} \beta^2\|w^o - \boldsymbol{w}\|^2 + 2\sigma_s^2\left(\frac{\|w^o - \boldsymbol{w}\|^2}{\epsilon^2}\right) \\
&= \left(\beta^2 + \frac{2\sigma_s^2}{\epsilon^2}\right)\|w^o - \boldsymbol{w}\|^2 \\
&\triangleq \kappa_3\|w^o - \boldsymbol{w}\|^2
\end{aligned} \tag{4.23}$$

for some nonnegative constant κ_3 and where in step (a) we used the fact that $\|w^o - \boldsymbol{w}\| > \epsilon$. Combining this result with the localized assumption (4.19) we conclude that the conditional noise covariance matrix satisfies more globally a condition of the following form for *any* $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\begin{aligned}
\|R_{s,i}(\boldsymbol{w}) - R_{s,i}(w^o)\| &\leq \max\left\{\kappa_2\,\|\widetilde{\boldsymbol{w}}\|^{\gamma},\ \kappa_3\,\|\widetilde{\boldsymbol{w}}\|^2\right\} \\
&\leq \kappa_2\,\|\widetilde{\boldsymbol{w}}\|^{\gamma} + \kappa_3\,\|\widetilde{\boldsymbol{w}}\|^2
\end{aligned} \tag{4.24}$$

where $\widetilde{\boldsymbol{w}} = w^o - \boldsymbol{w}$.

One useful conclusion that follows from the smoothness condition (4.19) and from (4.24) is that, after sufficient iterations, we can express the covariance matrix of the gradient noise process in terms of the same limiting value R_s defined by (4.12) for the absolute noise component.

This fact is established next and will be employed later in the proof of Theorem 4.7.

Lemma 4.1 (Limiting second-order moment of gradient noise: Real case)**.** Under the smoothness condition (4.19), and for sufficiently small step-sizes, it holds for $i \gg 1$ that:[1]

$$\mathbb{E}\, \boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}} \;=\; R_s \;+\; O\left(\mu^{\min\{1,\frac{\gamma}{2}\}}\right) \tag{4.25}$$

where $0 < \gamma \leq 4$ is from (4.19) and R_s is defined by (4.12). Consequently, it holds for $i \gg 1$ that the trace of the covariance matrix satisfies:

$$\mathrm{Tr}(R_s) - b_o \;\leq\; \mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 \;\leq\; \mathrm{Tr}(R_s) + b_o \tag{4.26}$$

for some nonnegative value $b_o = O\left(\mu^{\min\{1,\frac{\gamma}{2}\}}\right)$.

Proof. By adding and subtracting the same term, we have

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}} \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \;&=\; \mathbb{E}\left[\boldsymbol{s}_i(w^o)\,(\boldsymbol{s}_i(w^o))^{\mathsf{T}} \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] + \\
&\quad\; \mathbb{E}\left[\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}} \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] - \\
&\quad\; \mathbb{E}\left[\boldsymbol{s}_i(w^o)\,(\boldsymbol{s}_i(w^o))^{\mathsf{T}} \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \\
&\overset{(4.11)}{=}\; \mathbb{E}\left[\boldsymbol{s}_i(w^o)\,(\boldsymbol{s}_i(w^o))^{\mathsf{T}} \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] + \\
&\quad\; R_{s,i}(\boldsymbol{w}_{i-1}) - R_{s,i}(w^o)
\end{aligned} \tag{4.27}$$

so that by subtracting the covariance matrix R_s defined by (4.12) from both sides, and computing expectations, we get:

$$\begin{aligned}
\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}} - R_s \;&=\; \mathbb{E}\left(\mathbb{E}\left[\boldsymbol{s}_i(w^o)\,(\boldsymbol{s}_i(w^o))^{\mathsf{T}} \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] - R_s\right) + \\
&\quad\; \mathbb{E}\,(R_{s,i}(\boldsymbol{w}_{i-1}) - R_{s,i}(w^o))
\end{aligned} \tag{4.28}$$

It then follows from the triangle inequality of norms, and from Jensen's inequality (F.29) in the appendix, that:

[1] The notation $X = O(\mu)$ for a matrix X signifies that the magnitude of the individual entries of X are $O(\mu)$ or $\|X\| = O(\mu)$.

$$\begin{aligned}
&\Big\|\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}} - R_s\Big\| \\
&\qquad \leq \Big\|\mathbb{E}\,\Big(\mathbb{E}\,\Big[\boldsymbol{s}_i(w^o)\,(\boldsymbol{s}_i(w^o))^{\mathsf{T}}\;|\;\boldsymbol{\mathcal{F}}_{i-1}\Big] - R_s\Big)\Big\| + \\
&\qquad\qquad \|\mathbb{E}\,(R_{s,i}(\boldsymbol{w}_{i-1}) - R_{s,i}(w^o))\| \\
&\qquad \overset{(F.29)}{\leq} \mathbb{E}\,\Big\|\mathbb{E}\,\Big[\boldsymbol{s}_i(w^o)\,(\boldsymbol{s}_i(w^o))^{\mathsf{T}}\;|\;\boldsymbol{\mathcal{F}}_{i-1}\Big] - R_s\Big\| + \\
&\qquad\qquad \mathbb{E}\,\|R_{s,i}(\boldsymbol{w}_{i-1}) - R_{s,i}(w^o)\|
\end{aligned} \tag{4.29}$$

where the notation $\|X\|$ denotes the 2−induced norm of its matrix argument, X. If we now compute the limit superior of both sides, and recall definition (4.12), we get

$$\begin{aligned}
&\limsup_{i\to\infty}\Big\|\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}} - R_s\Big\| \\
&\qquad\qquad \leq \limsup_{i\to\infty}\mathbb{E}\,\|R_{s,i}(\boldsymbol{w}_{i-1}) - R_{s,i}(w^o)\|
\end{aligned} \tag{4.30}$$

The limit superior on the right-hand side can be evaluated by calling upon (4.24) to get:

$$\begin{aligned}
&\limsup_{i\to\infty}\mathbb{E}\,\|R_{s,i}(\boldsymbol{w}_{i-1}) - R_{s,i}(w^o)\| \\
&\qquad \leq \limsup_{i\to\infty}\mathbb{E}\,\left\{\kappa_2\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^{\gamma} \;+\; \kappa_3\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\right\} \\
&\qquad \leq \limsup_{i\to\infty}\left\{\kappa_2\,\mathbb{E}\,\left(\|\widetilde{\boldsymbol{w}}_{i-1}\|^4\right)^{\gamma/4} \;+\; \kappa_3\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\right\} \\
&\qquad \overset{(a)}{\leq} \limsup_{i\to\infty}\left\{\kappa_2\,\left(\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4\right)^{\gamma/4} \;+\; \kappa_3\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\right\} \\
&\qquad \overset{(3.39)}{=} O(\mu^{\gamma'/2})
\end{aligned} \tag{4.31}$$

where in step (a) we applied Jensen's inequality (F.30) to the function $f(x) = x^{\gamma/4}$; this function is concave over $x \geq 0$ for $\gamma \in (0, 4]$. Moreover, in the last step we called upon results (3.39) and (3.67), namely, that the second and fourth-order moments of $\widetilde{\boldsymbol{w}}_{i-1}$ are asymptotically bounded by $O(\mu)$ and $O(\mu^2)$, respectively. Accordingly, the exponent γ' in the last step is given by

$$\gamma' \;\triangleq\; \min\{\gamma,\, 2\} \tag{4.32}$$

since $O(\mu^{\gamma/2})$ dominates $O(\mu)$ for values of $\gamma \in (0, 2]$ and $O(\mu)$ dominates $O(\mu^{\gamma/2})$ for values of $\gamma \in [2, 4]$. Substituting (4.31) into (4.30) we conclude that

$$\limsup_{i\to\infty}\Big\|\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}} - R_s\Big\| = O(\mu^{\gamma'/2}) \tag{4.33}$$

If we denote the difference between R_s and the covariance matrix $\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}}$ by Δ_i, then result (4.33) implies that, for $i \gg 1$, we have $\|\Delta_i\| = O(\mu^{\gamma'/2})$ and we arrive at (4.25). Moreover, since for any square matrix X, it can be verified that $|\mathrm{Tr}(X)| \leq c\,\|X\|$, for some constant c that is independent of γ', we also conclude from (4.33) that

$$\limsup_{i\to\infty} \big|\,\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 - \mathrm{Tr}(R_s)\,\big| \;=\; O(\mu^{\gamma'/2}) \;\;\triangleq\;\; b_1 \tag{4.34}$$

in terms of the absolute value of the difference. We are denoting the value of the limit superior by the nonnegative number b_1; we know from (4.34) that $b_1 = O(\mu^{\gamma'/2})$. The above relation then implies that, given $\epsilon > 0$, there exists an I_o large enough such that for all $i > I_o$ it holds that

$$\big|\,\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 - \mathrm{Tr}(R_s)\,\big| \;\leq\; b_1 + \epsilon \tag{4.35}$$

If we select $\epsilon = O(\mu^{\gamma'/2})$ and introduce the sum $b_o = b_1 + \epsilon$, then we arrive at the desired result (4.26).

□

4.2 Stability of First-Order Error Moment

Using the Lipschitz property (4.20), we can now examine the mean stability of the error vector, $\widetilde{\boldsymbol{w}}_i$, and show that the limit superior of $\|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i\|$ is bounded by $O(\mu)$.

Indeed, using the fact that $(\mathbb{E}\,\boldsymbol{a})^2 \leq \mathbb{E}\,\boldsymbol{a}^2$, for any real-valued random variable $\boldsymbol{a}$, we note that we may conclude from (3.39) that

$$\limsup_{i\to\infty}\; \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\| \;=\; O(\mu^{1/2}) \tag{4.36}$$

However, a tighter bound is possible with $\mu^{1/2}$ replaced by μ by appealing to (4.20) and bounding the limiting value of $\|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i\|$.

Let us reconsider recursion (3.42), namely,

$$\widetilde{\boldsymbol{w}}_i \;=\; (I_M - \mu\boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1} \;+\; \mu\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \tag{4.37}$$

where

$$\boldsymbol{H}_{i-1} \;\triangleq\; \int_0^1 \nabla_w^2\, J(w^o - t\widetilde{\boldsymbol{w}}_{i-1})dt \tag{4.38}$$

We introduce the deviation matrix

$$\widetilde{\boldsymbol{H}}_{i-1} \;\triangleq\; H - \boldsymbol{H}_{i-1} \tag{4.39}$$

where the constant symmetric and positive-definite matrix H is defined as the value of the Hessian matrix at the minimizer w^o:

$$H \triangleq \nabla_w^2 J(w^o) \tag{4.40}$$

Substituting (4.39) into (4.37) gives

$$\widetilde{\boldsymbol{w}}_i = (I_M - \mu H)\widetilde{\boldsymbol{w}}_{i-1} + \mu\, \boldsymbol{s}_i(\boldsymbol{w}_{i-1}) + \mu \boldsymbol{c}_{i-1} \tag{4.41}$$

in terms of the perturbation term

$$\boldsymbol{c}_{i-1} \triangleq \widetilde{\boldsymbol{H}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1} \tag{4.42}$$

Lemma 4.2 (Mean-error stability: Real case)**.** Assume the requirements under Assumptions 4.1 and 4.2 and condition (4.18) on the cost function and the gradient noise process hold. Then, for sufficiently small step-sizes it holds that

$$\limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i\| = O(\mu) \tag{4.43}$$

Proof. Conditioning both sides of (4.41) on $\boldsymbol{\mathcal{F}}_{i-1}$, and using the fact that $\mathbb{E}\,[\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}] = 0$, we conclude that

$$\mathbb{E}\,[\widetilde{\boldsymbol{w}}_i\,|\,\boldsymbol{\mathcal{F}}_{i-1}] = (I_M - \mu H)\,\widetilde{\boldsymbol{w}}_{i-1} + \mu \boldsymbol{c}_{i-1} \tag{4.44}$$

Taking expectations again we arrive at the mean recursion

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i = (I_M - \mu H)\,\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1} + \mu \mathbb{E}\,\boldsymbol{c}_{i-1} \tag{4.45}$$

The limit superior of the right-most expectation is bounded by $O(\mu^2)$ for the following reason. Note that

$$\begin{aligned}
\|\boldsymbol{c}_{i-1}\| &\overset{(4.42)}{\leq} \|\widetilde{\boldsymbol{H}}_{i-1}\|\,\|\widetilde{\boldsymbol{w}}_{i-1}\| \\
&\overset{(4.38)}{\leq} \|\widetilde{\boldsymbol{w}}_{i-1}\| \int_0^1 \big\|\nabla_w^2 J(w^o - t\widetilde{\boldsymbol{w}}_{i-1}) - \nabla_w^2 J(w^o)\big\|\, dt \\
&\overset{(4.20)}{\leq} \kappa_1' \,\|\widetilde{\boldsymbol{w}}_{i-1}\| \int_0^1 \|t\widetilde{\boldsymbol{w}}_{i-1}\| dt \\
&= \frac{\kappa_1'}{2}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2
\end{aligned} \tag{4.46}$$

Thus, using (3.39), we conclude that the mean-norm value of the correction term converges asymptotically to the region:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\boldsymbol{c}_{i-1}\| \;=\; O(\mu) \tag{4.47}$$

Now the matrix $(I_M - \mu H)$ is symmetric so that its $2-$induced norm agrees with its spectral radius:

$$\|I_M - \mu H\| \;=\; \rho(I_M - \mu H) \tag{4.48}$$

Moreover, for sufficiently small step-sizes $\mu \ll 1$, it holds that this spectral radius is strictly smaller than one and given by

$$\rho(I_M - \mu H) \;=\; 1 - \mu\lambda_{\min}(H) \tag{4.49}$$

It then follows from (4.45) that

$$\begin{aligned} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i\| &\leq \|I_M - \mu H\|\,\|\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}\| \;+\; \mu\|\mathbb{E}\,\boldsymbol{c}_{i-1}\| \\ &\leq (1 - \mu\lambda_{\min}(H))\|\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}\| \;+\; \mu\mathbb{E}\,\|\boldsymbol{c}_{i-1}\| \end{aligned} \tag{4.50}$$

so that

$$\begin{aligned} \limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i\| &\leq \frac{1}{1-(1-\mu\lambda_{\min}(H))}\left(\limsup_{i\to\infty} \mu\mathbb{E}\,\|\boldsymbol{c}_{i-1}\|\right) \\ &= O(\mu) \end{aligned} \tag{4.51}$$

as claimed.

□

4.3 Long-Term Error Dynamics

Continuing with model (4.41), we can use it to motivate a useful long-term model for the evolution of the error vector $\widetilde{\boldsymbol{w}}_i$ after sufficient iterations, i.e., for $i \gg 1$. For this purpose, we note first that we can deduce from (4.47) that $\|\boldsymbol{c}_{i-1}\| = O(\mu)$ asymptotically with *high probability*. Indeed, let us introduce the nonnegative random variable $\boldsymbol{u} = \|\boldsymbol{c}_{i-1}\|$ and let us recall Markov's inequality [89, 91, 186], which states that for any *nonnegative* random variable $\boldsymbol{u}$ and $\xi > 0$ it holds that

$$\text{Prob}(\boldsymbol{u} \geq \xi) \;\leq\; \mathbb{E}\,\boldsymbol{u}/\xi \tag{4.52}$$

That is, the probability of the event $\boldsymbol{u} \geq \xi$ is upper bounded by a term that is proportional to $\mathbb{E}\,\boldsymbol{u}$. We employ this result as follows. Let

$r_c = n\mu$, for any constant integer $n \geq 1$ that we are free to choose. We then conclude from (4.47) and (4.52) that for $i \gg 1$:

$$\begin{aligned} \text{Prob}\left(\|\boldsymbol{c}_{i-1}\| < r_c\right) &= 1 - \text{Prob}\left(\|\boldsymbol{c}_{i-1}\| \geq r_c\right) \\ &\geq 1 - \left(\mathbb{E}\,\|\boldsymbol{c}_{i-1}\|/r_c\right) \\ &\overset{(4.47)}{\geq} 1 - O\left(1/n\right) \end{aligned} \tag{4.53}$$

where the term $O(1/n)$ is independent of μ. This result shows that the probability of having $\|\boldsymbol{c}_{i-1}\|$ bounded by r_c can be made arbitrarily close to one by selecting a large enough value for n. Once the value for n has been fixed to meet a desired confidence level, then $r_c = O(\mu)$.

Referring to recursion (4.41), this analysis suggests that we can assess its mean-square performance by examining the following long-term model, which holds with high probability after sufficient iterations:

$$\widetilde{\boldsymbol{w}}_i \;=\; (I_M - \mu H)\widetilde{\boldsymbol{w}}_{i-1} \;+\; \mu\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}), \qquad i \gg 1 \tag{4.54}$$

In this model, the perturbation term $\mu\boldsymbol{c}_{i-1}$ that appears in (4.41) is removed. We may also consider an alternative long-term model where $\mu\boldsymbol{c}_{i-1}$ is instead replaced by a constant driving term in the order of $O(\mu^2)$. However, the conclusions that will follow about the performance of the original recursion (4.37) will be the same whether we remove $\mu\boldsymbol{c}_{i-1}$ altogether or replace it by $O(\mu^2)$. We therefore continue our analysis by using model (4.54). Obviously, the iterates $\{\widetilde{\boldsymbol{w}}_i\}$ that are generated by (4.54) are generally different from the iterates that are generated by the original recursion (4.37). To highlight this fact, we rewrite the long-term model (4.54) more explicitly as follows.

Lemma 4.3 (Long-term error dynamics)**.** Assume the requirements under Assumptions 4.1 and 4.2 and condition (4.18) on the cost function and the gradient noise process hold. After sufficient iterations, $i \gg 1$, the error dynamics of the stochastic-gradient algorithm (4.5) is well-approximated by the following model (as confirmed by future result (4.70)):

$$\widetilde{\boldsymbol{w}}'_i \;=\; (I_M - \mu H)\widetilde{\boldsymbol{w}}'_{i-1} \;+\; \mu\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \tag{4.55}$$

with the iterates denoted by $\widetilde{\boldsymbol{w}}'_i$ using the prime notation.

Note that the driving process $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$ in (4.55) continues to be the same gradient noise process from the original recursion (4.37) and is evaluated at $\boldsymbol{w}_{i-1}$. We can view the long-term model (4.55) as a dynamic recursion that is fed by the gradient noise sequence, $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$. Therefore, assuming both the original system (4.37) and the long-term model (4.55) are launched from the same initial conditions, we observe by iterating (4.55) that $\widetilde{\boldsymbol{w}}_i'$ will still be determined by the past history of the iterates $\{\boldsymbol{w}_j,\, j \leq i-1\}$ through its dependence on the gradient noise process $\{\boldsymbol{s}_j(\boldsymbol{w}_{j-1}),\, j \leq i\}$. Therefore, it also holds that $\widetilde{\boldsymbol{w}}_i' \in \boldsymbol{\mathcal{F}}_{i-1}$.

Now working with recursion (4.55) is much more tractable because its dynamics is driven by the constant matrix H as opposed to the random matrix $\boldsymbol{H}_{i-1}$ in the original error recursion (4.37). We shall therefore follow the following route to evaluate the MSD of the stochastic-gradient algorithm (4.5). We shall work with the long-term model (4.55) and evaluate its MSD. Subsequently, we will argue that, under a condition on the fourth-order moment of the gradient noise process, this MSD value is within $O(\mu^{3/2})$ from the true MSD expression that would result had we worked directly with the original error recursion (4.37) without the approximation of ignoring $\mu\boldsymbol{c}_{i-1}$ in the long-term. Therefore, the MSD expression that we shall derive based on the long-term model (4.55) will provide an accurate representation for the MSD of the original stochastic-gradient algorithm to first-order in μ.

We already know from the result of Lemma 3.1 that the original error recursion (4.37) is mean-square stable in the sense that $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ tends asymptotically to a region that is bounded by $O(\mu)$. We now verify that the long-term model (4.55) is also mean-square stable.

Lemma 4.4 (Mean-square stability of long-term model)**.** Assume the conditions under Assumptions 4.1 and 4.2 on the cost function and the gradient noise process hold. Then, for sufficiently small step-sizes, the iterate that is generated by the long-term model (4.55) satisfies:

$$\limsup_{i\to\infty}\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i'\|^2 \;=\; O(\mu) \tag{4.56}$$

Proof. Note first that since $\widetilde{\boldsymbol{w}}_{i-1}' \in \boldsymbol{\mathcal{F}}_{i-1}$ and $\mathbb{E}\,[\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}] = 0$, we conclude from (4.55) that

$$\mathbb{E}\,\left[\,\|\widetilde{\boldsymbol{w}}_i'\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] = \left\|(I_M-\mu H)\widetilde{\boldsymbol{w}}_{i-1}'\right\|^2 \;+\; \mu^2\,\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\|^2 \quad (4.57)$$

Taking expectations again, we get

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i'\|^2 \;=\; \mathbb{E}\,\left\|(I_M-\mu H)\widetilde{\boldsymbol{w}}_{i-1}'\right\|^2 \;+\; \mu^2\,\mathbb{E}\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2 \quad (4.58)$$

Using an argument similar to (2.33) and assuming sufficiently small μ such that $\mu < \nu/\delta^2$, we have:

$$\|I_M-\mu H\|^2 \;\;\leq\;\; 1-2\mu\nu+\mu^2\delta^2 \;\leq\; 1-\mu\nu \quad (4.59)$$

and, therefore,

$$\begin{aligned}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i'\|^2 &\overset{(4.10)}{\leq} \|I_M-\mu H\|^2\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}'\|^2+\mu^2\,\left[\beta^2\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\sigma_s^2\right]\\ &\overset{(4.59)}{\leq} (1-\mu\nu)\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}'\|^2+\mu^2\,\beta^2\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\mu^2\sigma_s^2 \end{aligned} \quad (4.60)$$

We already know from (3.39) that sufficiently small step-sizes ensure the convergence of $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2$ towards a region that is bounded by $O(\mu)$. It follows that

$$\begin{aligned}\limsup_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i'\|^2 &\leq \frac{1}{1-(1-\mu\nu)}\left(\mu^2\,\beta^2\cdot O(\mu)\;+\;\mu^2\sigma_s^2\right)\\ &= O(\mu)\end{aligned} \quad (4.61)$$

We therefore conclude that (4.56) holds for sufficiently small step-sizes. □

We can also establish the stability of the mean error for the long-term model (4.55) under the Lipschitz property (4.20).

Lemma 4.5 (Mean stability of long-term model)**.** Assume the requirements under Assumptions 4.1 and 4.2 and condition (4.20) on the cost function and the gradient noise process hold. Then, for sufficiently small step-sizes, the iterates of the long-term model (4.55) are asymptotically zero mean:

$$\lim_{i\to\infty}\;\mathbb{E}\,\widetilde{\boldsymbol{w}}_i' \;=\; 0 \quad (4.62)$$

Proof. The derivation is similar to the argument used to conclude the proof of Lemma 4.2. Specifically, we first use (4.55) to obtain

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i' \;=\; (I_M - \mu H)\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}' \tag{4.63}$$

And since $I_M - \mu H$ is a stable matrix for $\mu \ll 1$, we conclude that (4.62) holds.

□

4.4 Size of Approximation Error

We can also examine how close the trajectories of the original error recursion (4.37) and the long-term model (4.55) are to each other. We reproduce both recursions below, with the state variable for the long-term model denoted by $\widetilde{\boldsymbol{w}}_i'$, namely,

$$\widetilde{\boldsymbol{w}}_i \;=\; (I_M - \mu \boldsymbol{H}_{i-1})\widetilde{\boldsymbol{w}}_{i-1} \;+\; \mu\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \tag{4.64}$$

$$\widetilde{\boldsymbol{w}}_i' \;=\; (I_M - \mu H)\widetilde{\boldsymbol{w}}_{i-1}' \;+\; \mu\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \tag{4.65}$$

Observe that both models are driven by the *same* gradient noise process; in this way, the evolution of the long-term model is coupled to the evolution of the original recursion (but not the other way around). The closeness of the trajectories of both recursions is established under the fourth-order condition (3.50) on the gradient noise process, which we repeat below for ease of reference.

Assumption 4.4 (Conditions on gradient noise)**.** It is assumed that the first and fourth-order conditional moments of the gradient noise process satisfy the following conditions for any iterates $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{4.66}$$

$$\mathbb{E}\,\big[\,\|\boldsymbol{s}_i(\boldsymbol{w})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\big] \;\leq\; \bar{\beta}^4\,\|\boldsymbol{w}\|^4 \;+\; \bar{\sigma}_s^4 \tag{4.67}$$

almost surely, for some nonnegative coefficients $\bar{\sigma}_s^4$ and $\bar{\beta}^4$. These conditions were shown in (3.55)–(3.56) to imply that the gradient noise process also satisfies for any $\boldsymbol{w}_{i-1} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\mathbb{E}\,[\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,|\,\boldsymbol{\mathcal{F}}_{i-1}] \;=\; 0 \tag{4.68}$$

$$\mathbb{E}\,\big[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\big] \;\leq\; \beta_4^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 + \sigma_{s4}^4 \tag{4.69}$$

almost surely, for some nonnegative coefficients β_4^4 and σ_{s4}^4.

The next statement establishes two useful facts: (a) it shows that the mean-square difference between the trajectories $\{\widetilde{\boldsymbol{w}}_i,\, \widetilde{\boldsymbol{w}}'_i\}$ is asymptotically bounded by $O(\mu^2)$, and (b) it shows that the MSD values for the original model (4.64) and the long-term model (4.55) are within $O(\mu^{3/2})$ from each other.

Lemma 4.6 (Performance error is $O(\mu^{3/2})$)**.** Assume the conditions under Assumptions 4.1, 4.3, and 4.4 on the cost function and the gradient noise process are satisfied. It then holds that, for sufficiently small step-sizes:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i - \widetilde{\boldsymbol{w}}'_i\|^2 \;=\; O(\mu^2) \tag{4.70}$$

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; \limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_i\|^2 \;+\; O(\mu^{3/2}) \tag{4.71}$$

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_H^2 \;=\; \limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_i\|_H^2 \;+\; O(\mu^{3/2}) \tag{4.72}$$

where the last line involves weighted norms of $\{\widetilde{\boldsymbol{w}}'_i, \widetilde{\boldsymbol{w}}_i\}$ with weighting matrix equal to H.

Proof. Subtracting recursions (4.64) and (4.65) we get

$$\widetilde{\boldsymbol{w}}_i - \widetilde{\boldsymbol{w}}'_i = (I_M - \mu H)(\widetilde{\boldsymbol{w}}_{i-1} - \widetilde{\boldsymbol{w}}'_{i-1}) \;+\; \mu \boldsymbol{c}_{i-1} \tag{4.73}$$

where, from (4.20), $\boldsymbol{c}_{i-1} = \widetilde{\boldsymbol{H}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}$. Using again an argument similar to (2.33) and assuming sufficiently small μ such that $\mu < \nu/\delta^2$, we have:

$$\begin{aligned} \|I_M - \mu H\|^2 &\le 1 - 2\mu\nu + \mu^2\delta^2 \\ &\le 1 - \mu\nu \\ &\le 1 - \mu\nu + \frac{\mu^2\nu^2}{4} \\ &= \left(1 - \frac{\mu\nu}{2}\right)^2 \end{aligned} \tag{4.74}$$

We now call upon Jensen's inequality (F.26) from the appendix and apply it to the convex function $f(x) = \|x\|^2$. Indeed, selecting

$$t = \mu\nu/2 \tag{4.75}$$

and for any small μ that ensures $0 < t < 1$, we can write

$$
\begin{aligned}
&\left\|(I_M-\mu H)(\widetilde{\boldsymbol{w}}_{i-1}-\widetilde{\boldsymbol{w}}'_{i-1})\;+\;\mu\boldsymbol{c}_{i-1}\right\|^2\\
&\quad=\quad\left\|(1-t)\frac{1}{1-t}(I_M-\mu H)(\widetilde{\boldsymbol{w}}_{i-1}-\widetilde{\boldsymbol{w}}'_{i-1})\;+\;t\frac{1}{t}\left(\mu\boldsymbol{c}_{i-1}\right)\right\|^2\\
&\quad\le\quad(1-t)\left\|\frac{1}{1-t}(I_M-\mu H)\right\|^2\|\widetilde{\boldsymbol{w}}_{i-1}-\widetilde{\boldsymbol{w}}'_{i-1}\|^2+t\left\|\frac{1}{t}\left(\mu\boldsymbol{c}_{i-1}\right)\right\|^2\\
&\quad\overset{(4.74)}{\le}\quad\frac{1}{1-t}\left(1-\frac{\mu\nu}{2}\right)^2\|\widetilde{\boldsymbol{w}}_{i-1}-\widetilde{\boldsymbol{w}}'_{i-1}\|^2\;+\;\frac{1}{t}\|\mu\boldsymbol{c}_{i-1}\|^2\\
&\quad\overset{(4.75)}{=}\quad\left(1-\frac{\mu\nu}{2}\right)\|\widetilde{\boldsymbol{w}}_{i-1}-\widetilde{\boldsymbol{w}}'_{i-1}\|^2\;+\;\frac{2}{\mu\nu}\|\mu\boldsymbol{c}_{i-1}\|^2
\end{aligned}
\tag{4.76}
$$

Using (4.46), we conclude from (4.73) and (4.76) that

$$
\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i-\widetilde{\boldsymbol{w}}'_i\|^2\le\left(1-\frac{\mu\nu}{2}\right)\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}-\widetilde{\boldsymbol{w}}'_{i-1}\|^2+\frac{\mu(\kappa'_1)^2}{2\nu}\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \tag{4.77}
$$

Now using (3.67) we conclude that (4.70) holds. With regards to (4.71), we note that

$$
\begin{aligned}
\mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_i\|^2&=\mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_i-\widetilde{\boldsymbol{w}}_i+\widetilde{\boldsymbol{w}}_i\|^2\\
&=\mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_i-\widetilde{\boldsymbol{w}}_i\|^2+\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2\;+\;2\left|\mathbb{E}\,(\widetilde{\boldsymbol{w}}'_i-\widetilde{\boldsymbol{w}}_i)^{\mathsf{T}}\widetilde{\boldsymbol{w}}_i\right|\\
&\le\mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_i-\widetilde{\boldsymbol{w}}_i\|^2+\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2\;+\;2\sqrt{\mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_i-\widetilde{\boldsymbol{w}}_i\|^2\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2}
\end{aligned}
\tag{4.78}
$$

where in the last step we used the property that $|\mathbb{E}\,\boldsymbol{a}^{\mathsf{T}}\boldsymbol{b}|^2\le\mathbb{E}\,\|\boldsymbol{a}\|^2\,\mathbb{E}\,\|\boldsymbol{b}\|^2$ for any two real random vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. Therefore, from (3.39) and (4.70) we get

$$
\limsup_{i\to\infty}\left(\mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_i\|^2-\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2\right)\;\le\;O(\mu^2)\;+\;\sqrt{O(\mu^3)}\;=\;O(\mu^{3/2}) \tag{4.79}
$$

since $\mu^2<\mu^{3/2}$ for small $\mu\ll 1$, which establishes (4.71). Similarly, we can write for any two real random vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ and constant symmetric positive-definite matrix H:

$$
\begin{aligned}
|\mathbb{E}\,\boldsymbol{a}^{\mathsf{T}}H\boldsymbol{b}|^2&\le\mathbb{E}\,\|\boldsymbol{a}\|^2\,\mathbb{E}\,\|H\boldsymbol{b}\|^2\\
&=\mathbb{E}\,\|\boldsymbol{a}\|^2\,\mathbb{E}\,\|\boldsymbol{b}\|^2_{H^2}\\
&\overset{(a)}{\le}\rho^2(H)\,\mathbb{E}\,\|\boldsymbol{a}\|^2\,\mathbb{E}\,\|\boldsymbol{b}\|^2
\end{aligned}
\tag{4.80}
$$

where the notation $\|x\|^2_A$ denotes the weighted quantity $x^{\mathsf{T}}Ax$, and in step (a) we used the Rayleigh-Ritz characterization for the eigenvalues of any symmetric matrix A [104, 113, 263]:

$$
\lambda_{\min}(A)\|x\|^2\;\le\;x^{\mathsf{T}}Ax\;\le\;\lambda_{\max}(A)\|x\|^2 \tag{4.81}
$$

In particular, by setting $\boldsymbol{b} = \boldsymbol{a}$, it also follows from (4.80) that $\mathbb{E}\,\|\boldsymbol{a}\|_H^2 \leq \rho(H)\mathbb{E}\,\|\boldsymbol{a}\|^2$. Therefore, repeating the argument that led to (4.78) using weighted norms we obtain

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i'\|_H^2 \leq \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_H^2 + \rho(H)\left[\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i' - \widetilde{\boldsymbol{w}}_i\|^2 + 2\sqrt{\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i' - \widetilde{\boldsymbol{w}}_i\|^2\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2}\right] \tag{4.82}$$

and we arrive at (4.72).

□

4.5 Performance Metrics

Two useful metrics for assessing the performance of stochastic gradient algorithms are the mean-square-deviation (MSD) and the excess-risk (ER). We define these two measures below before explaining how the long-term model (4.55) can be used to evaluate their values.

Mean-Square-Deviation (MSD)

To motivate the definition of the MSD, we first remark that we will be establishing further ahead in (4.97) and (4.128) the following two expressions for the limit superior and limit inferior of the error variance:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; \mu\cdot\overline{\text{MSD}} \;+\; o(\mu) \tag{4.83}$$

$$\liminf_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; \mu\cdot\overline{\text{MSD}} \;-\; o(\mu) \tag{4.84}$$

for some common positive constant $\overline{\text{MSD}}$ whose exact value is not relevant for the current discussion. We explained the meaning of the limit superior operation earlier prior to the statement of Lemma 3.1. We can similarly view the *limit inferior* of a sequence as essentially corresponding to the largest lower bound for the limiting behavior of the sequence; this concept is again useful when the sequence is not necessarily convergent but tends towards a small bounded region [89, 144, 202]. A schematic illustration of the limit superior and limit inferior values for the error variance, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, is shown in Figure 4.1. If the sequence happens to be convergent, then both its limit superior and limit inferior values will coincide and they will be equal to the regular limiting value of the sequence.

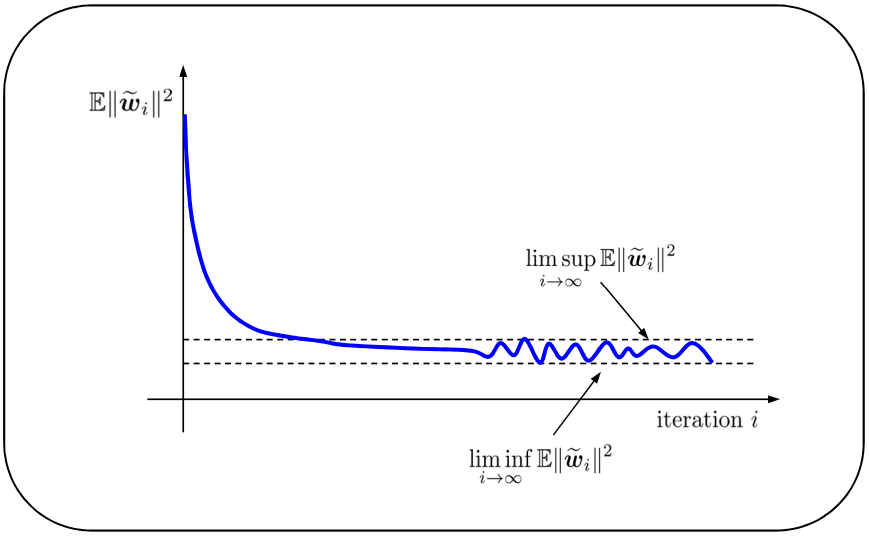

Figure 4.1: Schematic illustration of the limit superior and limit inferior bounds on the error variance sequence, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$.

Now, comparing the first relation (4.83) with (4.2), it is observed that (4.83) characterizes the size of the coefficient of the first-order term in μ as being equal to $\overline{\text{MSD}}$. Moreover, if we divide both sides of (4.83) and (4.84) by μ and compute the limit as $\mu \to 0$, which corresponds to assuming operation in the slow adaptation regime, then we find that

$$\lim_{\mu\to 0}\left(\limsup_{i\to\infty}\frac{1}{\mu}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2\right) = \lim_{\mu\to 0}\left(\liminf_{i\to\infty}\frac{1}{\mu}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2\right) = \overline{\text{MSD}} \quad (4.85)$$

That is, the limiting values of the *scaled* limit superior and limit inferior expressions coincide with each other and they are both equal to $\overline{\text{MSD}}$. This fact indicates that as $\mu \to 0$, the quantity $\frac{1}{\mu}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ approaches a limiting value after sufficient iterations and, once multiplied by μ, this limiting value can be used to assess the size of the error variance, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, in steady-state. For this reason, we shall define the MSD measure as follows:

$$\text{MSD} \;\triangleq\; \mu\cdot\left(\lim_{\mu\to 0}\limsup_{i\to\infty}\frac{1}{\mu}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2\right) \quad (4.86)$$

In view of equality (4.85), we could have also defined the MSD by using the lim inf operation in (4.86) instead of the lim sup operation. For uniformity throughout this work, we shall adopt the lim sup notation.

Sometimes, with some abuse of notation, we write the definition for the MSD more simply, for sufficiently small step-sizes, as follows:

$$\text{MSD} \triangleq \lim_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \tag{4.87}$$

with the understanding that this limit is computed as in (4.86) since, strictly speaking, the limit on the right-hand side of (4.87) may not exist. Yet, it is useful to note that derivations that assume the validity of (4.87) still lead to the same expression for the MSD to first-order in μ as derivations that rely on the more formal expression (4.86) — this fact can be verified by examining and repeating the proof of Theorem 4.7 further ahead.

Excess Risk (ER)

The second useful metric for evaluating the performance of stochastic gradient algorithms relates to the mean excess-cost; which is also called the *excess-risk* (ER) in the machine learning literature [37, 233] and the *excess-mean-square-error* (EMSE) in the adaptive filtering literature [107, 206, 262]. We denote it by the letters ER and, similarly to (4.86), we can motivate the following expression for it:

$$\text{ER} \triangleq \mu \cdot \left(\lim_{\mu\to 0} \limsup_{i\to\infty} \frac{1}{\mu} \mathbb{E}\,\{J(\boldsymbol{w}_{i-1}) - J(w^o)\} \right) \tag{4.88}$$

In other words, the ER metric measures the average fluctuation of the cost function around its minimum value in steady-state. Again, we could have used the lim inf operation in (4.88) instead of the lim sup operation. We again adopt the lim sup convention.

Using the smoothness condition (4.20), and result (E.10) from the appendix, we recognize that the mean fluctuation that appears inside (4.88) satisfies:

$$\limsup_{i\to\infty} \mathbb{E}\,\{J(\boldsymbol{w}_{i-1}) - J(w^o)\} = \limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\frac{1}{2}H} + O(\mu^{3/2}) \tag{4.89}$$

in terms of a weighted mean-square-error norm. The appearance of the $O(\mu^{3/2})$ factor in the above expression can be motivated as follows. We

note from expression (E.10) in the appendix that the right-most term in (4.89) should be the asymptotic size of $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^3$. We then rely on result (3.67) to note that:

$$\begin{aligned}\limsup_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^3 &\overset{(F.30)}{\leq} \limsup_{i\to\infty}\left(\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4\right)^{3/4}\\ &\overset{(3.67)}{=} \left(O(\mu^2)\right)^{3/4}\\ &= O(\mu^{3/2})\end{aligned} \tag{4.90}$$

where in the first line we called upon Jensen's inequality (F.30) and the fact that the function $f(x) = x^{3/4}$ is concave over the range $x \geq 0$. It follows from (4.88) and (4.89) that we can also evaluate the ER metric by means of the following alternative expression:

$$\text{ER} \;=\; \mu\cdot\left(\lim_{\mu\to 0}\limsup_{i\to\infty}\frac{1}{\mu}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\frac{1}{2}H}\right) \tag{4.91}$$

Again, with some abuse in notation, we sometimes write more simply either of the following expressions for sufficiently small step-sizes in place of (4.88) and (4.91):

$$\text{ER} \;=\; \lim_{i\to\infty}\mathbb{E}\,\{J(\boldsymbol{w}_{i-1}) - J(w^o)\} \tag{4.92}$$

$$\text{ER} \;=\; \lim_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\frac{1}{2}H} \tag{4.93}$$

with the understanding that the limits in the above two expressions are computed as in (4.88) or (4.91) since, strictly speaking, these limits may not exist. Still, it is useful to note that derivations that assume the validity of (4.92)–(4.93) lead to the same expression for the ER to first-order in μ as derivations that rely on the more formal expressions (4.88) or (4.91) — this fact can be verified by examining and repeating the proof of Theorem 4.7. We collect the expressions for the MSD and ER measures in the following statement for ease of reference.

Definition 4.2 (Performance measures)**.** The mean-square-deviation (MSD) and excess-risk (ER) performance metrics are defined as follows:

$$\text{MSD} \triangleq \mu \cdot \left(\lim_{\mu \to 0} \limsup_{i \to \infty} \frac{1}{\mu} \mathbb{E} \, \|\widetilde{\boldsymbol{w}}_i\|^2 \right) \tag{4.94}$$

$$\text{ER} \triangleq \mu \cdot \left(\lim_{\mu \to 0} \limsup_{i \to \infty} \frac{1}{\mu} \mathbb{E} \, \{J(\boldsymbol{w}_{i-1}) - J(w^o)\} \right) \tag{4.95}$$

for sufficiently small step-sizes, where the MSD measures the size of the error variance, $\mathbb{E} \, \|\widetilde{\boldsymbol{w}}_i\|^2$, in steady-state, while the ER measures the size of the mean fluctuation, $\mathbb{E} \, \{J(\boldsymbol{w}_{i-1}) - J(w^o)\}$, also in steady state. Under result (3.67), and using the Hessian matrix H from (4.40), the ER expression can also be evaluated as:

$$\text{ER} = \mu \cdot \left(\lim_{\mu \to 0} \limsup_{i \to \infty} \frac{1}{\mu} \mathbb{E} \, \|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\frac{1}{2}H} \right) \tag{4.96}$$

It is noteworthy to observe from (4.94) and (4.96) that both expressions for the MSD and ER involve squared norms of the error vector, $\widetilde{\boldsymbol{w}}_i$, in steady-state. For this reason, in the argument that follows we will focus on evaluating the limit superior of a *weighted* mean-square-error norm of the form $\mathbb{E} \, \|\widetilde{\boldsymbol{w}}_i\|^2_\Sigma$, for some positive-definite weighting matrix Σ that we are free to choose. Then, by setting $\Sigma = I_M$ or $\Sigma = \frac{1}{2}H$, we will be able to arrive at the MSD and ER values.

Theorem 4.7 (Mean-square-error performance: Real case)**.** Assume the conditions under Assumptions 4.1, 4.2, and 4.4 on the cost function and the gradient noise process hold. Assume further that the step-size is sufficiently small to ensure mean-square stability, as already ascertained by Lemmas 3.1 and 4.4. Then, it holds that

$$\limsup_{i \to \infty} \mathbb{E} \, \|\widetilde{\boldsymbol{w}}_i\|^2 = \frac{\mu}{2} \operatorname{Tr}\left(H^{-1} R_s\right) + O\left(\mu^{1+\gamma_m}\right) \tag{4.97}$$

$$\limsup_{i \to \infty} \mathbb{E} \, \{J(\boldsymbol{w}_{i-1}) - J(w^o)\} = \frac{\mu}{4} \operatorname{Tr}\left(R_s\right) + O\left(\mu^{1+\gamma_m}\right) \tag{4.98}$$

where

$$\gamma_m \triangleq \frac{1}{2} \min\{1, \gamma\} > 0 \tag{4.99}$$

with $0 < \gamma \leq 4$ from (4.19), while R_s and H are defined by (4.12) and (4.40). Consequently, the MSD and ER metrics defined by (4.94) and (4.96) for the stochastic-gradient algorithm (4.5) are given by the following expressions:

$$\text{MSD} = \frac{\mu}{2}\,\text{Tr}\left(H^{-1}R_s\right) \tag{4.100}$$

$$\text{ER} = \frac{\mu}{4}\,\text{Tr}\left(R_s\right) \tag{4.101}$$

Moreover, for $i \gg 1$, the rate at which the error variance, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, approaches its steady-state region (4.97) is well-approximated to first-order in μ by

$$\alpha = 1 - 2\mu\lambda_{\min}(H) \tag{4.102}$$

Proof. We introduce the eigen-decomposition $H = U\Lambda U^{\mathsf{T}}$, where U is orthogonal and Λ is diagonal with positive entries, and rewrite (4.55) in terms of transformed quantities:

$$\overline{\boldsymbol{w}}_i = (I - \mu\Lambda)\,\overline{\boldsymbol{w}}_{i-1} + \mu\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1}) \tag{4.103}$$

where $\overline{\boldsymbol{w}}_i = U^{\mathsf{T}}\widetilde{\boldsymbol{w}}_i'$ and $\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1}) = U^{\mathsf{T}}\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$. Since the variables $\{\widetilde{\boldsymbol{w}}_i', \overline{\boldsymbol{w}}_i\}$ are related to each other via an orthogonal transformation, it is clear that their Euclidean norms are identical and, therefore, $\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|^2 = \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i'\|^2$. It follows that we can rely on the mean-square-error of $\overline{\boldsymbol{w}}_i$ to evaluate the mean-square-deviation (MSD) of the long-term model (4.55). We proceed to derive an expression for the MSD by employing energy conservation arguments [6, 205, 206, 269].

Let Σ denote an arbitrary $M \times M$ diagonal matrix with positive entries that we are free to choose. Then, equating the weighted squared norms of both sides of (4.103) and taking expectations conditioned on the past history $\boldsymbol{\mathcal{F}}_{i-1}$ gives :

$$\mathbb{E}\left[\,\|\overline{\boldsymbol{w}}_i\|_{\Sigma}^2|\boldsymbol{\mathcal{F}}_{i-1}\,\right] = \|\overline{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2 + \mu^2\mathbb{E}\left[\,\|\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1})\|_{\Sigma}^2|\boldsymbol{\mathcal{F}}_{i-1}\,\right] \tag{4.104}$$

where the cross terms are annihilated on the right-hand side because $\mathbb{E}\,[\,\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1})|\boldsymbol{\mathcal{F}}_{i-1}\,] = 0$. Moreover, the weighting matrix Σ' is given by

$$\begin{aligned}\Sigma' &\triangleq (I - \mu\Lambda)\Sigma(I - \mu\Lambda)\\ &= \Sigma - 2\mu\Lambda\Sigma + \mu^2\Lambda\Sigma\Lambda\end{aligned} \tag{4.105}$$

Taking expectations of both sides of (4.104) gives:

$$\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{\Sigma}^2 = \mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2 + \mu^2\,\mathbb{E}\,\|\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1})\|_{\Sigma}^2 \tag{4.106}$$

We now evaluate the two terms that appear on the right-hand side of this expression for $i \gg 1$. With regards to the first term, we use expression (4.105) for Σ' to note that:

$$\mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} \overset{(4.105)}{=} \mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2_{\Sigma-2\mu\Lambda\Sigma} + \mu^2\,\mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2_{\Lambda\Sigma\Lambda} \tag{4.107}$$

Now since Σ and Λ are diagonal matrices with positive entries, we observe that the rightmost term satisfies:

$$\begin{aligned}\mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2_{\Lambda\Sigma\Lambda} &\leq \rho(\Lambda^2)\cdot\rho(\Sigma)\cdot\mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2 \\ &\leq \rho(\Lambda^2)\cdot\mathrm{Tr}(\Sigma)\cdot\mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2 \end{aligned}\tag{4.108}$$

where $\rho(A)$ denotes the spectral radius of its matrix argument; obviously, for the matrices Σ and Λ, we have that $\rho(\Lambda)$ is equal to the largest entry in Λ while $\rho(\Sigma)$ is smaller than the trace of Σ. Combining the above result with the fact from (3.39) that the limit superior of $\mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2$ is in the order of $O(\mu)$, we conclude from (4.107) that for $i \gg 1$:

$$\mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} = \mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2_{\Sigma-2\mu\Lambda\Sigma} + \mathrm{Tr}(\Sigma)\cdot O(\mu^3) \tag{4.109}$$

where we are keeping the factor $\mathrm{Tr}(\Sigma)$ explicit in the rightmost term for later use in (4.129).

We next evaluate the second term on the right-hand side of (4.106). To do so, we shall call upon the results of Lemma 4.1. We start by noting that

$$\begin{aligned}\mathbb{E}\,\|\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1})\|^2_{\Sigma} &= \mathrm{Tr}\left[\Sigma\,\mathbb{E}\left(\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1})\,(\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}}\right)\right] \\ &= \mathrm{Tr}\left[U\Sigma U^{\mathsf{T}}\,\mathbb{E}\left(\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}}\right)\right] \end{aligned}\tag{4.110}$$

where the covariance matrix $\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}}$ was already evaluated earlier in (4.33). Using that result, and the sub-multiplicative property of norms, namely, $\|AB\| \leq \|A\|\,\|B\|$, we conclude that:

$$\limsup_{i\to\infty}\left\|U\Sigma U^{\mathsf{T}}\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\,(\boldsymbol{s}_i(\boldsymbol{w}_{i-1}))^{\mathsf{T}} - U\Sigma U^{\mathsf{T}}R_s\right\| = O(\mu^{\gamma'/2}) \tag{4.111}$$

where γ' was defined in (4.32) as $\gamma' = \min\{\gamma, 2\}$. Consequently, as stated earlier prior to (4.34), since $|\mathrm{Tr}(X)| \leq c\,\|X\|$ for any square matrix X, we have that:

$$\limsup_{i\to\infty}\left|\,\mathbb{E}\,\|\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1})\|^2_{\Sigma} - \mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s)\,\right| = O(\mu^{\gamma'/2}) \triangleq b_1 \tag{4.112}$$

in terms of the absolute value of the difference. We are denoting the value of the limit superior by the nonnegative number b_1; we know from (4.112) that $b_1 = O(\mu^{\gamma'/2})$. The same argument that led to (4.26) then leads to

$$\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) - b_o \leq \mathbb{E}\,\|\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1})\|^2_{\Sigma} \leq \mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) + b_o \tag{4.113}$$

for $i \gg 1$ and for some nonnegative constant $b_o = O(\mu^{\gamma'/2})$. It follows from (4.113) that we can also write, for $i \gg 1$:

$$\mathbb{E}\,\|\overline{\boldsymbol{s}}_i(\boldsymbol{w}_{i-1})\|_{\Sigma}^2 \;=\; \mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) \;+\; O(\mu^{\gamma'/2}) \tag{4.114}$$

Substituting results (4.109) and (4.113) into the variance relation (4.106) we obtain for $i \gg 1$ that:

$$\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{\Sigma}^2 \;\leq\; \mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|_{\Sigma-2\mu\Lambda\Sigma}^2 + \mu^2\left(\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) + b_o\right) + O(\mu^3) \tag{4.115}$$

$$\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{\Sigma}^2 \;\geq\; \mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|_{\Sigma-2\mu\Lambda\Sigma}^2 + \mu^2\left(\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) - b_o\right) + O(\mu^3) \tag{4.116}$$

Using the sub-additivity and super-additivity properties of the limit superior and limit inferior operations, namely, for bounded sequences $a(i)$ and $b(i)$ [89, 144, 202]:

$$\limsup_{i\to\infty}\,(a(i)+b(i)) \;\leq\; \limsup_{i\to\infty}\,a(i) \;+\; \limsup_{i\to\infty}\,b(i) \tag{4.117}$$

$$\liminf_{i\to\infty}\,(a(i)+b(i)) \;\geq\; \liminf_{i\to\infty}\,a(i) \;+\; \liminf_{i\to\infty}\,b(i) \tag{4.118}$$

we conclude from (4.115) and (4.116) that

$$\limsup_{i\to\infty}\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{\Sigma}^2 \;\leq\; \limsup_{i\to\infty}\mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|_{\Sigma-2\mu\Lambda\Sigma}^2 + \mu^2\left(\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) + b_o\right) + O(\mu^3) \tag{4.119}$$

and

$$\liminf_{i\to\infty}\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{\Sigma}^2 \;\geq\; \liminf_{i\to\infty}\mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|_{\Sigma-2\mu\Lambda\Sigma}^2 + \mu^2\left(\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) - b_o\right) + O(\mu^3) \tag{4.120}$$

Grouping terms we get:

$$\limsup_{i\to\infty}\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{2\mu\Lambda\Sigma}^2 \;\leq\; \mu^2\left(\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) + b_o\right) + O(\mu^3) \tag{4.121}$$

$$\liminf_{i\to\infty}\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{2\mu\Lambda\Sigma}^2 \;\geq\; \mu^2\left(\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) - b_o\right) + O(\mu^3) \tag{4.122}$$

and, consequently, by eliminating a common factor μ from all terms and using the fact that the limit inferior of a sequence is upper bounded by its limit superior, we obtain the following inequality relation:

$$\begin{aligned}\mu\left(\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) - b_o\right) + O(\mu^2) \;&\leq\; \liminf_{i\to\infty}\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{2\Lambda\Sigma}^2 \\ &\leq\; \limsup_{i\to\infty}\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{2\Lambda\Sigma}^2 \;\leq\; \mu\left(\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) + b_o\right) + O(\mu^2)\end{aligned} \tag{4.123}$$

Recalling that $b_o = O(\mu^{\gamma'/2})$ and $0 < \frac{\gamma'}{2} \leq 1$ so that μb_o dominates $O(\mu^2)$ for small μ, we conclude that the limit superior and limit inferior of the error variance satisfy:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|^2_{2\Lambda\Sigma} \;=\; \mu\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) \;+\; O\left(\mu^{\min\left\{2,\,1+\frac{\gamma}{2}\right\}}\right) \tag{4.124}$$

$$\liminf_{i\to\infty} \mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|^2_{2\Lambda\Sigma} \;=\; \mu\mathrm{Tr}(U\Sigma U^{\mathsf{T}}R_s) \;-\; O\left(\mu^{\min\left\{2,\,1+\frac{\gamma}{2}\right\}}\right) \tag{4.125}$$

Continuing with (4.124), and since we are free to choose Σ, we let $\Sigma = \frac{1}{2}\Lambda^{-1}$ so that the variance term on the left-hand side of (4.124) becomes $\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|^2$. Recalling that $\|\overline{\boldsymbol{w}}_i\|^2 = \|\widetilde{\boldsymbol{w}}'_i\|^2$ and noting that $U\Sigma U^{\mathsf{T}} = \frac{1}{2}H^{-1}$, we arrive at

$$\limsup_{i\to\infty}\; \mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_i\|^2 = \frac{\mu}{2}\,\mathrm{Tr}\left(H^{-1}R_s\right) \;+\; O\left(\mu^{\min\left\{2,\,1+\frac{\gamma}{2}\right\}}\right) \tag{4.126}$$

However, we know from result (4.71) that the error variance of the stochastic-gradient algorithm (4.5) is within $O(\mu^{3/2})$ from the error variance of the long-term model, which is given by the above expression. We therefore need to adjust the exponent of μ inside the big-O term to arrive at the desired expression (4.97) where the factor of 2 is replaced by 3/2 since

$$\min\left\{\frac{3}{2},\,2,\,1+\frac{\gamma}{2}\right\} \;=\; \min\left\{\frac{3}{2},\,1+\frac{\gamma}{2}\right\} \tag{4.127}$$

Likewise, if we select $\Sigma = \frac{1}{4}I_M$, then a similar argument leads to (4.98). Returning to (4.125), the argument that led to (4.126) would similarly imply that

$$\liminf_{i\to\infty}\; \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 = \frac{\mu}{2}\,\mathrm{Tr}\left(H^{-1}R_s\right) \;-\; o(\mu) \tag{4.128}$$

Although this result is unnecessary for the argument in this proof, we nevertheless established it because it was used earlier in (4.84) while motivating the definition of the MSD metric.

With regards to the rate at which $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ approaches its steady-state region (4.97) (and likewise for $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2_{\frac{1}{2}H}$), we refer back to (4.106) and substitute (4.109) and (4.114) to rewrite the former relation as follows for $i \gg 1$:

$$\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|^2_{(I_M-2\mu\Lambda)\Sigma} + \mu^2\mathrm{Tr}(\Sigma U^{\mathsf{T}}R_sU) + \mathrm{Tr}(\Sigma)\cdot o(\mu^2) \tag{4.129}$$

where we replaced the approximation error by $o(\mu^2)$ for brevity; it is sufficient to know for the current argument that the power of μ is strictly larger than two. For compactness of notation, we introduce the matrices

$$D \;\triangleq\; I_M - 2\mu\Lambda, \qquad Y \;\triangleq\; U^{\mathsf{T}}R_sU \tag{4.130}$$

It is clear that the matrix D is stable for sufficiently small step-sizes and, moreover,

$$\rho(D) \overset{(4.130)}{=} 1 - 2\mu\lambda_{\min}(H) \tag{4.131}$$

where we used the fact that the eigenvalues of Λ coincide with the eigenvalues of H and they are all positive. Therefore, $D^i \to 0$ as $i \to \infty$ and, moreover,

$$\sum_{n=0}^{\infty} D^n = (I_M - D)^{-1} = \frac{1}{2\mu}\Lambda^{-1} \tag{4.132}$$

so that

$$o(\mu^2) \cdot \mathrm{Tr}\left(\sum_{n=0}^{\infty} D^n\right) \overset{(4.132)}{=} o(\mu) \tag{4.133}$$

These two conclusions are used in the sequel. Indeed, from (4.129) we have that for any $i \gg 1$:

$$\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|_{\Sigma}^2 \;=\; \mathbb{E}\,\|\overline{\boldsymbol{w}}_{i-1}\|_{D\Sigma}^2 \;+\; \mu^2\mathrm{Tr}(\Sigma Y) \;+\; \mathrm{Tr}(\Sigma)\cdot o(\mu^2) \tag{4.134}$$

By setting Σ successively equal to the choices $\{I_M,\ D,\ D^2,\ D^3\ \ldots\}$, and by iterating the above recursion, we deduce that

$$\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|^2 \;=\; \mathbb{E}\,\|\overline{\boldsymbol{w}}_{-1}\|_{D^{i+1}}^2 \;+\; \mu^2\sum_{n=0}^{i}\mathrm{Tr}(D^nY) \;+\; o(\mu^2)\cdot\sum_{n=0}^{i}\mathrm{Tr}(D^n) \tag{4.135}$$

The first-term on the right-hand side corresponds to a transient component that dies out with time. The rate of its convergence towards zero determines the rate of convergence of $\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|^2$ towards its steady-state region. This rate can be characterized as follows. We express the weighted variance of $\overline{\boldsymbol{w}}_{-1}$ as the following trace relation in terms of its un-weighted covariance matrix:

$$\mathbb{E}\,\|\overline{\boldsymbol{w}}_{-1}\|_{D^{i+1}}^2 \;=\; \mathbb{E}\,\left(\overline{\boldsymbol{w}}_{-1}^{*}D^{i+1}\overline{\boldsymbol{w}}_{-1}\right) \;=\; \mathrm{Tr}\left(D^{i+1}\mathbb{E}\,\overline{\boldsymbol{w}}_{-1}\overline{\boldsymbol{w}}_{-1}^{*}\right) \tag{4.136}$$

Then, it is clear that the convergence rate of the transient component is dictated by $\rho(D)$ since this value characterizes the slowest rate at which the transient term dies out. We conclude that the convergence rate of $\mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|^2$ towards the steady-state regime is also dictated by $\rho(D)$, which we can approximate to first-order in μ by expression (4.102).

Additionally, if desired, computing the limit superior of both sides of (4.135), and using (4.133), we can re-derive the MSD value for the algorithm in an alternative route as follows. Note that

$$\limsup_{i\to\infty}\ \mathbb{E}\,\|\overline{\boldsymbol{w}}_i\|^2 \;=\; \mu^2\left(\sum_{n=0}^{\infty}\mathrm{Tr}\left(D^nY\right)\right) \;+\; o(\mu) \tag{4.137}$$

where the first term on the right-hand side is actually $O(\mu)$ and dominates the second term since

$$\begin{aligned}\mu^2\left(\sum_{n=0}^{\infty}\mathrm{Tr}\left(D^nY\right)\right) &= \mu^2\mathrm{Tr}\left[\left(I_M+D+D^2+D^3+\ldots\right)Y\right]\\ &= \mu^2\mathrm{Tr}\left((I_M-D)^{-1}Y\right)\\ &\overset{(4.132)}{=} \frac{\mu}{2}\mathrm{Tr}\left(\Lambda^{-1}Y\right)\\ &= O(\mu)\end{aligned} \tag{4.138}$$

If we now use the substitutions $Y = U^{\mathsf{T}}R_sU$, $\Lambda^{-1} = U^{\mathsf{T}}H^{-1}U$, and $\overline{\boldsymbol{w}}_i = U^{\mathsf{T}}\widetilde{\boldsymbol{w}}_i'$, we conclude that

$$\limsup_{i\to\infty}\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i'\|^2 \;=\; \frac{\mu}{2}\mathrm{Tr}\left(H^{-1}R_s\right) \;+\; o(\mu) \tag{4.139}$$

which is in agreement with (4.126).

□

Results (4.100)–(4.101) are useful expressions that apply to general $\nu-$strongly convex functions $J(w)$ that satisfy Assumptions 4.1 and 4.3. The following example shows that the approximation error in expressions (4.97)–(4.98) can be replaced by $O(\mu^2)$ in the quadratic case.

Example 4.2 (Quadratic cost functions). When $J(w)$ happens to be quadratic in w, as is the case with the mean-square-error cost of Example 3.1, then the matrices $\boldsymbol{H}_{i-1}$ and H defined by (4.38) and (4.39), respectively, will coincide with each other since the Hessian matrix $\nabla_w^2 J(w)$ will be constant for all w. Thus, in this case $\boldsymbol{H}_{i-1} \equiv H = \nabla_w^2 J(w^o)$. As a result, the perturbation term $\mu\boldsymbol{c}_{i-1}$ in (4.41) will be identically zero and recursions (4.37) and (4.55) will therefore coincide. Both models will then have the same MSD expressions. Therefore, we can rely on expression (4.126) without the need for the adjustment by $O(\mu^{3/2})$. We know from (4.16) that $\gamma = 2$ for mean-square-error costs. Using this value for γ in (4.126), we arrive at

$$\limsup_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; \frac{\mu}{2}\,\mathrm{Tr}\left(H^{-1}R_s\right) \;+\; O\left(\mu^2\right) \tag{4.140}$$

with an approximation error in the order of $O(\mu^2)$ rather than the term $O(\mu^{3/2})$ that would result from (4.97)–(4.98). Likewise, we obtain

$$\limsup_{i\to\infty}\mathbb{E}\left\{J(\boldsymbol{w}_{i-1})-J(w^o)\right\} \;=\; \frac{\mu}{4}\,\mathrm{Tr}\left(R_s\right) \;+\; O\left(\mu^2\right) \tag{4.141}$$

In re-deriving this expression for the ER, we called upon expression (E.20) in the appendix where it is shown that for quadratic costs, expression (4.89) is replaced by the exact relation

$$\limsup_{i\to\infty} \mathbb{E}\,\{J(\boldsymbol{w}_{i-1}) - J(w^o)\} = \limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\frac{1}{2}H} \tag{4.142}$$

without the $O(\mu^{3/2})$ correction term that appeared in (4.89).

The resulting expressions for the MSD and ER performance metrics will continue to be:

$$\text{MSD} = \frac{\mu}{2}\,\text{Tr}\left(H^{-1}R_s\right) \tag{4.143}$$

$$\text{ER} = \frac{\mu}{4}\,\text{Tr}\left(R_s\right) \tag{4.144}$$

With regards to the convergence rate, we use $\gamma = 2$ (and, hence, $\gamma' = 2$) in (4.114) and recognize that the $o(\mu^2)$ term in (4.129) will be replaced by $O(\mu^3)$. Continuing with the derivation, we will then conclude that the approximation error $o(\mu)$ in (4.137) is replaced by $O(\mu^2)$ and the convergence rate expression (4.102) will still hold in the quadratic case:

$$\alpha = 1 - 2\mu\lambda_{\min}(H) \tag{4.145}$$

■

The examples that follow show how expressions (4.100)–(4.101) can be used to recover classical results for mean-square-error adaptation and learning.

Example 4.3 (Performance of LMS adaptation)**.** We reconsider the LMS recursion (3.13). We know from Example 3.3 and (4.13) that this situation corresponds to $H = 2R_u$ and $R_s = 4\sigma_v^2 R_u$. Substituting into (4.100)–(4.101) leads to the following well-known expressions for the performance of the LMS filter for sufficiently small step-sizes — see [96, 97, 100, 107, 114, 130, 206, 261, 262]:

$$\text{MSD} = \mu M\,\sigma_v^2 = O(\mu) \tag{4.146}$$

$$\text{EMSE} = \mu\,\sigma_v^2\,\text{Tr}(R_u) = O(\mu) \tag{4.147}$$

where we are replacing ER by the notation EMSE, which is more common in the adaptive filtering literature.

Figure 4.2 illustrates this situation numerically. The figure plots the evolution of the ensemble-average learning curve, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, over i; the curve is generated by averaging the trajectories $\{\|\widetilde{\boldsymbol{w}}_i\|^2\}$ over 2000 repeated experiments. The label on the vertical axis in the figure refers to the learning

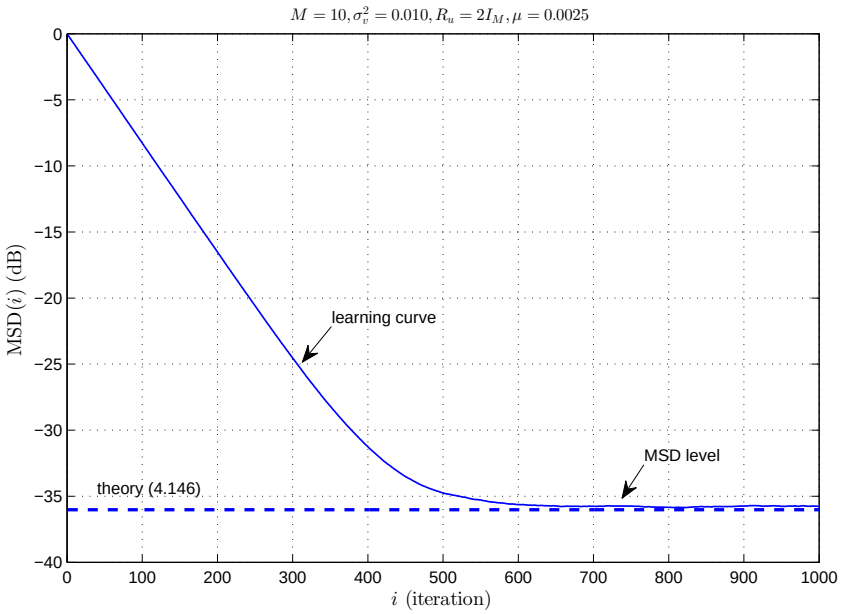

Figure 4.2: Learning curve, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, for the LMS rule (3.13) obtained by averaging over 2000 repeated experiments using $M = 10$, $\sigma_v^2 = 0.010$, $R_u = 2I_M$, and $\mu = 0.0025$. The horizontal dashed line indicates the steady-state MSD level predicted by the theoretical expression (4.146).

curve $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ by writing MSD(i), with an iteration index i. Each experiment involves running the LMS recursion (3.13) on data $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ generated according to the model $\boldsymbol{d}(i) = \boldsymbol{u}_i w^o + \boldsymbol{v}(i)$ with $M = 10$, $\sigma_v^2 = 0.010$, $R_u = 2I_M$, and using $\mu = 0.0025$. The unknown vector w^o is generated randomly and its norm is normalized to one. It is seen in the figure that the learning curve tends to the MSD value predicted by the theoretical expression (4.146). ■

Example 4.4 (Performance of logistic learners)**.** We reconsider the stochastic-gradient algorithm (3.16) from Example 3.2 for logistic regression. The absolute component of the gradient noise in that example is given by

$$\boldsymbol{s}_i(w^o) \;=\; \rho w^o \;-\; \boldsymbol{\gamma}(i)\boldsymbol{h}_i\left(\frac{1}{1+e^{\boldsymbol{\gamma}(i)\boldsymbol{h}_i^{\mathsf{T}} w^o}}\right) \tag{4.148}$$

with covariance matrix

$$R_s \triangleq \mathbb{E}\left\{\boldsymbol{h}_i\boldsymbol{h}_i^{\mathsf{T}}\cdot\left(\frac{1}{1+e^{\boldsymbol{\gamma}(i)\boldsymbol{h}_i^{\mathsf{T}}w^o}}\right)^2\right\} - \rho^2 w^o(w^o)^{\mathsf{T}} \tag{4.149}$$

Note in particular that $R_s \leq R_h$. Calling upon expression (4.101), we conclude that the excess-risk measure is given by

$$\text{ER} \;=\; \frac{\mu}{4}\,\text{Tr}\,(R_s) \;\leq\; \frac{\mu}{4}\,\text{Tr}\,(R_h) \;=\; O(\mu) \tag{4.150}$$

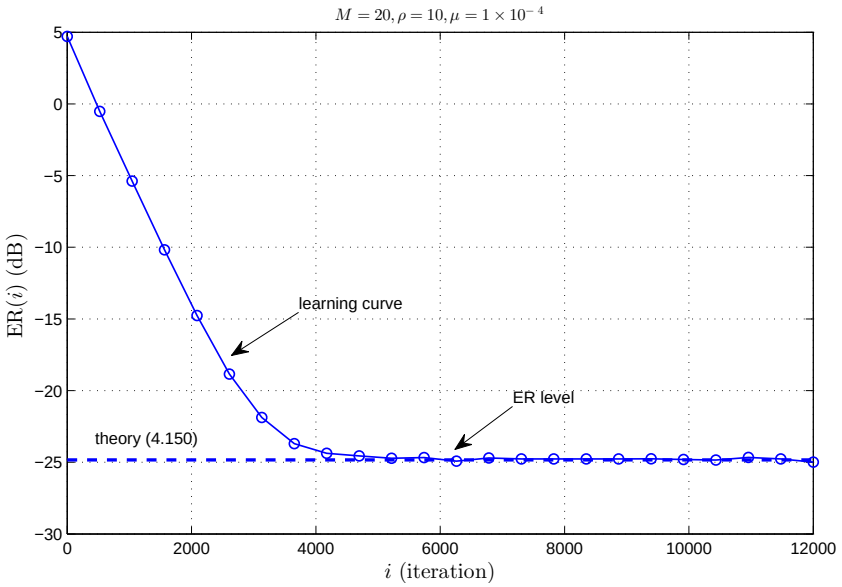

Figure 4.3: Learning curve, $\mathbb{E}\,\{J(\boldsymbol{w}_{i-1} - J(w^o)\}$, for the logistic rule (3.16) obtained by averaging over 100 repeated experiments using $M = 50$, $\rho = 10$, and $\mu = 4 \times 10^{-5}$. The horizontal dashed line indicates the steady-state ER level predicted by the theoretical expression (4.150).

Figure 4.3 illustrates this situation numerically. The figure plots the evolution of the ensemble-average excess-risk curve, $\mathbb{E}\,\{J(\boldsymbol{w}_{i-1}) - J(w^o)\}$, over i; the curve is generated by averaging the curves $\{J(\boldsymbol{w}_{i-1}) - J(w^o)\}$ over 100 repeated experiments. The label on the vertical axis in the figure refers to the learning curve $\mathbb{E}\,\{J(\boldsymbol{w}_{i-1}) - J(w^o)\}$ by writing ER(i), with an iteration index i. Each experiment involves running the logistic recursion (3.16) on data $\{\boldsymbol{\gamma}(i), \boldsymbol{h}_i\}$ with $M = 50$, $\rho = 10$, and $\mu = 1 \times 10^{-4}$. The data used for the

simulation originate from the alpha data set [223]; we use the first 50 features for illustration purposes so that $M = 50$. To generate the trajectories for the experiments in this example, the optimal w^o and the gradient noise covariance matrix, R_s, are first estimated off-line by applying a batch algorithm to all data points. For the data used in this example we have $\mathrm{Tr}(R_s) \approx 131.48$ and $\mathrm{Tr}(R_h) \approx 528.10$. It is seen in the figure that the learning curve tends to the ER value predicted by the theoretical expression (4.150).

■

Example 4.5 (Performance of online learners)**.** More generally, consider a stand-alone learner receiving a streaming sequence of independent data vectors $\{\boldsymbol{x}_i, i \geq 0\}$ that arise from some fixed probability distribution $\mathcal{X}$. The goal is to learn the vector w^o that optimizes some $\nu-$strongly convex risk function $J(w)$ defined in terms of a loss function [236, 252]:

$$w^o \triangleq \arg\min_w J(w) = \arg\min_w \mathbb{E}\, Q(w; \boldsymbol{x}_i) \tag{4.151}$$

The learner seeks w^o by running the stochastic-gradient algorithm:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \nabla_{w^\mathsf{T}} Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_i), \quad i \geq 0 \tag{4.152}$$

so that the gradient noise vector is given by

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) = \nabla_{w^\mathsf{T}} Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_i) - \nabla_{w^\mathsf{T}} J(\boldsymbol{w}_{i-1}) \tag{4.153}$$

Since $\nabla_w J(w^o) = 0$, and since the distribution of $\boldsymbol{x}_i$ is assumed stationary, it follows that the covariance matrix of $\boldsymbol{s}_i(w^o)$ is constant and given by

$$R_s = \mathbb{E}\, \nabla_{w^\mathsf{T}} Q(w^o; \boldsymbol{x}_i) \nabla_w Q(w^o; \boldsymbol{x}_i) \tag{4.154}$$

The excess-risk measure that will result from this stochastic implementation is then given by (4.101) so that

$$\mathrm{ER} = \frac{\mu}{4} \mathrm{Tr}(R_s) \tag{4.155}$$

■

4.6 Performance in the Complex Domain

We now extend the performance results of the previous sections to the complex domain in which case the argument $w \in \mathbb{C}^M$ is complex-valued. We explained in Sec. 3.6 that the strongly convex function, $J(w) \in \mathbb{R}$, is now required to satisfy condition (3.114), namely,

$$0 < \frac{\nu}{h} I_{hM} \leq \nabla_w^2 J(w) \leq \frac{\delta}{h} I_{hM} \tag{4.156}$$

in terms of the data-type variable

$$h \triangleq \begin{cases} 1, & \text{when } w \text{ is real} \\ 2, & \text{when } w \text{ is complex} \end{cases} \tag{4.157}$$

As was the case in the real domain, we continue to assume that the now $2M \times 2M$ Hessian Hessian matrix of $J(w)$ satisfies the local Lipschitz condition (4.18).

We also explained that the constant step-size stochastic gradient recursion is given by

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \mu\,\widehat{\nabla_{w^*} J}(\boldsymbol{w}_{i-1}), \quad i \geq 0 \tag{4.158}$$

and that the gradient noise process is now complex-valued as well, i.e.,

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;\triangleq\; \widehat{\nabla_{w^*} J}(\boldsymbol{w}_{i-1}) \;-\; \nabla_{w^*} J(\boldsymbol{w}_{i-1}) \tag{4.159}$$

The first and second-order moments of this noise process are assumed to satisfy the same conditions in Assumption 3.4. The result in Theorem 4.8 further ahead extends the conclusion from Theorem 4.7 to the complex case. Comparing the performance expressions in the lemma below to the earlier expressions in the real case from Theorem 4.7, we observe that in the MSD case, two moment matrices are now involved, and which are denoted by R_s and R_q. These matrices are defined as follows.

For any $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$, we introduce the extended gradient noise vector of size $2M \times 1$:

$$\boldsymbol{s}_i^e(\boldsymbol{w}) \;\triangleq\; \begin{bmatrix} \boldsymbol{s}_i(\boldsymbol{w}) \\ (\boldsymbol{s}_i^*(\boldsymbol{w}))^{\mathsf{T}} \end{bmatrix} \tag{4.160}$$

where we are using the superscript "e" to denote the extended variable. We then let

$$R_{s,i}^e(\boldsymbol{w}) \;\triangleq\; \mathbb{E}\,[\,\boldsymbol{s}_i^e(\boldsymbol{w})\boldsymbol{s}_i^{e*}(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \tag{4.161}$$

denote the conditional second-order moment of this extended noise process. It is a $2M \times 2M$ matrix whose blocks are given by

$$R_{s,i}^e(\boldsymbol{w}) \;=\; \begin{bmatrix} \mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w})\boldsymbol{s}_i^*(\boldsymbol{w}) & \mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w})\boldsymbol{s}_i^{\mathsf{T}}(\boldsymbol{w}) \\ \mathbb{E}\,\Big(\boldsymbol{s}_i(\boldsymbol{w})\boldsymbol{s}_i^{\mathsf{T}}(\boldsymbol{w})\Big)^* & \mathbb{E}\,(\boldsymbol{s}_i(\boldsymbol{w})\boldsymbol{s}_i^*(\boldsymbol{w}))^{\mathsf{T}} \end{bmatrix} \tag{4.162}$$

Compared with the earlier definition (4.11) in the real case, we see that now two moment quantities of the form $\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w})\boldsymbol{s}_i^*(\boldsymbol{w})$ and $\mathbb{E}\,\boldsymbol{s}_i(\boldsymbol{w})\boldsymbol{s}_i^{\mathsf{T}}(\boldsymbol{w})$ appear in (4.162), with the first one using conjugate transposition and the second one using standard transposition. We assume that, in the limit, these moment matrices tend to constant values when evaluated at w^o and we denote their limits by

$$R_s \;\triangleq\; \lim_{i\to\infty} \mathbb{E}\,[\,\boldsymbol{s}_i(w^o)\boldsymbol{s}_i^*(w^o)\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \tag{4.163}$$

$$R_q \;\triangleq\; \lim_{i\to\infty} \mathbb{E}\,\Big[\,\boldsymbol{s}_i(w^o)\boldsymbol{s}_i^{\mathsf{T}}(w^o)\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] \tag{4.164}$$

Comparing (4.163) with (4.164) we see that $\boldsymbol{s}_i^*(\boldsymbol{w})$ is used in the expression for R_s while $\boldsymbol{s}_i^{\mathsf{T}}(\boldsymbol{w})$ is used in the expression for R_q. The two moment matrices, $\{R_s, R_q\}$, are in general different. It is the first moment, R_s, that is an actual covariance matrix in the complex domain (and is therefore Hermitian and non-negative definite), while the second moment, R_q, is symmetric. Both matrices $\{R_s, R_q\}$ are needed to characterize the second-order moment of $\boldsymbol{s}_i(w^o)$ in the complex domain. When $\boldsymbol{s}_i(w^o)$ happens to be real-valued, then R_s and R_q will obviously coincide. Nevertheless, we will continue to use the universal notation R_s (and not R_q) to denote the covariance matrix of $\boldsymbol{s}_i(w^o)$. In other words, whether $\boldsymbol{s}_i(w^o)$ is real or complex-valued, the notation R_s will always denote its limiting covariance matrix:

$$R_s \;\triangleq\; \begin{cases} \lim\limits_{i\to\infty} \mathbb{E}\,\Big[\,\boldsymbol{s}_i(w^o)\boldsymbol{s}_i^{\mathsf{T}}(w^o)\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] & \text{(for real data)} \\ \\ \lim\limits_{i\to\infty} \mathbb{E}\,[\,\boldsymbol{s}_i(w^o)\boldsymbol{s}_i^*(w^o)\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] & \text{(for complex data)} \end{cases} \tag{4.165}$$

Before establishing the next result, we mention that the smoothness condition (4.19) takes the following form in the complex case in terms of the extended covariance matrix:

$$\Big\|R_{s,i}^e(w^o+\Delta w) - R_{s,i}^e(w^o)\Big\| \;\leq\; \kappa_2\,\|\Delta w\|^{\gamma} \tag{4.166}$$

for small perturbations $\|\Delta w\| \leq \epsilon$, and for some constant $\kappa_2 \geq 0$ and exponent $0 < \gamma \leq 4$.

Theorem 4.8 (Mean-square-error performance: Complex case)**.** Assume the cost function $J(w)$ satisfies conditions (4.156) and (4.18). Assume further that the gradient noise process satisfies the conditions in Assumption 3.4 and the smoothness condition (4.166), and that the step-size is sufficiently small to ensure mean-square stability, as already ascertained by Lemma 3.5. Then, it holds that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; \frac{\mu}{4}\,\mathrm{Tr}\left(H^{-1}\begin{bmatrix} R_s & R_q \\ R_q^* & R_s^{\mathsf{T}} \end{bmatrix}\right) \;+\; O\left(\mu^{1+\gamma_m}\right) \tag{4.167}$$

$$\limsup_{i\to\infty} \mathbb{E}\,\{J(\boldsymbol{w}_{i-1}) - J(w^o)\} \;=\; \frac{\mu}{2}\,\mathrm{Tr}\left(R_s\right) \;+\; O\left(\mu^{1+\gamma_m}\right) \tag{4.168}$$

where

$$\gamma_m \;\triangleq\; \frac{1}{2}\min\{1,\gamma\} > 0 \tag{4.169}$$

and $\gamma \in (0,4]$ is from (4.166). Moreover, $\{R_s, R_q\}$ are defined by (4.163)–(4.164) and $H = \nabla_w^2 J(w^o)$ is $2M \times 2M$. Consequently, the MSD and ER metrics for the complex stochastic-gradient algorithm (4.158) are given by:

$$\mathrm{MSD} \;=\; \frac{\mu}{4}\,\mathrm{Tr}\left(H^{-1}\begin{bmatrix} R_s & R_q \\ R_q^* & R_s^{\mathsf{T}} \end{bmatrix}\right) \tag{4.170}$$

$$\mathrm{ER} \;=\; \frac{\mu}{2}\,\mathrm{Tr}\left(R_s\right) \tag{4.171}$$

Moreover, for $i \gg 1$, the rate at which the error variance, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, approaches its steady-state region is well-approximated to first-order in μ by

$$\alpha = 1 - 2\mu\lambda_{\min}(H) \tag{4.172}$$

When $J(w)$ is quadratic in w, the approximation errors in (4.167)–(4.168) are replaced by $O(\mu^2)$.

Proof. We explained in the proof of Lemma 3.5 that results for the complex recursion (4.158) can be recovered by working with the following recursion in terms of an extended $2M \times 1$ *real* variable $\boldsymbol{v}_i$:

$$\boldsymbol{v}_i \;=\; \boldsymbol{v}_{i-1} \;-\; \mu'\,\widehat{\nabla_{v^{\mathsf{T}}} J}(\boldsymbol{v}_{i-1}) \tag{4.173}$$

where $\mu' = \mu/2$ and $\boldsymbol{v}_i = \mathrm{col}\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ in terms of the real and imaginary parts of $\boldsymbol{w}_i = \boldsymbol{x}_i + j\boldsymbol{y}_i$. The gradient noise process that is associated with this $v-$domain recursion was denoted by

$$\boldsymbol{t}_i(\boldsymbol{v}_{i-1}) \;\triangleq\; \widehat{\nabla_{v^{\mathsf{T}}} J}(\boldsymbol{v}_{i-1}) \;-\; \nabla_{v^{\mathsf{T}}} J(\boldsymbol{v}_{i-1}) \tag{4.174}$$

and it was shown in (3.150) to be given by

$$\boldsymbol{t}_i(\boldsymbol{v}_{i-1}) \;=\; 2\left[\begin{array}{c} \boldsymbol{s}_{R,i}(\boldsymbol{w}_{i-1}) \\ \boldsymbol{s}_{I,i}(\boldsymbol{w}_{i-1}) \end{array}\right] \tag{4.175}$$

in terms of the real and imaginary parts of the original gradient noise vector $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$, defined by (4.159):

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;\triangleq\; \boldsymbol{s}_{R,i}(\boldsymbol{w}_{i-1}) \;+\; j\boldsymbol{s}_{I,i}(\boldsymbol{w}_{i-1}) \tag{4.176}$$

Therefore, in order to apply the results of Theorem 4.7 to the $v-$domain recursion (4.173) under the conditions in Assumption 3.4, we need to determine two quantities:

(a) First, we need to determine an expression for the Hessian matrix of the cost function $J(v)$, in the $v-$domain, which will play the role of the matrix H in expressions (4.100)–(4.101).

(b) Second, we need to determine an expression for the second-order moment of the noise component, $\boldsymbol{t}_i(v^o)$, which will play the role of R_s in the same expressions (4.100)–(4.101).

With regards to the Hessian matrix, we recall result (B.26) from the appendix, which relates the Hessian matrix of $J(v)$ in the $v-$domain to the complex Hessian matrix of $J(w)$ in the $w-$domain, and use it to write

$$\nabla_v^2\, J(v^o) \;=\; D^*\left[\nabla_w^2\, J(w^o)\right] D \;=\; D^*HD \tag{4.177}$$

in terms of the matrix D defined by (B.27) and which satisfies $DD^* = 2I_{2M}$. Note that this result also implies that $\nabla_v^2\, J(v^o)$ is similar to $2H$ so that

$$\lambda_{\min}\left(\nabla_v^2\, J(v^o)\right) \;=\; 2\lambda_{\min}(H) \tag{4.178}$$

With regards to the second-order moment of the absolute component of $\boldsymbol{t}_i(\boldsymbol{v}_{i-1})$, we let

$$R_t \;\triangleq\; \lim_{i\to\infty}\; \mathbb{E}\left[\boldsymbol{t}_i(v^o)\boldsymbol{t}_i^{\mathsf{T}}(v^o)\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] \tag{4.179}$$

Using (4.175), as well as definitions (4.163)–(4.164) for the second-order moments $\{R_s, R_q\}$ associated with the original gradient noise component, $\boldsymbol{s}_i(w^o)$, it can be verified that

$$\begin{aligned} DR_tD^* \;&=\; 4\cdot\lim_{i\to\infty}\; \mathbb{E}\left(\left[\begin{array}{cc} \boldsymbol{s}_i(w^o)\boldsymbol{s}_i^*(w^o) & \boldsymbol{s}_i(w^o)\boldsymbol{s}_i^{\mathsf{T}}(w^o) \\ \big(\boldsymbol{s}_i(w^o)\boldsymbol{s}_i^{\mathsf{T}}(w^o)\big)^* & \big(\boldsymbol{s}_i(w^o)\boldsymbol{s}_i^*(w^o)\big)^{\mathsf{T}} \end{array}\right]\right) \\ &\triangleq\; 4\left[\begin{array}{cc} R_s & R_q \\ R_q^* & R_s^{\mathsf{T}} \end{array}\right] \end{aligned} \tag{4.180}$$

We already know from (3.152)–(3.153) and (3.168) that the second and fourth-order moments of the gradient noise process $\boldsymbol{t}_i(\boldsymbol{v}_{i-1})$ satisfy conditions similar to (4.9)–(4.10) and (4.67) in the real case. Therefore, the results of Theorem 4.7 can be applied to the $v-$domain recursion (4.173). Let

$$m \triangleq 1+\gamma_m \tag{4.181}$$

We conclude from the expressions in Theorem 4.7 that the limit superior for each of the error variance and the mean fluctuation for the $v-$domain recursion are given by (using $\mu' = \mu/2$)

$$\begin{aligned}
\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{v}}_i\|^2 &= \frac{\mu'}{2}\,\mathrm{Tr}\left(\left[\nabla_v^2\, J(v^o)\right]^{-1} R_t\right) + O((\mu')^m) \\
&= \frac{\mu}{4}\,\mathrm{Tr}\left(D^{-1}H^{-1}D^{-*}R_t\right) + O(\mu^m) \\
&= \frac{\mu}{4}\,\mathrm{Tr}\left(H^{-1}D^{-*}R_tD^{-1}\right) + O(\mu^m) \\
&= \frac{\mu}{4}\,\mathrm{Tr}\left(H^{-1}\frac{1}{2}DR_t\frac{1}{2}D^{*}\right) + O(\mu^m) \\
&= \frac{\mu}{4}\,\mathrm{Tr}\left(H^{-1}\left[\begin{array}{cc} R_s & R_q \\ R_q^{*} & R_s^{\mathsf{T}} \end{array}\right]\right) + O(\mu^m)
\end{aligned} \tag{4.182}$$

and

$$\begin{aligned}
\limsup_{i\to\infty} \mathbb{E}\,\{J(\boldsymbol{v}_{i-1}) - J(v^o)\} &= \frac{\mu'}{4}\,\mathrm{Tr}\,(R_t) + O((\mu')^m) \\
&= \frac{\mu}{8}\,\mathrm{Tr}\left(D^{-1}DR_t\right) + O(\mu^m) \\
&= \frac{\mu}{8}\,\mathrm{Tr}\left(DR_tD^{-1}\right) + O(\mu^m) \\
&= \frac{\mu}{16}\,\mathrm{Tr}\,(DR_tD^{*}) + O(\mu^m) \\
&= \frac{\mu}{4}\,\mathrm{Tr}\left(\left[\begin{array}{cc} R_s & R_q \\ R_q^{*} & R_s^{\mathsf{T}} \end{array}\right]\right) + O(\mu^m) \\
&= \frac{\mu}{2}\,\mathrm{Tr}\,(R_s) + O(\mu^m)
\end{aligned} \tag{4.183}$$

Finally, using (4.172) we conclude that the convergence rate in the $v-$domain is given by the following expression to first-order in μ:

$$\begin{aligned}
\alpha &= 1 - 2\mu'\lambda_{\min}(\nabla_v^2\, J(v^o)) \\
&= 1 - 2\left(\frac{\mu}{2}\right)2\lambda_{\min}(H) \\
&\overset{(4.178)}{=} 1 - 2\mu\lambda_{\min}(H)
\end{aligned} \tag{4.184}$$

□

Example 4.6 (Performance of complex LMS adaptation)**.** We reconsider the complex LMS recursion (3.125) from Example 3.4. In this case we have

$$R_s = \sigma_v^2 R_u, \quad H = \begin{bmatrix} R_u & 0 \\ 0 & R_u^{\mathsf{T}} \end{bmatrix}, \qquad G_k = \sigma_{v,k}^2 \begin{bmatrix} R_u & \times \\ \times & R_u^{\mathsf{T}} \end{bmatrix} \tag{4.185}$$

where the block off-diagonal entries of G_k are not be not needed because H_k is block-diagonal. Substituting into (4.170) and (4.171) we find that the MSD and ER performance levels are given by

$$\text{MSD} = \frac{\mu M \sigma_v^2}{2} \tag{4.186}$$

$$\text{ER} = \frac{\mu \sigma_v^2}{2} \operatorname{Tr}(R_s) \tag{4.187}$$

■

It is useful to remark that the block matrix that appears in expression (4.170) for the MSD is equal to the limiting covariance matrix of the extended gradient noise vector when evaluated at $w = w^o$:

$$\boldsymbol{s}_i^e(w^o) \triangleq \begin{bmatrix} \boldsymbol{s}_i(w^o) \\ (\boldsymbol{s}_i^*(w^o))^{\mathsf{T}} \end{bmatrix} \tag{4.188}$$

Specifically, it holds that

$$\begin{bmatrix} R_s & R_q \\ R_q^* & R_s^{\mathsf{T}} \end{bmatrix} = \lim_{i \to \infty} \mathbb{E}\, [\, \boldsymbol{s}_i^e(w^o)\, (\boldsymbol{s}_i^e(w^o))^* \mid \boldsymbol{\mathcal{F}}_{i-1} \,] \triangleq R_s^e \tag{4.189}$$

If we use R_s^e to denote this extended covariance matrix, then we can rewrite the MSD and ER expressions (4.170)–(4.171) in the equivalent forms:

$$\text{MSD} = \frac{\mu}{4} \operatorname{Tr}\left(H^{-1} R_s^e\right) \tag{4.190}$$

$$\text{ER} = \frac{\mu}{4} \operatorname{Tr}(R_s^e) \tag{4.191}$$

5

Centralized Adaptation and Learning

The discussion in the last two chapters established the mean-square stability of stand-alone adaptive agents for small constant step-sizes (Lemmas 3.1 and 3.5), and provided expressions for their MSD and ER metrics (Theorems 4.7 and 4.8) for both cases of real and complex-valued data. In this chapter, and in preparation for our treatment of networked agents in future chapters, we examine two situations involving a *multitude* of similar agents behaving in one of two modes [207]. In the first scenario, each agent senses data and analyzes it independently of the other agents. We refer to this mode of operation as *non-cooperative* processing. In the second scenario, the agents transmit the collected data for processing at a fusion center. We refer to this mode of operation as *centralized* or batch processing. We motivate the discussion by considering first the case of mean-square-error costs. Subsequently, we extend the results to more general costs.

5.1 Non-Cooperative Processing

Thus, consider *separate* agents, labeled $k = 1, 2, \ldots, N$. Following the framework discussed in Examples 3.1 and 3.4 on LMS adaptation in

the real and complex domains, each agent, k, receives streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i},\, i \geq 0\}$, where we are using the subscript k to index the data at agent k. We treat the real and complex data cases uniformly by using the data-type variable in the expressions that follow:

$$h \;\triangleq\; \begin{cases} 1 & \text{(real data)} \\ 2 & \text{(complex data)} \end{cases} \tag{5.1}$$

We assume the data at each agent satisfies the same statistical properties as in Examples 3.1 and 3.4, and the same linear regression model (3.119) with a common w^o albeit with noise $\boldsymbol{v}_k(i)$:

$$\boldsymbol{d}_k(i) \;=\; \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i), \qquad k = 1, 2, \ldots, N \tag{5.2}$$

We denote the statistical moments of the data at agent k by

$$\sigma_{v,k}^2 = \mathbb{E}\,|\boldsymbol{v}_k(i)|^2 \tag{5.3}$$

and

$$R_{u,k} \;\triangleq\; \begin{cases} \mathbb{E}\,\boldsymbol{u}_{k,i}^{\mathsf{T}} \boldsymbol{u}_{k,i} > 0 & \text{(real data)} \\ \mathbb{E}\,\boldsymbol{u}_{k,i}^{*} \boldsymbol{u}_{k,i} > 0 & \text{(complex data)} \end{cases} \tag{5.4}$$

We further assume in this motivating section that the $R_{u,k}$ are uniform across the agents so that

$$R_{u,k} \equiv R_u, \qquad k = 1, 2, \ldots, N \tag{5.5}$$

In this way, the mean-square-error cost,

$$J_k(w) \;\triangleq\; \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}|^2 \tag{5.6}$$

which is associated with agent k, will satisfy a condition similar to (3.114), namely,

$$0 < \frac{\nu}{h} I_{hM} \;\leq\; \nabla_w^2\, J_k(w) \;\leq\; \frac{\delta}{h} I_{hM} \tag{5.7}$$

with the corresponding parameters $\{\nu, \delta\}$ given by (cf. (2.19)):

$$\nu = 2\lambda_{\min}(R_u), \quad \delta = 2\lambda_{\max}(R_u) \tag{5.8}$$

Now, assume each agent estimates w^o by running the LMS learning rule, say, (3.13) for real data or (3.125) for complex data, which we can describe uniformly in terms of the single recursion:

$$\boldsymbol{w}_{k,i} \;=\; \boldsymbol{w}_{k,i-1} \;+\; \frac{2\mu}{h} \boldsymbol{u}_{k,i}^{*} [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}], \quad i \geq 0 \tag{5.9}$$

using the data-type variable, h, and with the understanding that complex conjugation, $\boldsymbol{u}_{k,i}^*$, is replaced by real transposition, $\boldsymbol{u}_{k,i}^{\mathsf{T}}$, when the data are real. Then, according to (4.146) and (4.186), each agent k will attain an individual MSD level that is given by

$$\mathrm{MSD}_{\mathrm{ncop},k} \;=\; \frac{\mu}{h} M\,\sigma_{v,k}^2, \qquad k = 1, 2, \ldots, N \tag{5.10}$$

Moreover, according to (3.38) and (3.142), each agent k will converge towards this level at a rate dictated by:

$$\alpha_{\mathrm{ncop},k} \;=\; 1 - \frac{4\mu}{h}\lambda_{\min}(R_u) \tag{5.11}$$

If we average the performance level (5.10) across the N agents, we find that the average MSD metric is given by

$$\mathrm{MSD}_{\mathrm{ncop,av}} \;=\; \frac{\mu}{h} M \left(\frac{1}{N}\sum_{k=1}^{N}\sigma_{v,k}^2\right) \tag{5.12}$$

in terms of the average noise power across the agents.

The subscript "ncop" is used in (5.10)–(5.12) to indicate that these expressions are for the non-cooperative mode of operation. It is seen from (5.10) that agents with noisier data (i.e., larger $\sigma_{v,k}^2$) will perform worse and have larger MSD levels than agents with cleaner data. In other words, whenever adaptive agents act individually, the quality of their solution will be as good as the quality of their noisy data.

This is a sensible conclusion and it is illustrated numerically in Figure 5.1. The figure plots the ensemble-average learning curves, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, for two agents. The curves are generated by averaging the trajectories $\{\|\widetilde{\boldsymbol{w}}_{k,i}\|^2\}$ over 2000 repeated experiments. The label on the vertical axis in the figure refers to the learning curves by writing $\mathrm{MSD}(i)$, with an iteration index i. Each experiment involves running the non-cooperative LMS recursion (5.9) on complex-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ generated according to the model $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i)$ with $M = 10$, $R_u = 2I_M$, and $\mu = 0.005$. The noise variances are set to $\sigma_{v,1}^2 = 0.032$ and $\sigma_{v,2}^2 = 0.010$. The noise and regressor processes are both Gaussian distributed in this simulation. The unknown vector w^o is generated randomly and its norm is normalized to one. It is seen

in the figure that the learning curves by the agents tend to the MSD levels predicted by the theoretical expression (5.10).

We are going to show in later chapters that cooperation among agents, whereby agents share information with their neighbors, can help enhance their individual performance levels. The analysis will show that both types of agents can benefit from cooperation: agents with "bad" data and agents with "good" data; this is because all data carry information about w^o. However, for these conclusions to hold, it is necessary for cooperation to be carried out in proper ways — see Chapter 12.

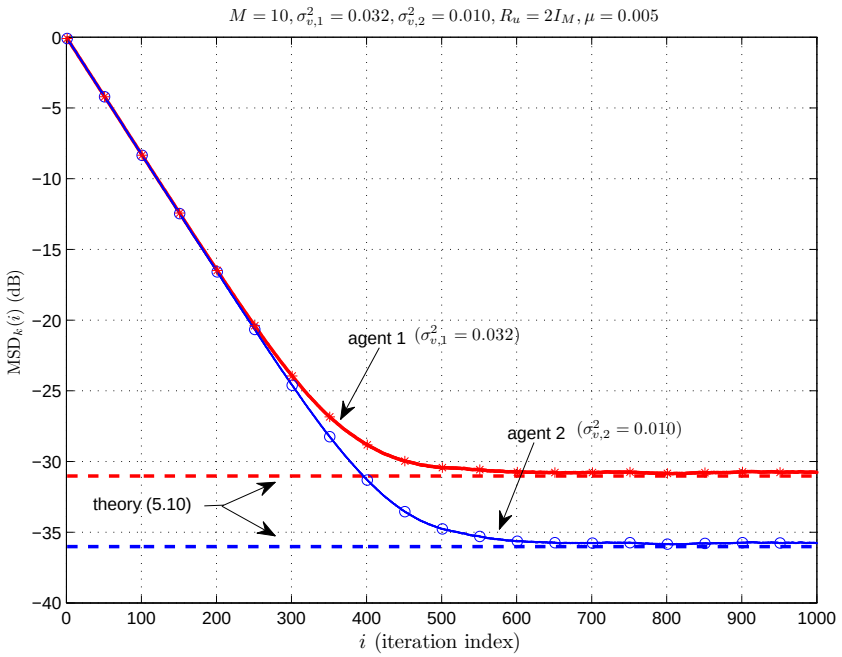

Figure 5.1: Learning curves for two non-cooperative agents running (5.9) on complex data. The curves are obtained by averaging over 2000 repeated experiments using $M = 10$, $\sigma^2_{v,1} = 0.032$, $\sigma^2_{v,2} = 0.010$, $R_u = 2I_M$ and $\mu = 0.005$. The horizontal dashed lines indicate the steady-state MSD levels predicted by the theoretical expression (5.10) for complex data ($h = 2$).

5.2 Centralized Processing

Let us now contrast the above non-cooperative solution with a centralized implementation whereby, at every iteration i, the N agents transmit their raw data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ to a fusion center for processing. One could also consider situations where agents transmit processed data, e.g., as happens with useful techniques for combining adaptive filter outputs [10]. Once the fusion center receives the raw data, we assume it runs a stochastic-gradient update of the form:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu \left(\frac{1}{N} \sum_{k=1}^{N} \frac{2}{h} \boldsymbol{u}_{k,i}^{*} (\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{i-1}) \right) \tag{5.13}$$

where the term between parentheses multiplying μ can be interpreted as corresponding to the sample average of several approximate gradient vectors; one for the data originating from each agent, since

$$\widehat{\nabla_{w^{\mathsf{T}}} J}_k(\boldsymbol{w}_{i-1}) = 2\boldsymbol{u}_{k,i}^{\mathsf{T}} (\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{i-1}) \quad \text{(real data)} \tag{5.14}$$

and

$$\widehat{\nabla_{w^{*}} J}_k(\boldsymbol{w}_{i-1}) = \boldsymbol{u}_{k,i}^{*} (\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{i-1}) \quad \text{(complex data)} \tag{5.15}$$

The analysis in the sequel will show that the MSD performance that results from implementation (5.13) is given by (using future expression (5.65) with the identifications $H_k = 2R_u/h$ and $R_{s,k} = 4\sigma_{v,k}^2 R_u/h^2$):

$$\text{MSD}_{\text{cent}} \;=\; \frac{\mu}{h} M \frac{1}{N} \left(\frac{1}{N} \sum_{k=1}^{N} \sigma_{v,k}^2 \right) \tag{5.16}$$

Moreover, using expression (5.60) given further ahead, this centralized solution will converge towards the above MSD level at the same rate (5.11) as the non-cooperative solution:

$$\alpha_{\text{cent}} \;=\; 1 - \frac{4\mu}{h} \lambda_{\min}(R_u) \tag{5.17}$$

Observe from (5.16) that the MSD level attained by the centralized solution is proportional to $1/N$ times the *average* noise power across all non-cooperative agents in (5.10). At least two conclusions follow from this observation.

First, comparing (5.16) with the average performance (5.12) in the non-cooperative case, we observe that the centralized solution provides an $N-$fold improvement in MSD performance in the mean-square-error case. Figure 5.2 illustrates this situation numerically.

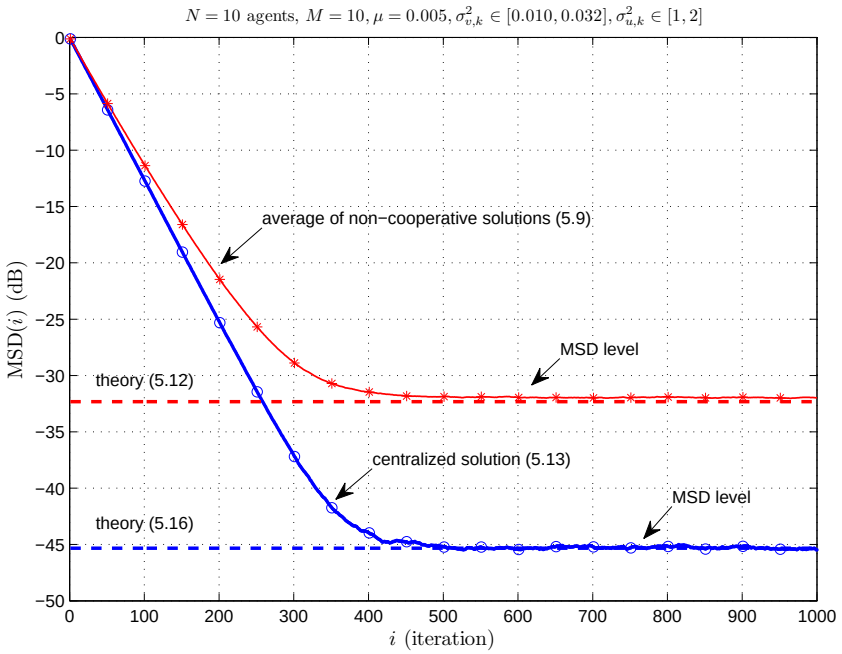

Figure 5.2: Learning curves for the centralized LMS solution (5.13) and for the average of the non-cooperative solution (5.9) over $N = 20$ agents. The curves are obtained by averaging over 2000 repeated experiments using $M = 10$, $\sigma_v^2 \in [0.010, 0.032]$, $R_u = \sigma_{u,k}^2 I_M$ with $\sigma_{u,k}^2 \in [1, 2]$, and $\mu = 0.005$. The horizontal dashed lines indicate the steady-state MSD levels predicted by the theoretical expressions (5.12) and (5.16) for complex data ($h = 2$).

The figure plots two ensemble-average learning curves. One curve represents the evolution of the variance $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ for the centralized solution and is generated by averaging the trajectories $\{\|\widetilde{\boldsymbol{w}}_i\|^2\}$ over 200 repeated experiments. The second ensemble-average curve is obtained by averaging the individual learning curves, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, of all N non-cooperative agents. Again, a total of 2000 repeated experiments are

used to generate each individual learning curve. The label on the vertical axis in the figure refers to the learning curves by writing $\text{MSD}(i)$, with an iteration index i. Each experiment involves running either the centralized LMS recursion (5.13) or the non-cooperative recursion (5.9) on complex-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ generated according to the model $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i)$ with $N = 20$ agents, $M = 10$, and $\mu = 0.005$. The noise variances, $\{\sigma_{v,k}^2\}$, are chosen randomly from within the range $[0.010, 0.032]$, while the covariance matrices are chosen of the form $R_{u,k} = \sigma_{u,k}^2 I_M$ with $\sigma_{u,k}^2$ chosen randomly within the range $[1, 2]$. The noise and regressor processes are both Gaussian distributed in this simulation. The unknown vector w^o is generated randomly and its norm is normalized to one. It is seen in the figure that the learning curve by the centralized solution tends to an MSD level that is $N-$fold superior to the average non-cooperative solution; this translates into the difference of $10\log_{10}(N) \approx 13$dB seen in the figure between the two dashed horizontal lines.

The second observation that follows from (5.16) is that, although the centralized solution outperforms the averaged non-cooperative performance, it does not generally hold that the centralized solution outperforms each individual non-cooperative agent [276]. This is because the average noise power is scaled by $1/N$ in (5.16), and this scaled power can be larger than some of the individual noise variances and smaller than the remaining noise variances. For example, consider a situation with $N = 2$ agents, $\sigma_{v,2}^2 = 5\sigma_v^2$ and $\sigma_{v,1}^2 = \sigma_v^2$. Then,

$$\frac{1}{N}\left(\frac{1}{N}\sum_{k=1}^{N}\sigma_{v,k}^2\right) \;=\; 1.5\sigma_v^2 \tag{5.18}$$

which is larger than $\sigma_{v,1}^2$ and smaller than $\sigma_{v,2}^2$. In this case, the centralized solution (5.16) performs better than non-cooperative agent 2 (i.e., leads to a smaller MSD) but worse than non-cooperative agent 1.

5.3 Stochastic-Gradient Centralized Solution

The last two sections focused on mean-square-error adaptation. Next, we extend the conclusions to more general costs. Thus, consider a collection of N agents, each with an individual twice-differentiable convex

cost function, $J_k(w)$. The objective is to determine the unique minimizer w^o of the aggregate cost:

$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} J_k(w) \tag{5.19}$$

It is now the above aggregate cost, $J^{\text{glob}}(w)$, that will be required to satisfy conditions similar to (4.4) and (4.18) relative to some parameters $\{\nu_c, \delta_c, \kappa_c\}$, with the subscript "c" used to indicate that these factors correspond to the centralized implementation.

Assumption 5.1 (Conditions on aggregate cost function)**.** The aggregate cost function, $J^{\text{glob}}(w)$, is twice-differentiable and satisfies

$$0 < \frac{\nu_c}{h} I_{hM} \leq \nabla_w^2 J^{\text{glob}}(w) \leq \frac{\delta_c}{h} I_{hM} \tag{5.20}$$

for some positive parameters $\nu_c \leq \delta_c$. Condition (5.20) is equivalent to requiring $J^{\text{glob}}(w)$ to be ν_c-strongly convex and for its gradient vector to be δ_c-Lipschitz. In addition, it is assumed that the aggregate cost is smooth enough so that its Hessian matrix is locally Lipschitz continuous in a small neighborhood around $w = w^o$, i.e.,

$$\left\|\nabla_w^2 J^{\text{glob}}(w^o + \Delta w) - \nabla_w^2 J^{\text{glob}}(w^o)\right\| \leq \kappa_c \|\Delta w\| \tag{5.21}$$

for small perturbations $\|\Delta w\| \leq \epsilon$ and for some $\kappa_c \geq 0$.

Under these conditions, the cost $J^{\text{glob}}(w)$ will have a unique minimizer, which we continue to denote by w^o. We will not be requiring each individual cost, $J_k(w)$, to be strongly convex. It is sufficient for at least one of these costs to be strongly convex while the remaining costs can be simply convex; this condition ensures the strong convexity of $J^{\text{glob}}(w)$. Moreover, minimizers of the individual costs $\{J_k(w)\}$ need not coincide with each other or with w^o; we shall write w_k^o to refer to a minimizer of $J_k(w)$.

There are many centralized solutions that can be used to determine the unique minimizer w^o of (5.19), with some solution techniques being more powerful than other techniques. Nevertheless, we shall focus on centralized implementations of the *stochastic gradient* type. The reason

we consider the *same* class of stochastic gradient algorithms for non-cooperative, centralized, and distributed solutions in this work is to enable a *meaningful* comparison among the various implementations. Thus, we consider a centralized strategy of the following form:

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \frac{\mu}{N}\sum_{k=1}^{N}\widehat{\nabla_{w^*}J_k}(\boldsymbol{w}_{i-1}), \quad i\geq 0 \tag{5.22}$$

in terms of approximations for the individual gradient vectors at $\boldsymbol{w}_{i-1}$. Here, again, we will be treating the case of real and complex data jointly. For this reason, although we are computing the gradient vector relative to w^* in the above recursion, it is to be understood that this step should be replaced by differentiation relative to w^{T} in the real case; i.e., complex conjugation should be replaced by real transposition when the data are real in which case the update would take the form:

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \frac{\mu}{N}\sum_{k=1}^{N}\widehat{\nabla_{w^{\mathsf{T}}}J_k}(\boldsymbol{w}_{i-1}), \quad i\geq 0 \tag{5.23}$$

5.4 Gradient Noise Model

Continuing with the general form (5.22), we note that the sum multiplying μ/N is an approximation for the true gradient vector of $J^{\mathrm{glob}}(w)$; the scaling of μ by N in (5.22) is meant to ensure similar convergence rates for the non-cooperative and centralized solutions — as explained further ahead in (5.78). We introduce the *individual* gradient noise processes:

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1}) \;\;\stackrel{\Delta}{=}\;\; \widehat{\nabla_{w^*}J_k}(\boldsymbol{w}_{i-1}) \;-\; \nabla_{w^*}J_k(\boldsymbol{w}_{i-1}) \tag{5.24}$$

for $k=1,2,\ldots,N$, and note that the overall gradient noise corresponding to (5.22) is given by:

$$\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;\;=\;\; \sum_{k=1}^{N}\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1}) \tag{5.25}$$

We also introduce the covariance matrices of the individual noise processes. Specifically, for any $\boldsymbol{w}\in\boldsymbol{\mathcal{F}}_{i-1}$ and for every $k=1,2,\ldots,N$, we

define the extended gradient noise vector of size $2M \times 1$:

$$\boldsymbol{s}^{e}_{k,i}(\boldsymbol{w}) \triangleq \begin{bmatrix} \boldsymbol{s}_{k,i}(\boldsymbol{w}) \\ \big(\boldsymbol{s}^{*}_{k,i}(\boldsymbol{w})\big)^{\mathsf{T}} \end{bmatrix} \tag{5.26}$$

and denote its conditional covariance matrix by

$$R^{e}_{s,k,i}(\boldsymbol{w}) \triangleq \mathbb{E}\left[\boldsymbol{s}^{e}_{k,i}(\boldsymbol{w})\boldsymbol{s}^{e*}_{k,i}(\boldsymbol{w}) \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \tag{5.27}$$

We further assume that, in the limit, the following moment matrices tend to constant values when evaluated at w^o:

$$R_{s,k} \triangleq \lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{s}_{k,i}(w^o)\boldsymbol{s}^{*}_{k,i}(w^o) \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \tag{5.28}$$

$$R_{q,k} \triangleq \lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{s}_{k,i}(w^o)\boldsymbol{s}^{\mathsf{T}}_{k,i}(w^o) \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \tag{5.29}$$

We define similar quantities for the aggregate noise process (5.25) and denote them by

$$R^{e}_{s,i}(\boldsymbol{w}) \triangleq \mathbb{E}\left[\boldsymbol{s}^{e}_{i}(\boldsymbol{w})\boldsymbol{s}^{e*}_{i}(\boldsymbol{w}) \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \tag{5.30}$$

$$R_{s} \triangleq \lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{s}_{i}(w^o)\boldsymbol{s}^{*}_{i}(w^o) \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \tag{5.31}$$

$$R_{q} \triangleq \lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{s}_{i}(w^o)\boldsymbol{s}^{\mathsf{T}}_{i}(w^o) \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \tag{5.32}$$

Now since the centralized iteration (5.22) has the form of a stochastic gradient recursion, we should be able to infer its mean-square-error behavior from Lemma 3.5 and Theorem 4.8 if the aggregate noise process (5.25) satisfies conditions similar to Assumption 3.4. It is straightforward to verify that this is possible, for example, if the *individual* components satisfy conditions similar to Assumption 3.4 and condition (4.67) and when, additionally, these individual components are uncorrelated with each other and second-order circular as described by the following statement.

Assumption 5.2 (Conditions on gradient noise)**.** It is assumed that the first and fourth-order conditional moments of the individual gradient noise processes, $\boldsymbol{s}_{k,i}(\boldsymbol{w})$, defined by (5.24) satisfy the following conditions for any iterates $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$ and for all $k, \ell = 1, 2, \ldots, N$:

$$\mathbb{E}\,[\,\boldsymbol{s}_{k,i}(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{5.33}$$

$$\mathbb{E}\,\big[\boldsymbol{s}_{k,i}(\boldsymbol{w})\boldsymbol{s}_{\ell,i}^{*}(\boldsymbol{w})|\boldsymbol{\mathcal{F}}_{i-1}\big] \;=\; 0,\quad k\neq\ell \tag{5.34}$$

$$\mathbb{E}\,\big[\boldsymbol{s}_{k,i}(\boldsymbol{w})\boldsymbol{s}_{\ell,i}^{\mathsf{T}}(\boldsymbol{w})|\boldsymbol{\mathcal{F}}_{i-1}\big] \;=\; 0,\quad k\neq\ell \tag{5.35}$$

$$\mathbb{E}\,[\,\|\boldsymbol{s}_{k,i}(\boldsymbol{w})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;\leq\; \big(\bar{\beta}_k/h\big)^4\,\|\boldsymbol{w}\|^4 \;+\; \bar{\sigma}_{s,k}^4 \tag{5.36}$$

almost surely, for some nonnegative scalars $\bar{\beta}_k^4$ and $\bar{\sigma}_{s,k}^4$ and where $h=2$ for complex data and $h=1$ for real data. We also assume that the conditional second-order moments of the aggregate noise process satisfies a smoothness condition similar to (4.166), namely,

$$\big\|R_{s,i}^{e}(w^o+\Delta w)-R_{s,i}^{e}(w^o)\big\| \;\leq\; \kappa_{c,2}\,\|\Delta w\|^{\gamma} \tag{5.37}$$

in terms of the extended covariance matrix, for small perturbations $\|\Delta w\|\leq\epsilon$, and for some constants $\kappa_{c,2}\geq 0$ and exponent $0<\gamma\leq 4$.

It is straightforward to verify from conditions (5.34)–(5.35) that

$$R_s \;=\; \sum_{k=1}^{N} R_{s,k} \tag{5.38}$$

$$R_q \;=\; \sum_{k=1}^{N} R_{q,k} \tag{5.39}$$

Moreover, in a manner similar to (3.134), we conclude from (5.36) that the second-order moments of the individual gradient noise processes satisfy:

$$\mathbb{E}\,\Big[\,\|\boldsymbol{s}_{k,i}(\boldsymbol{w})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\Big] \;\leq\; \Big(\bar{\beta}_k/h\Big)^2\,\|\boldsymbol{w}\|^2 \;+\; \bar{\sigma}_{s,k}^2 \tag{5.40}$$

Using this condition, along with (5.33), it is again straightforward to verify that the aggregate noise satisfies

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{5.41}$$

$$\mathbb{E}\,\Big[\|\boldsymbol{s}_i(\boldsymbol{w})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\Big] \;\leq\; \frac{1}{h^2}\left(\sum_{k=1}^{N}\bar{\beta}_k^2\right)\|\boldsymbol{w}\|^2+\sum_{k=1}^{N}\bar{\sigma}_{s,k}^2 \tag{5.42}$$

so that repeating argument (3.28) we can deduce that

$$\mathbb{E}\,\Big[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\Big] \;\leq\; (\beta_c/h)^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \sigma_s^2 \tag{5.43}$$

where $\widetilde{\boldsymbol{w}}_{i-1} = w^o - \boldsymbol{w}_{i-1}$, and

$$\beta_c^2 \;\triangleq\; 2\left(\sum_{k=1}^{N}\bar{\beta}_k^2\right) \tag{5.44}$$

$$\sigma_s^2 \;\triangleq\; 2\left(\sum_{k=1}^{N}\bar{\beta}_k^2\right)\|w^o\|^2 \;+\; \sum_{k=1}^{N}\bar{\sigma}_{s,k}^2 \tag{5.45}$$

Likewise, we can conclude from (5.36) that

$$\mathbb{E}\left[\,\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\right] \;\leq\; (\beta_{4,k}/h)^4\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; \sigma_{s4,k}^4 \tag{5.46}$$

in terms of the scalars

$$\beta_{4,k}^4 \;\triangleq\; 8\bar{\beta}_k^4 \tag{5.47}$$

$$\sigma_{s4,k}^4 \;\triangleq\; 8(\bar{\beta}_{4,k}^4/h^4)\|w^o\|^4 \;+\; \bar{\sigma}_{s,k}^4 \tag{5.48}$$

By extrapolation, we also conclude that the fourth-order moment of the aggregate noise, $\boldsymbol{s}_i(\boldsymbol{w}_{i-1})$, is similarly bounded. More explicitly, it will hold that

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4|\boldsymbol{\mathcal{F}}_{i-1}\right] \;&\leq\; N^3\left(\sum_{k=1}^{N}(\beta_{4,k}^4/h^4)\right)\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 + N^3\left(\sum_{k=1}^{N}\sigma_{s4,k}^4\right)\\ &\triangleq\; (\beta_a/h)^4\|\widetilde{\boldsymbol{w}}_{i-1}\|^4 \;+\; \sigma_a^4\end{aligned} \tag{5.49}$$

for some nonnegative constants β_a^4 and σ_a^4. This can be seen as follows. Exploiting the convexity of the norm function $f(x) = \|x\|^4$ and using Jensen's inequality (F.26) we can write

$$\begin{aligned}\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4 \;&\overset{(5.25)}{=}\; \left\|\sum_{k=1}^{N}\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\right\|^4\\ &=\; \left\|\sum_{k=1}^{N}\frac{1}{N}N\,\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\right\|^4\\ &\overset{(F.26)}{\leq}\; \frac{1}{N}\sum_{k=1}^{N}N^4\,\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\|^4\\ &\leq\; N^3\left(\sum_{k=1}^{N}\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\|^4\right)\end{aligned} \tag{5.50}$$

from which we conclude that

$$\mathbb{E}\left[\,\|\boldsymbol{s}_i(\boldsymbol{w}_{i-1})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] \leq N^3\left(\sum_{k=1}^{N}\mathbb{E}\left[\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1})\|^4\,|\boldsymbol{\mathcal{F}}_{i-1}\right]\right) \quad (5.51)$$

and result (5.49) follows.

5.5 Performance of Centralized Solution

Motivated by the discussion that led to expressions (4.94) and (4.95) for the MSD and ER metrics in the single agent case, we similarly define the MSD and ER performance measures for the centralized solution as follows:

$$\mathrm{MSD}_{\mathrm{cent}} \triangleq \mu\cdot\left(\lim_{\mu\to 0}\limsup_{i\to\infty}\frac{1}{\mu}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2\right) \quad (5.52)$$

$$\mathrm{ER}_{\mathrm{cent}} \triangleq \frac{\mu}{N}\cdot\left(\lim_{\mu\to 0}\limsup_{i\to\infty}\frac{1}{\mu}\mathbb{E}\left\{J^{\mathrm{glob}}(\boldsymbol{w}_{i-1})-J^{\mathrm{glob}}(w^o)\right\}\right) \quad (5.53)$$

where the scaling by $1/N$ in (5.53) is meant to ensure that $\mathrm{ER}_{\mathrm{cent}}$ is compatible with the definition used for non-cooperative agents in (4.95) and later for multi-agent networks in (11.34). For example, when the individual costs happen to coincide, say, $J_k(w)\equiv J(w)$ for $k=1,2,\ldots,N$, then the aggregate cost (5.19) reduces to $J^{\mathrm{glob}}(w)=N\,J(w)$ and expression (5.53) becomes consistent with the earlier expression (4.95). Note that we are adding the subscript "cent" to indicate that the above MSD and ER measures are associated with the centralized solution. As explained earlier in Sec. 4.5, we sometimes rewrite the above definitions for the MSD and ER measures more compactly (but less rigorously) as

$$\mathrm{MSD}_{\mathrm{cent}} = \lim_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \quad (5.54)$$

$$\mathrm{ER}_{\mathrm{cent}} = \lim_{i\to\infty}\frac{1}{N}\mathbb{E}\left\{J^{\mathrm{glob}}(\boldsymbol{w}_{i-1})-J^{\mathrm{glob}}(w^o)\right\} \quad (5.55)$$

with the understanding that the limits on the right-hand side in the above two expressions are computed according to the definitions (5.52)–(5.53).

The conclusions in the next theorem now follow from Lemma 3.5 and Theorem 4.8. The performance expressions given in the theorem are expressed in terms of the following quantities, defined for both cases of real or complex data.

Definition 5.1 (Hessian and moment matrices)**.** We associate with each agent k a pair of matrices $\{H_k, G_k\}$, both of which are evaluated at the location of the minimizer $w = w^o$. The matrices are defined as follows:

$$H_k \triangleq \nabla_w^2 J_k(w^o), \qquad G_k \triangleq \begin{cases} R_{s,k} & \text{(real case)} \\ \begin{bmatrix} R_{s,k} & R_{q,k} \\ R_{q,k}^* & R_{s,k}^{\mathsf{T}} \end{bmatrix} & \text{(complex case)} \end{cases} \tag{5.56}$$

Both matrices are dependent on the data type (whether real or complex); in particular, each is $2M \times 2M$ for complex data and $M \times M$ for real data. Note that $H_k \geq 0$ and $G_k \geq 0$.

In view of the lower bound condition in (5.20), it follows that

$$\sum_{k=1}^{N} H_k \; > \; 0 \tag{5.57}$$

so that the sum of the $\{H_k\}$ matrices is invertible. This matrix sum appears in the performance expressions below.

Theorem 5.1 (Performance of centralized solution)**.** Assume the aggregate cost (5.19) satisfies condition (5.20) for some parameters $0 < \nu_c \leq \delta_c$. Assume also that the gradient noise processes satisfy conditions (5.40)–(5.33). For any μ satisfying

$$\frac{\mu}{hN} < \frac{2\nu_c}{\delta_c^2 + \beta_c^2} \tag{5.58}$$

it holds that

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;\leq\; \alpha\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \;+\; \left(\frac{\mu}{N}\right)^2 \sigma_s^2 \tag{5.59}$$

where the parameters $\{\sigma_s^2, \beta_c^2\}$ are defined by (5.44)–(5.45), and where the scalar α satisfies $0 \leq \alpha < 1$ and is given by

$$\alpha \;=\; 1 - 2\nu_c\left(\frac{\mu}{hN}\right) \;+\; (\delta_c^2 + \beta_c^2)\left(\frac{\mu}{hN}\right)^2 \tag{5.60}$$

It follows from (5.59) that for sufficiently small step-sizes:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O(\mu) \tag{5.61}$$

Moreover, under the additional smoothness conditions (5.21) on $J^{\rm glob}(w)$ and (5.37) on the individual noise covariance matrices, it holds that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 = \mathrm{MSD}_{\rm cent} + O\left(\mu^{1+\gamma_m}\right) \tag{5.62}$$

$$\limsup_{i\to\infty} \frac{1}{N}\mathbb{E}\left\{J^{\rm glob}(\boldsymbol{w}_{i-1}) - J^{\rm glob}(w^o)\right\} = \mathrm{ER}_{\rm cent} + O\left(\mu^{1+\gamma_m}\right) \tag{5.63}$$

where

$$\gamma_m \;\triangleq\; \frac{1}{2}\min\{1,\gamma\} > 0 \tag{5.64}$$

with $\gamma \in (0,4]$ from (5.37), and where

$$\mathrm{MSD}_{\rm cent} \;=\; \frac{\mu}{2hN}\,\mathrm{Tr}\left[\left(\sum_{k=1}^{N} H_k\right)^{-1}\left(\sum_{k=1}^{N} G_k\right)\right] \tag{5.65}$$

$$\mathrm{ER}_{\rm cent} \;=\; \frac{\mu h}{4N^2}\,\mathrm{Tr}\left(\sum_{k=1}^{N} R_{s,k}\right) \tag{5.66}$$

The N^2 factor in the denominator of (5.66) is because of the normalization by $1/N$ in the definition (5.53). Moreover, for $i \gg 1$, the rate at which the error variance, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, approaches its steady-state region (5.62) is well-approximated to first-order in μ by

$$\alpha = 1 - \frac{2\mu}{N}\lambda_{\min}\left(\sum_{k=1}^{N} H_k\right) \tag{5.67}$$

If desired, we can relax conditions (5.33)–(5.36) and replace them by requirements on the aggregate noise process (5.25) directly, such as requiring:

$$\mathbb{E}\,[\,\boldsymbol{s}_i(\boldsymbol{w})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] \;=\; 0 \tag{5.68}$$

$$\mathbb{E}\left[\,\|\boldsymbol{s}_i(\boldsymbol{w})\|^4\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\right] \;\le\; (\beta_c/h)^4\,\|\boldsymbol{w}\|^4 \;+\; \sigma_s^4 \tag{5.69}$$

for some nonnegative constants β_c^4 and σ_s^4. Note in particular that these assumptions do not impose the uncorrelatedness and circularity conditions (5.34)–(5.35) on the individual noise processes. We also replace

condition (5.37), which involves the individual agents, by the requirement

$$\left\|R_{s,i}^{e}(w^{o}+\Delta w)-R_{s,i}^{e}(w^{o})\right\| \;\leq\; \kappa_{c,2}\,\|\Delta w\|^{\gamma} \tag{5.70}$$

in terms of the covariance matrix of the extended aggregate noise vector, $\boldsymbol{s}_i^e(\boldsymbol{w})$. Then, the conclusions of Theorem 5.1 will continue to hold using $\{\beta_c^2, \sigma_s^2\}$ from (5.69), and with the sum of the $\{G_k\}$ appearing in (5.65) replaced by

$$G_c \;\stackrel{\Delta}{=}\; \begin{cases} R_s & \text{(real case)} \\ \begin{bmatrix} R_s & R_q \\ R_q^{*} & R_s^{\mathsf{T}} \end{bmatrix} & \text{(complex case)} \end{cases} \tag{5.71}$$

in terms of the moment matrices (5.31)–(5.32) for the aggregate noise process. More specifically, let

$$H_c \;\stackrel{\Delta}{=}\; \sum_{k=1}^{N} H_k \tag{5.72}$$

denote the aggregate Hessian matrix. It will then hold that

$$\mathrm{MSD}_{\mathrm{cent}} \;=\; \frac{\mu}{2hN}\,\mathrm{Tr}\left(H_c^{-1}G_c\right) \tag{5.73}$$

$$\mathrm{ER}_{\mathrm{cent}} \;=\; \frac{\mu h}{8N^2}\,\mathrm{Tr}\left(G_c\right) \tag{5.74}$$

When the individual gradient noise processes satisfy conditions (5.34)–(5.35), it is easy to verify that the moment matrix G_c will be given by

$$G_c \;=\; \sum_{k=1}^{N} G_k \tag{5.75}$$

so that the above MSD and ER expressions reduce to (5.65)–(5.66).

5.6 Comparison with Single Agents

Continuing with the conditions in Assumption 5.2, we now compare the performance of the centralized solution (5.22) to that of non-cooperative processing where agents act independently of each other and run the recursion:

$$\boldsymbol{w}_{k,i} \;=\; \boldsymbol{w}_{k,i-1} - \mu\,\widehat{\nabla_{w^*}J}_k(\boldsymbol{w}_{k,i-1}), \quad i\geq 0 \tag{5.76}$$

This comparison is *meaningful* only when all agents share the same minimizer, i.e., when

$$w_k^o = w^o, \quad k = 1, 2, \ldots, N \tag{5.77}$$

so that we can compare how well the individual agents are able to recover the same w^o as the centralized solution. For this reason, we need to re-introduce in this section only the requirement that all individual costs $\{J_k(w)\}$ are $\nu-$strongly convex with a uniform parameter ν. Since $J^{\text{glob}}(w)$ is the aggregate sum of the individual costs, then we can set the lower bound ν_c for the Hessian of $J^{\text{glob}}(w)$ in (5.20) at $\nu_c = N\nu$. From expressions (3.142) and (5.60) we then conclude that, for a sufficiently small μ, the convergence rates of the non-cooperative and centralized solutions will be similar to first-order in μ:

$$\begin{aligned} \alpha_{\text{cent}} &\overset{(5.60)}{\approx} 1 - 2\nu_c\left(\frac{\mu}{hN}\right) \\ &= 1 - 2\nu\left(\frac{\mu}{h}\right) \\ &\overset{(3.142)}{\approx} \alpha_{\text{ncop},k} \end{aligned} \tag{5.78}$$

where the symbol $\approx$ signifies (here and elsewhere) that we are ignoring higher-order terms in μ. Moreover, we observe from (4.170) that the average MSD level across N non-cooperative agents is given by

$$\begin{aligned} \text{MSD}_{\text{ncop,av}} &\triangleq \frac{1}{N}\sum_{k=1}^{N} \text{MSD}_{\text{ncop},k} \\ &= \frac{1}{N}\sum_{k=1}^{N} \frac{\mu}{2h}\text{Tr}\left(H_k^{-1}G_k\right) \\ &= \frac{\mu}{2hN}\,\text{Tr}\left(\sum_{k=1}^{N} H_k^{-1}G_k\right) \end{aligned} \tag{5.79}$$

so that comparing with (5.65), some simple algebra allows us to conclude the following statement.

Lemma 5.2 (Centralized MSD is superior to non-cooperative MSD)**.** Comparing the MSD performance levels (5.79) and (5.65) it holds that for sufficiently small step-sizes:

$$\text{MSD}_{\text{cent}} \;<\; \text{MSD}_{\text{ncop,av}} \tag{5.80}$$

Proof. First recall that $H_k > 0$ and $G_k \geq 0$ for each k; note that the individual $\{H_k\}$ are now positive-definite in view of the strong convexity assumption on the individual costs in this section. Let

$$G_k = L_k L_k^*, \quad k = 1, 2, \ldots, N \tag{5.81}$$

denote a square-root factorization for G_k where the L_k are full-rank matrices. Then, using the property $\text{Tr}(AB) = \text{Tr}(BA)$ for any matrices A and B of compatible dimensions, the MSD expressions can be re-written as (using H_c from (5.72)):

$$\text{MSD}_{\text{ncop,av}} \;=\; \frac{\mu}{2Nh}\,\text{Tr}\left[\sum_{k=1}^{N} L_k^* H_k^{-1} L_k\right] \tag{5.82}$$

$$\text{MSD}_{\text{cent}} \;=\; \frac{\mu}{2Nh}\,\text{Tr}\left[\sum_{k=1}^{N} L_k^* H_c^{-1}\, L_k\right] \tag{5.83}$$

so that

$$\text{MSD}_{\text{ncop,av}} - \text{MSD}_{\text{cent}} \;=\; \frac{\mu}{2Nh}\,\text{Tr}\left[\sum_{k=1}^{N} L_k^* (H_k^{-1} - H_c^{-1}) L_k\right] \tag{5.84}$$

The result follows by noting that $H_c^{-1} < H_k^{-1}$ for any k.

□

That is, while the centralized solution need not outperform every individual non-cooperative agent in general, its performance outperforms the average performance across all non-cooperative agents. The next example illustrates the above result by considering the scenario where all agents have the same Hessian matrices at $w = w^o$, namely,

$$H_k \equiv H, \quad k = 1, 2, \ldots, N \tag{5.85}$$

This situation occurs, for example, when the individual costs are identical across the agents, say, $J_k(w) \equiv J(w)$, as is common in machine

learning applications. This situation also occurs for mean-square-error costs of the form described by (5.5)–(5.6), when the regression covariance matrices, $\{R_{u,k}\}$, are uniform across all agents. In these cases when the Hessian matrices H_k are uniform, the example below establishes that the centralized solution actually improves over the average MSD performance of the non-cooperative solution by a factor of N [207].

Example 5.1 (N-fold improvement in performance)**.** Consider a collection of N agents whose individual cost functions, $J_k(w)$, are $\nu-$strongly convex and are minimized at the same location $w = w^o$. The costs are also assumed to have identical Hessian matrices at $w = w^o$, i.e., $H_k \equiv H$. Then, using (5.65), the MSD of the centralized implementation is given by

$$\mathrm{MSD}_{\mathrm{cent}} \;=\; \frac{1}{N}\left(\frac{\mu}{2Nh}\sum_{k=1}^{N}\mathrm{Tr}(H^{-1}G_k)\right) \overset{(5.79)}{=} \frac{1}{N}\,\mathrm{MSD}_{\mathrm{ncop,av}} \tag{5.86}$$

■

Example 5.2 (Multi-fold improvement in performance)**.** Assume in this example that all data are real-valued, and consider a situation in which the matrices $\{R_{s,k}\}$ are uniform across all agents so that $R_{s,k} \equiv R_s$, while $H_k = \alpha_k I_M > 0$ for some scalars $\{\alpha_k\}$. This situation arises, for instance, in the mean-square-error case (5.6) when $R_{u,k} = \sigma_{u,k}^2 I_M$ and the noise variances $\sigma_{v,k}^2$ across the agents are such that the product $\sigma_{v,k}^2\sigma_{u,k}^2 \equiv \sigma^2/4$ remains invariant over the agents. Then, in this case,

$$H_k \overset{(2.8)}{=} 2R_{u,k} = 2\sigma_{u,k}^2 I_M \equiv \alpha_k I_M \tag{5.87}$$

$$R_{s,k} \overset{(4.14)}{=} 4\sigma_{v,k}^2 R_{u,k} \;=\; 4\sigma_{v,k}^2\sigma_{u,k}^2 I_M = \sigma^2 I_M \equiv R_s \tag{5.88}$$

Let α_A and α_H denote the arithmetic and harmonic means of the scalars $\{\alpha_k\}$:

$$\alpha_A \;\triangleq\; \frac{1}{N}\sum_{k=1}^{N}\alpha_k, \quad \alpha_H \;\triangleq\; \left(\frac{1}{N}\sum_{k=1}^{N}\alpha_k^{-1}\right)^{-1} \tag{5.89}$$

Then, expressions (5.79) and (5.65) give

$$\mathrm{MSD}_{\mathrm{ncop,av}} \;=\; \mu\,\frac{1}{\alpha_H}\,M\sigma^2, \qquad \mathrm{MSD}_{\mathrm{cent}} \;=\; \frac{\mu}{N}\,\frac{1}{\alpha_A}\,M\sigma^2 \tag{5.90}$$

so that

$$\frac{\mathrm{MSD}_{\mathrm{cent}}}{\mathrm{MSD}_{\mathrm{ncop,av}}} \;=\; \frac{1}{N}\left(\frac{\alpha_H}{\alpha_A}\right) \tag{5.91}$$

in terms of the ratio of the harmonic mean to the arithmetic mean of the $\{\alpha_k\}$. Recall that the harmonic mean of a set of numbers is always smaller than or equal to the arithmetic mean of these numbers (and, moreover, its value tends to be close to the smaller numbers), it then holds that, for sufficiently small step-sizes:

$$\frac{\mathrm{MSD}_{\mathrm{cent}}}{\mathrm{MSD}_{\mathrm{ncop,av}}} \;\leq\; \frac{1}{N} \tag{5.92}$$

■

Example 5.3 (Centralized learner)**.** We revisit Example 4.5 and consider now a collection of N learners labeled $k = 1, 2, \ldots, N$. As before, each learner k receives a streaming sequence of real-valued vector samples $\{\boldsymbol{x}_{k,i}, i = 1, 2, \ldots\}$ arising from some fixed distribution $\mathcal{X}$. The goal is to determine the $M \times 1$ minimizer w^o of the $\nu-$strongly convex risk function $J(w)$ in (4.151). In Example 4.5 we examined the non-cooperative solution (4.152) where agents worked independently of each other to estimate w^o. In this example, we examine a centralized solution of the following stochastic-gradient form:

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \frac{\mu}{N}\sum_{k=1}^{N} \nabla_{w^{\mathsf{T}}} Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_{k,i}), \quad i \geq 0 \tag{5.93}$$

The gradient noise vector corresponding to each individual agent k is given by

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1}) \;=\; \nabla_{w^{\mathsf{T}}} Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_{k,i}) \;-\; \nabla_{w^{\mathsf{T}}} J(\boldsymbol{w}_{i-1}) \tag{5.94}$$

so that evaluating the expression for $\boldsymbol{s}_{k,i}(\boldsymbol{w})$ at $\boldsymbol{w} = w^o$, and using the fact that $\nabla_w J(w^o) = 0$, we get

$$\boldsymbol{s}_{k,i}(w^o) \;=\; \nabla_{w^{\mathsf{T}}} Q(w^o; \boldsymbol{x}_{k,i}) \tag{5.95}$$

Since we are assuming the distribution of the random process $\boldsymbol{x}_{k,i}$ is stationary and fixed across all agents, it follows that the covariance matrix of $\boldsymbol{s}_{k,i}(w^o)$ is constant across all agents:

$$R_{s,k} \;\triangleq\; \mathbb{E}\,\boldsymbol{s}_{k,i}(w^o)\boldsymbol{s}_{k,i}^{\mathsf{T}}(w^o) \;\equiv\; R_s, \qquad k = 1, 2, \ldots, N \tag{5.96}$$

Moreover, since all data are real-valued, it follows that the moment matrix G_k is $M \times M$ and given by

$$G_k \;=\; R_s, \quad k = 1, 2, \ldots, N \tag{5.97}$$

Substituting into (5.66), and using $h = 1$ for real data, we conclude that the excess-risk of the centralized solution (per unit agent) is given by

$$\mathrm{ER}_{\mathrm{cent}} \;=\; \frac{\mu}{4N^2}\,\mathrm{Tr}(NR_s) \;=\; \frac{\mu}{4N}\,\mathrm{Tr}(R_s) \tag{5.98}$$

which is $N-$fold superior to the performance of the non-cooperative agent given by (4.155) when $\mu_k \equiv \mu$. Similarly, using (5.65) we find that the MSD performance of the centralized solution is given by

$$\text{MSD}_{\text{cent}} \;=\; \frac{\mu}{2N}\,\text{Tr}(H^{-1}R_s) \tag{5.99}$$

■

Example 5.4 (Fully-connected networks)**.** In preparation for the discussion on networked agents, it is useful to describe one extreme situation where a collection of N agents are fully connected to each other — see Figure 5.3. In this case, each agent is able to access the data from all other agents and, therefore, each agent can run a centralized implementation of the same form as (5.22), namely,

$$\boldsymbol{w}_{k,i} \;=\; \boldsymbol{w}_{k,i-1} \;-\; \frac{\mu}{N}\sum_{\ell=1}^{N}\widehat{\nabla_{w^*}J_\ell}(\boldsymbol{w}_{k,i-1}), \quad i\geq 0 \tag{5.100}$$

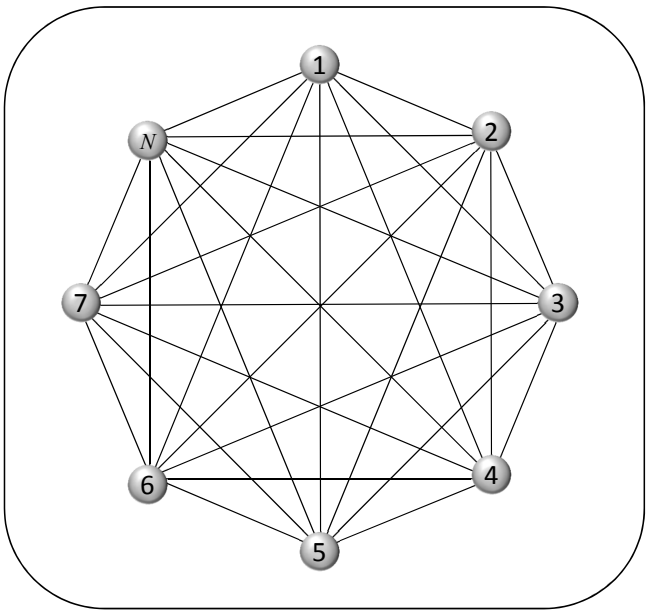

Figure 5.3: Example of a fully-connected network, where each agent can access information from all other agents.

When this happens, each agent will attain the same performance level as that of the centralized solution. Two observations are in place [207]. First, note from (5.100) that the information that agent k is receiving from all other agents is their gradient vector approximations. Obviously, other pieces of information could be shared among the agents, such as their iterates $\{\boldsymbol{w}_{\ell,i-1}\}$. Second, note that the right-most term multiplying μ in (5.100) corresponds to a convex combination of the approximate gradients from the various agents, with the combination coefficients being uniform and equal to $1/N$. In general, there is no need for these combination weights to be identical. Even more importantly, agents do not need to have access to information from all other agents in the network. We are going to see in the future chapters that interactions with a limited number of neighbors is sufficient for the agents to attain performance that is comparable to that of the centralized solution.

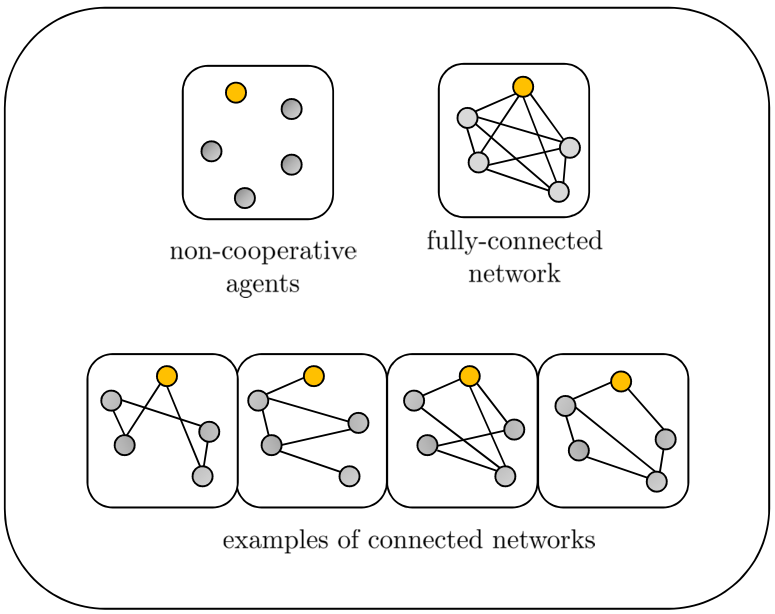

Figure 5.4: Examples of connected networks, with the left-most panel on the first row representing a collection of non-cooperative agents.

Figure 5.4 shows a sample selection of connected topologies for five agents. The panels in the first row correspond to the non-cooperative case (left) and the fully-connected case (right). The panels in the bottom row illustrate some

other topologies. In the coming chapters, we are going to present results that allow us to answer useful questions about such networked agents such as [207]: (a) which topology has best performance in terms of mean-square error and convergence rate? (b) Given any connected topology, can it be made to approach the performance of the centralized stochastic-gradient solution? (c) Which aspects of the topology influence performance? (d) Which aspects of the combination weights (policy) influence performance? (e) Can different topologies deliver similar performance levels? (f) Is cooperation always beneficial? (g) If the individual agents are able to solve the inference task individually in a stable manner, does it follow that the connected network will remain stable regardless of the topology and regardless of the cooperation strategy?

■

5.7 Decaying Step-Size Sequences

We finally examine the convergence and performance of the centralized solution (5.22) with a decaying step-size sequence, namely,

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \frac{\mu(i)}{N}\sum_{k=1}^{N}\widehat{\nabla_{w^*}J_k}(\boldsymbol{w}_{i-1}), \quad i\geq 0 \tag{5.101}$$

where $\mu(i) > 0$ satisfies either of the following two sets of conditions:

$$\sum_{i=0}^{\infty}\mu(i)=\infty, \qquad \lim_{i\to\infty}\mu(i)=0 \tag{5.102}$$

or

$$\sum_{i=0}^{\infty}\mu(i)=\infty, \qquad \sum_{i=0}^{\infty}\mu^2(i)<\infty \tag{5.103}$$

The following statement follows from the results of Lemmas 3.7 and 3.8 applied to the stochastic-gradient recursion (5.101).

Lemma 5.3 (Performance with decaying step-size)**.** Assume the aggregate cost (5.19) satisfies condition (5.20) for some parameters $0 < \nu_c \leq \delta_c$. Assume also that the individual gradient noise processes defined by (5.24) satisfy conditions (5.40)–(5.33). Then, the following convergence properties hold for (5.101):

(a) If the step-size sequence $\mu(i)$ satisfies (5.103), then $\boldsymbol{w}_i$ converges almost surely to w^o, written as $\boldsymbol{w}_i \to w^o$ a.s.

(b) If the step-size sequence $\mu(i)$ satisfies (5.102), then $\boldsymbol{w}_i$ converges in the mean-square-error sense to w^o, i.e., $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \to 0$.

(c) If the step-size sequence is selected as $\mu(i) = \tau_c/(i+1)$, where $\tau_c > 0$, then three convergence rates are possible. Specifically, for large enough i, it holds that:

$$\begin{cases} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;\leq\; \left(\frac{(\tau_c/N)^2\sigma_s^2}{(\nu_c/h)(\tau_c/N)-1}\right)\frac{1}{i} \;+\; o\left(\frac{1}{i}\right), & \nu_c\tau_c/hN > 1 \\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O\left(\frac{\log i}{i}\right), & \nu_c\tau_c/hN = 1 \\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; O\left(\frac{1}{i^{(\nu_c/h)(\tau_c/N)}}\right), & \nu_c\tau_c/hN < 1 \end{cases} \tag{5.104}$$

where $h = 2$ for complex data and $h = 1$ for real data. The fastest convergence rate occurs when $\nu_c\tau_c/hN > 1$ (i.e., for large enough τ_c) and is in the order of $O(1/i)$.

6

Multi-Agent Network Model

Moving forward, we shall study several distributed strategies for the solution of adaptation, learning, and optimization problems by networked agents. In preparation for these discussions, we describe in this chapter the network model and comment on some of its properties.

6.1 Connected Networks

We focus in our treatment on *connected* networks with N agents. In a connected network, there always exists at least one path connecting any two agents: the agents may be connected directly by an edge if they are neighbors, or they may be connected by a path that passes through other intermediate agents. The topology of a network can be described in terms of graphs, vertices, and edges (e.g., [256]).

Definition 6.1 (Graphs, vertices, and edges)**.** A network of size N is generally represented by a graph consisting of N vertices (which we will refer to more frequently as nodes or agents), and a set of edges connecting the vertices to each other. An edge that connects a vertex to itself is called a self-loop. Vertices connected by edges are called neighbors.

We assume the graph is undirected so that if agent k is a neighbor of agent ℓ, then agent ℓ is also a neighbor of agent k. Any two neighbors can share information both ways over the edge connecting them. This fact does not necessarily mean that the flow of information between the agents is symmetrical [208]. This is because we shall assign a pair of nonnegative weights, $\{a_{k\ell}, a_{\ell k}\}$, to the edge connecting agents k and ℓ. The scalar $a_{\ell k}$ will be used by agent k to scale data it receives from agent ℓ; this scaling can be interpreted as a measure of the confidence that agent k assigns to its interaction with agent ℓ. The subscripts ℓ and k in $a_{\ell k}$, with ℓ coming before k, designate agent ℓ as the source and agent k as the sink. Likewise, the scalar $a_{k\ell}$ will be used by agent ℓ to scale the data it receives from agent k. In this case, agent k is the source and agent ℓ is the sink. The weights $\{a_{k\ell}, a_{\ell k}\}$ can be different, and one or both weights can also be zero. We can therefore refer to the graph representing the network as a *weighted* graph with weights $\{a_{\ell k}, a_{k\ell}\}$ attached to the edges.

Figure 6.1 shows one example of a connected network. For emphasis in the figure, each edge between two neighboring agents is being represented (for now) by two directed arrows to indicate that information can flow both ways between the agents. The neighborhood of any agent k is denoted by $\mathcal{N}_k$ and it consists of all agents that are connected to k by edges; we assume by default that this set includes agent k regardless of whether agent k has a self-loop or not.

Definition 6.2 (Neighborhoods over weighted graphs)**.** The neighborhood of an agent k is denoted by $\mathcal{N}_k$ and it consists of all agents that are connected to k by an edge, in addition to agent k itself. Any two neighboring agents k and ℓ have the ability to share information over the edge connecting them. Whether this exchange of information occurs, and whether it is uni-directional, bi-directional, or non-existent, will depend on the values of the weighting scalars $\{a_{k\ell}, a_{\ell k}\}$ assigned to the edge.

When at least one a_{kk} is positive for some agent k, the connected network will be said to be *strongly-connected*. In other words, a strongly-connected network contains at least one self-loop, as is the case with

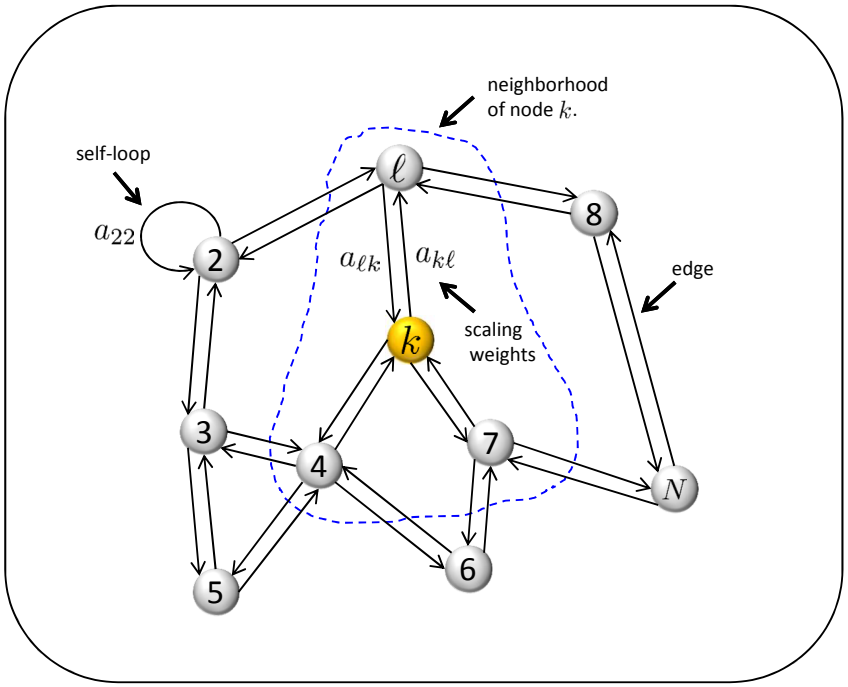

Figure 6.1: Agents that are linked by edges can share information. The neighborhood of agent k is marked by the broken line and consists of the set $\mathcal{N}_k = \{4, 7, \ell, k\}$. Likewise, the neighborhood of agent 2 consists of the set $\mathcal{N}_2 = \{2, 3, \ell\}$. For emphasis in the figure, we are representing edges between agents by two separate directed arrows with weights $\{a_{k\ell}, a_{\ell k}\}$. In future network representations, we will replace the two arrows by a single bi-directional edge.

agent 2 in Figure 6.1. More formally, we adopt the following terminology and define connected networks over weighted graphs as follows.

Definition 6.3 (Connected networks)**.** We distinguish between three types of connected networks; the third class of *strongly-connected* networks will be the focus of our study:

(a) Weakly-connected network: A network is said to be weakly connected if paths with nonzero scaling weights can be found linking any two distinct vertices in *at least one* direction either directly when they are neighbors or by

passing through intermediate vertices when they are not neighbors. In this way, it is possible for information to flow in *at least one* direction between any two distinct vertices in the network.

(b) Connected network: A network is said to be connected if paths with nonzero scaling weights can be found linking any two distinct vertices in *both* directions either directly when they are neighbors or by passing through intermediate vertices when they are not neighbors. In this way, information can flow in *both* directions between any two distinct vertices in the network, although the forward path from a vertex k to some other vertex ℓ need not be the same as the backward path from ℓ to k.

(c) Strongly-connected network: A network is said to be strongly-connected if it is a connected network with at least one self-loop with a positive scaling weight, meaning that $a_{kk} > 0$ for some vertex k. In this way, information can flow in both directions between any two distinct vertices in the network and, moreover, some vertices possess self-loops with positive weights.

Figure 6.2 illustrates these definitions by means of an example. The graph on the left represents a strongly-connected network: if we select any two agents k and ℓ, we can find paths linking them in both directions with positive weights on the edges along these paths. In the figure, we continue to represent edges between agents by two arrows. However, in order not to overwhelm the figure with combination weights, we are not showing arrows that correspond to zero weights on them; we are only showing arrows that correspond to positive weights. Thus, observe in the graph on the left that for agents 2 and 4, a valid path from 2 to 4 goes through agent 3 and one valid path for the reverse direction from 4 to 2 goes through agents 8 and 1. Similarly, paths can be determined linking all other combinations of agents in both directions.

Consider now the graph on the right in Figure 6.2. In this graph, we simply reversed the direction of the arrow that emanated from agent 1 towards agent 2 in the graph on the left (and which is represented in broken form for emphasis). Observe that now information cannot reach agent 2 from any of the other agents in the network, even though information from agent 2 can reach all other agents. At the same time, the information from agent 1 cannot reach any other agent in the network and agent 1 is only at the receiving end. This graph therefore

corresponds to a weakly-connected network. When some agents (like agent 2) are never able to receive information from other agents in the network, then these isolated agents will not be able to benefit from network interactions.

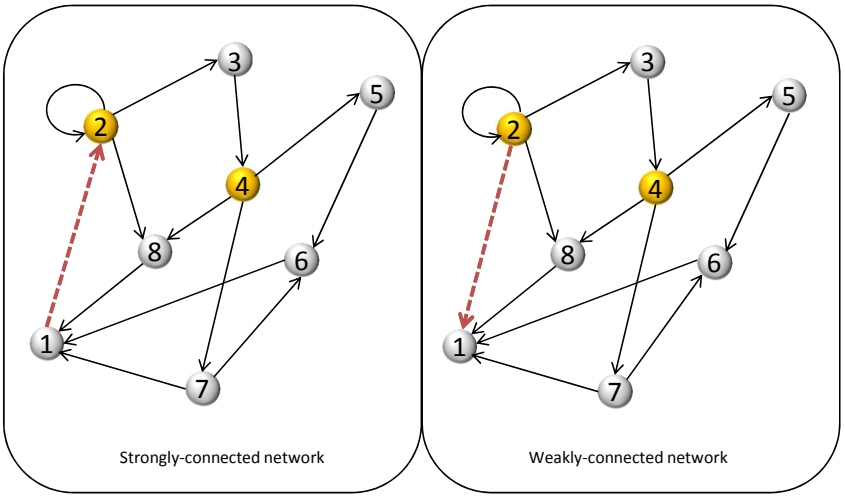

Figure 6.2: The graph on the left represents a strongly-connected network, while the graph on the right represents a weakly-connected network. The difference between both graphs is the reversal of the arrow connecting agents 1 and 2 (represented in broken form for emphasis). In the graph on the right, agent 2 is incapable of receiving (sensing) information from any of the other agents in the network, even though information from agent 2 can reach all other agents (directly or indirectly).

6.2 Strongly-Connected Networks

Observe that since we will be dealing with weighted graphs, we are therefore defining connected networks not in terms of whether paths can be found connecting their vertices but in terms of whether these paths allow for the *meaningful* exchange of information between the vertices. This fact is reflected by the requirement that all scaling weights must be *positive* over at least one of the paths connecting any two dis-

tinct vertices. This is a useful condition for the study of adaptation and learning over networks. As we are going to see in future chapters, agents will exchange information over the edges linking them. The information will be scaled by weights $\{a_{k\ell}, a_{\ell k}\}$. Therefore, for information to flow between agents, it is not sufficient for paths to exist linking these agents. It is also necessary that the information is not annihilated by zero scaling while it traverses the path. If information is never able to arrive at some particular agent, ℓ_o, because scaling is annihilating it before reaching ℓ_o then, for all practical (adaptation and learning) purposes, agent ℓ_o is disconnected from the other agents in the network even if information can still flow in the other direction from agent ℓ_o to the other agents. In this situation, agent ℓ_o will not benefit from cooperation with other agents in the network, while the other agents will benefit from information provided by agent ℓ_o. The assumption of a connected network therefore ensures that information will be flowing between any two arbitrary agents in the network and that this flow of information is bi-directional: information flows from k to ℓ and from ℓ to k, although the paths over which the flows occur need not be the same and the manner by which information is scaled over these paths can also be different.

The condition of a strongly-connected network implies that the network is connected and, additionally, there is at least one agent in the network that trusts its own information and will assign some positive weight to it. This is a reasonable condition and is characteristic of many real networks, especially biological networks. If $a_{kk} = 0$ for all k, then this means that all agents will be ignoring their individual information and will be relying instead on information received from other agents. The terminology of "strongly-connected networks" is perhaps somewhat excessive because it may unnecessarily convey the impression that the network needs to have more connectivity than is actually necessary.

The strong connectivity of a network translates into a useful property to be satisfied by the scaling weights $\{a_{\ell k}\}$; this property will be exploited to great effect in our analysis so we derive it here. Assume we collect the coefficients $\{a_{\ell k}\}$ into an $N \times N$ matrix $A = [a_{\ell k}]$, such

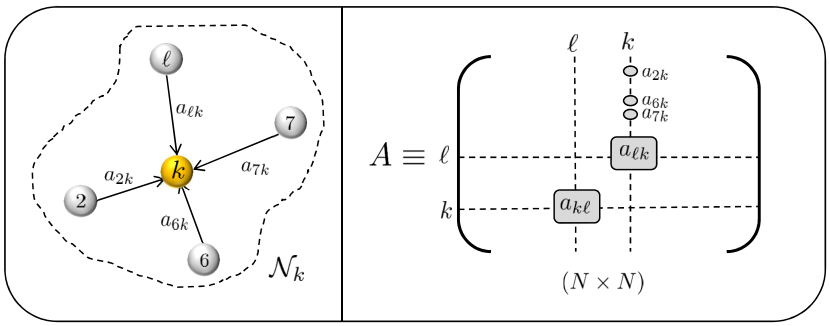

Figure 6.3: We associate an $N \times N$ combination matrix A with every network of N agents. The $(\ell, k)-$th entry of A contains the combination weight $a_{\ell k}$, which scales the data arriving at agent k and originating from agent ℓ.

that the entries on the $k-$th column of A contain the coefficients used by agent k to scale data arriving from its neighbors $\ell \in \mathcal{N}_k$; we set $a_{\ell k} = 0$ if $\ell \notin \mathcal{N}_k$ — see Figure 6.3. In this way, the row index in (ℓ, k) designates the source agent and the column index designates the sink agent (or destination). We refer to A as the *combination* matrix or combination policy. Even though the entries of A are non-negative (and several of them can be zero), it turns out that for combination matrices A that originate from strongly-connected networks, there exists an integer power of A such that all its entries are strictly positive, i.e., there exists some finite integer $n_o > 0$ such that

$$[A^{n_o}]_{\ell k} \; > \; 0 \tag{6.1}$$

for all $1 \leq \ell, k \leq N$. Combination matrices that satisfy this property are called *primitive* matrices.

Lemma 6.1 (Combination matrices of strongly-connected networks)**.** The combination matrix of a strongly-connected network is a primitive matrix.

Proof. Pick two arbitrary agents ℓ and k. Since the network is assumed to be connected, then this implies that there exists a sequence of agent indices $(\ell, m_1, m_2, \ldots, m_{n_{\ell k}-1}, k)$ of shortest length that forms a path from agent ℓ to agent k, say, with $n_{\ell k}$ nonzero scaling weights $\{a_{\ell m_1}, a_{m_1,m_2}, \ldots, a_{m_{n_{\ell k}-1},k}\}$:

$$\ell \overset{a_{\ell m_1}}{\longrightarrow} m_1 \overset{a_{m_1,m_2}}{\longrightarrow} m_2 \longrightarrow \ldots \longrightarrow m_{n_{\ell k}-1} \overset{a_{m_{n_{\ell k}-1},k}}{\longrightarrow} k \qquad [n_{\ell k} \text{ edges}] \tag{6.2}$$

From the rules of matrix multiplication, the $(\ell, k)-$th entry of the $n_{\ell k}-$th power of A is given by:

$$[A^{n_{\ell k}}]_{\ell k} = \sum_{m_1=1}^{N} \sum_{m_2=1}^{N} \ldots \sum_{m_{n_{\ell k}-1}=1}^{N} a_{\ell m_1} a_{m_1 m_2} \ldots a_{m_{n_{\ell k}-1} k} \tag{6.3}$$

We already know that the sum in (6.3) should be nonzero because of the existence of the aforementioned path linking agents ℓ and k with *nonzero* scaling weights. It follows that $[A^{n_{\ell k}}]_{\ell k} > 0$. This means that the matrix A is *irreducible*; a matrix A with nonnegative entries is said to irreducible if, and only if, for every pair of indices (ℓ, k), there exists a finite integer $n_{\ell k} > 0$ such that $[A^{n_{\ell k}}]_{\ell k} > 0$; which is what we have established so far. We assume that $n_{\ell k}$ is the smallest integer that satisfies this property. Note that under irreducibility, the power $n_{\ell k}$ is allowed to be dependent on the indices (ℓ, k). Therefore, network connectivity ensures the irreducibility of A. We now go a step further and show that *strong* network connectivity ensures the primitiveness of A. Recall from Definition 6.3 in the text that a strongly connected network is a connected network with the additional requirement that there exists at least one agent with a self-loop. We now verify that an irreducible matrix A with at least one positive diagonal element is necessarily primitive so that a common power n_o satisfies (6.1) for all (ℓ, k) (see, e.g., [168, p. 678] and [220]).

Since the network is strongly connected, this means that there exists at least one agent k_o with $a_{k_o,k_o} > 0$. We know from (6.3) that for any agent ℓ in the network, it holds that $[A^{n_{\ell k_o}}]_{\ell k_o} > 0$. Then,

$$\begin{aligned} \left[A^{(n_{\ell k_o}+1)}\right]_{\ell k_o} &= [A^{n_{\ell k_o}} A]_{\ell k_o} \\ &= \sum_{m=1}^{N} [A^{n_{\ell k_o}}]_{\ell m} \; a_{m k_o} \\ &\geq [A^{n_{\ell k_o}}]_{\ell k_o} \; a_{k_o,k_o} \\ &> 0 \end{aligned} \tag{6.4}$$

so that the positivity of the $(\ell, k_o)-$th entry is maintained at higher powers of A once it is satisfied at power $n_{\ell k_o}$. The integers $\{n_{\ell k_o}\}$ are bounded by N. Let

$$m_o \triangleq \max_{1 \leq \ell \leq N} \{n_{\ell k_o}\} \tag{6.5}$$

Then, the above result implies that

$$[A^{m_o}]_{\ell k_o} > 0, \quad \text{for all } \ell \tag{6.6}$$

so that the entries on the k_o-th column of A^{m_o} are all positive. Similarly, repeating the argument (6.3) we can verify that for arbitrary agents (k,ℓ), with the roles of k and ℓ now reversed, there exists a path of length $n_{k\ell}$ such that $[A^{n_{k\ell}}]_{k\ell} > 0$. For the same agent k_o with $a_{k_o,k_o} > 0$ as above, it holds that

$$\begin{aligned}
\left[A^{(n_{k_o\ell}+1)}\right]_{k_o\ell} &= [AA^{n_{k_o\ell}}]_{k_o\ell} \\
&= \sum_{m=1}^{N} a_{k_o m}\,[A^{n_{k_o\ell}}]_{m\ell} \\
&\geq a_{k_o,k_o}\,[A^{n_{k_o\ell}}]_{k_o\ell} \\
&> 0
\end{aligned} \tag{6.7}$$

so that the positivity of the $(k_o,\ell)-$th entry is maintained at higher powers of A once it is satisfied at power $n_{k_o\ell}$. Likewise, the integers $\{n_{k_o\ell}\}$ are bounded by N. Let

$$m_o' \triangleq \max_{1\leq\ell\leq N} \{n_{k_o\ell}\} \tag{6.8}$$

Then, the above result implies that

$$\left[A^{m_o'}\right]_{k_o\ell} > 0, \quad \text{for all } \ell \tag{6.9}$$

so that the entries on the k_o-th row of $A^{m_o'}$ are all positive.

Now, let $n_o = m_o + m_o'$ and let us examine the entries of the matrix A^{n_o}. We can write schematically

$$A^{n_o} = A^{m_o}A^{m_o'} = \begin{bmatrix} \times & \times & + & \times \\ \times & \times & + & \times \\ \times & \times & + & \times \\ \times & \times & + & \times \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ + & + & + & + \\ \times & \times & \times & \times \end{bmatrix} \tag{6.10}$$

where the plus signs are used to refer to the *positive* entries on the k_o-th column and row of A^{m_o} and $A^{m_o'}$, respectively, and the $\times$ signs are used to refer to the remaining entries of A^{m_o} and $A^{m_o'}$, which are necessarily nonnegative. It is clear from the above equality that the resulting entries of A^{n_o} will all be positive, and we conclude that A is primitive.

□

One important consequence of the primitiveness of A is that a famous result in matrix theory, known as the *Perron-Frobenius* Theorem [27, 113, 189] allows us to characterize the eigen-structure of A in the following manner — see Lemma F.4 in the appendix:

(a) The matrix A has a *single* eigenvalue at one.

(b) All other eigenvalues of A are strictly inside the unit circle (and, hence, have magnitude strictly less than one). Therefore, the spectral radius of A is equal to one, $\rho(A) = 1$.

(c) With proper sign scaling, all entries of the right-eigenvector of A corresponding to the single eigenvalue at one are *positive*. Let p denote this right-eigenvector, with its entries $\{p_k\}$ normalized to add up to one, i.e.,

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0, \quad k = 1, 2, \ldots, N \tag{6.11}$$

We refer to p as the *Perron eigenvector* of A. All other eigenvectors of A associated with the other eigenvalues will have at least one negative or complex entry.

6.3 Network Objective

In the remaining chapters of this treatment we are interested in showing how network cooperation can be exploited to solve a variety of problems in an advantageous manner. We are particularly interested in formulations that can solve adaptation, learning, and optimization problems in a decentralized and online manner in response to *streaming* data. It turns out that useful commonalities run across these three domain problems. For this reason, we shall keep the development general enough and then show, by means of examples, how the results can be used to handle many situations of interest as special cases.

Thus, consider a connected network consisting of a total of N agents, labeled $k = 1, 2, \ldots, N$. We associate with each agent a twice-differentiable individual *cost* function, denoted by $J_k(w) \in \mathbb{R}$. This function is sometimes called the *utility* function in applications

involving resource management issues and the *risk* function in machine learning applications; it may be called by other names in other domains. We adopt the generic terminology of a "cost" function. The function $J_k(w) \in \mathbb{R}$ is itself *real-valued*. However, for generality, its argument $w \in \mathbb{C}^M$ is assumed to be possibly *complex-valued*, say, of size $M \times 1$. This set-up is illustrated in Figure 6.4 where we are now representing the bi-directional edges between agents by single segment lines for ease of representation.

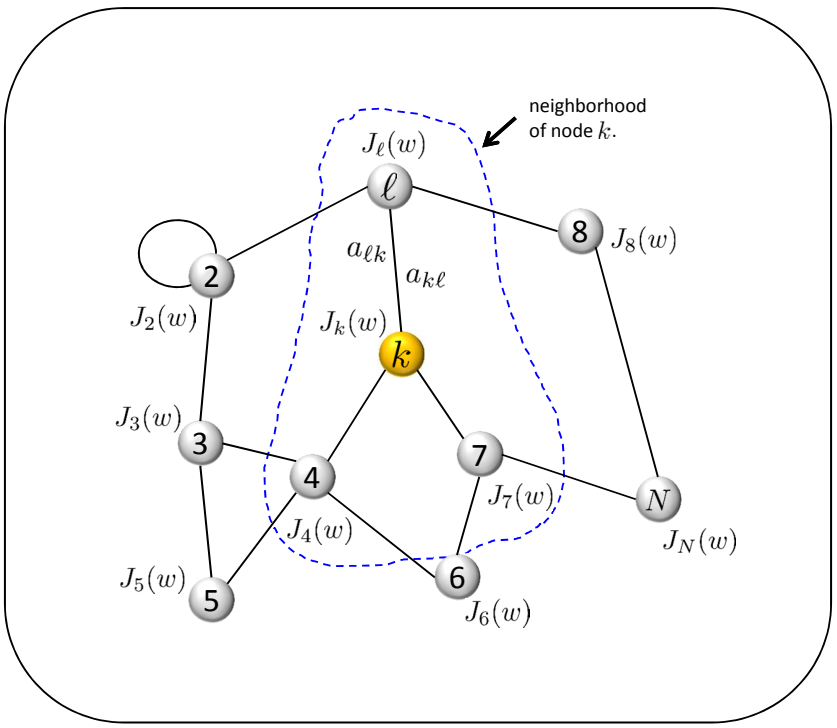

Figure 6.4: A cost function $J_k(w)$ is associated with each individual agent k in the network. The bi-directional edges between agents are being represented by single segment lines for ease of representation. Information can flow both ways over these edges with scalings $\{a_{k\ell}, a_{\ell k}\}$.

The objective of the network of agents is still to seek the unique minimizer of the aggregate cost function, $J^{\rm glob}(w)$, defined earlier by (5.19) and which we repeat below

$$J^{\rm glob}(w) \triangleq \sum_{k=1}^{N} J_k(w) \tag{6.12}$$

Now, however, we seek a *distributed* (as opposed to a centralized) solution. In a distributed implementation, each agent k can only rely on its own data and on data from its neighbors. We continue to assume that $J^{\rm glob}(w)$ satisfies the conditions of Assumption 5.1 with parameters $\{\nu_d, \delta_d, \kappa_d\}$, with the subscript "$d$" now used to indicate that these parameters are related to the distributed implementation.

Assumption 6.1 (Conditions on aggregate and individual costs)**.** It is assumed that the individual cost functions, $J_k(w)$, are each twice-differentiable and convex, with at least one of them being ν_d-strongly convex. Moreover, the aggregate cost function, $J^{\rm glob}(w)$, is also twice-differentiable and satisfies

$$0 < \frac{\nu_d}{h} I_{hM} \;\leq\; \nabla_w^2 J^{\rm glob}(w) \;\leq\; \frac{\delta_d}{h} I_{hM} \tag{6.13}$$

for some positive parameters $\nu_d \leq \delta_d$.

Under these conditions, the cost $J^{\rm glob}(w)$ will have a unique minimizer, which we continue to denote by w^o. Note that we are not requiring the individual costs $J_k(w)$ to be strongly convex. As mentioned earlier, it is sufficient to assume that at least one of these costs is ν_d-strongly convex while the remaining costs are simply convex; this condition ensures that $J^{\rm glob}(w)$ will be strongly convex.

The individual costs $\{J_k(w)\}$ can be distinct across the agents or they can all be identical, i.e., $J_k(w) \equiv J(w)$ for $k = 1, 2, \ldots, N$; in the latter situation, the problem of minimizing (6.12) would correspond to the case in which the agents work together to optimize the same cost function. Moreover, when they exist, the minimizers of the individual costs, $\{J_k(w)\}$, need not coincide with each other or with w^o; we shall write w_k^o to refer to a minimizer of $J_k(w)$. There are important

situations in practice where all minimizers $\{w_k^o\}$ happen to coincide with each other. For instance, examples abound where agents need to work cooperatively to attain a common objective such as tracking a target, locating a food source, or evading a predator (see, e.g., [56, 208, 214, 246]). This scenario is also common in machine learning problems [4, 37, 85, 192, 233, 239] when data samples at the various agents are generated by a common distribution parameterized by some vector, w^o. One such situation is illustrated in the next example.

Example 6.1 (Common minimizer)**.** Consider the same setting of Example 3.4 except that we now have N agents observing streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ that satisfy the regression model (3.119) with regression covariance matrices $R_{u,k} = \mathbb{E}\,\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i} > 0$ and with the same unknown w^o, i.e.,

$$\boldsymbol{d}_k(i) \;=\; \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i) \tag{6.14}$$

where the noise process, $\boldsymbol{v}_k(i)$, is independent of the regression data, $\boldsymbol{u}_{k,i}$. The individual mean-square-error costs are defined by

$$J_k(w) = \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2 \tag{6.15}$$

and are strongly convex in this case, with the minimizer of each $J_k(w)$ occurring at

$$w_k^o \;=\; R_{u,k}^{-1}r_{du,k}, \quad k = 1, 2, \ldots, N \tag{6.16}$$

If we multiply both sides of (6.14) by $\boldsymbol{u}_{k,i}^*$ from the left, and take expectations, we find that w^o satisfies

$$r_{du,k} = R_{u,k}w^o \tag{6.17}$$

This relation shows that the unknown w^o from (6.14) satisfies the same expression as w_k^o in (6.16), for any $k = 1, 2, \ldots, N$, so that we must have

$$w^o = w_k^o, \qquad k = 1, 2, \ldots, N \tag{6.18}$$

Therefore, this example amounts to a situation where all costs $\{J_k(w)\}$ attain their minima at the same location, w^o, even though the moments $\{r_{du,k}, R_{u,k}\}$ and, therefore, the individual costs $\{J_k(w)\}$, may be different from each other. This example highlights one convenience of working with mean-square-error (MSE) costs: under linear regression models of the form (6.14), the MSE formulation (6.15) allows each agent to recover w^o *exactly*.

■

One natural question that arises in the case of a common minimizer is to inquire why agents should cooperate to determine w^o when each one of them is capable of determining w^o on its own through (6.16)? There are at least two good reasons to justify cooperation even in this case. First, agents will rarely have access to the full information they need to determine w^o independently. For example, in many situations, agents may not know fully their own costs $J_k(w)$. For instance, agents may not know beforehand the statistical moments $\{r_{du,k}, R_{u,k}\}$ of the data that they are sensing; this is the situation we encountered earlier in Examples 3.1 and 3.4 when we developed recursive adaptation schemes to address this lack of information. When this occurs, agents would not be able to use (6.16) to determine w^o. Instead, they would need to replace the unavailable moments $\{r_{du,k}, R_{u,k}\}$ by some approximations before attempting (6.16). Moreover, different agents will generally be subject to different noise conditions and the quality of their moment approximations will therefore vary. In that case, their estimates for w^o will be as good as the quality of their data, as we already remarked earlier following result (5.10). Through cooperation with each other, not only agents with "bad" noise conditions will benefit, but also agents with "good" noise conditions can benefit and improve the accuracy of their estimation (see, e.g., Chapter 12 and also [208, 214]).

A second reason to motivate cooperation among the agents is that even when they know the moments $\{r_{du,k}, R_{u,k}\}$, the individual costs need not be strongly convex and the agents may not be able to recover w^o on their own due to ambiguities or ill-conditioning. For example, if some of the covariance matrices $\{R_{u,k}\}$ in Example 6.1 are singular, then the corresponding cost functions $\{J_k(w)\}$ will not be strongly convex and the individual agents will not be able to determine w^o uniquely. In that case, cooperation among agents would help them resolve the ambiguity about w^o.

Example 6.2 (Linear regression models)**.** Linear data models of the form (6.14) are common in practice. We provide two examples from [208]. Consider first a situation in which agents are spread over a geographical region and observe realizations of an auto-regressive (AR) random process $\{\boldsymbol{d}_k(i)\}$ of order M. The AR process observed by agent k satisfies the model:

$$\boldsymbol{d}_k(i) = \sum_{m=1}^{M} \beta_m \boldsymbol{d}_k(i-m) + \boldsymbol{v}_k(i), \quad k = 1, 2, \ldots, N \tag{6.19}$$

where i is the time index, the scalars $\{\beta_m\}$ represent the model parameters that the agents wish to identify, and $\boldsymbol{v}_k(i)$ represents the additive noise process. If we collect the $\{\beta_m\}$ into an $M \times 1$ column vector:

$$w^o \triangleq \operatorname{col}\{\beta_1, \beta_2, \ldots, \beta_M\} \tag{6.20}$$

and the past data into a $1 \times M$ regression vector:

$$\boldsymbol{u}_{k,i} \triangleq \begin{bmatrix} \boldsymbol{d}_k(i-1) & \boldsymbol{d}_k(i-2) & \ldots & \boldsymbol{d}_k(i-M) \end{bmatrix} \tag{6.21}$$

then we can rewrite the measurement equation (6.19) in the form (6.14) for each time instant i.

Consider a second example where the agents are now interested in estimating the taps of a communications channel or the parameters of some physical model of interest. Assume the agents are able to independently probe the unknown model and observe its response to excitations in the presence of additive noise. Each agent k probes the model with an input sequence $\{\boldsymbol{u}_k(i)\}$ and measures the response sequence, $\{\boldsymbol{d}_k(i)\}$, in the presence of additive noise. The system dynamics for each agent k is assumed to be described by a moving-average (MA) model of the form:

$$\boldsymbol{d}_k(i) = \sum_{m=0}^{M-1} \beta_m \boldsymbol{u}_k(i-m) + \boldsymbol{v}_k(i) \tag{6.22}$$

If we again collect the parameters $\{\beta_m\}$ into an $M \times 1$ column vector w^o, and the input data into a $1 \times M$ regression vector:

$$\boldsymbol{u}_{k,i} = \begin{bmatrix} \boldsymbol{u}_k(i) & \boldsymbol{u}_k(i-1) & \ldots & \boldsymbol{u}_k(i-M+1) \end{bmatrix} \tag{6.23}$$

then we arrive again at the same linear model (6.14).

■

Example 6.3 (Mean-square-error (MSE) networks). The data model introduced in Example 6.1 will be called upon frequently in our presentation to illustrate various concepts and results. We shall refer to strongly-connected networks with agents receiving data according to model (6.14) and seeking to estimate w^o by adopting the mean-square-error costs $J_k(w)$ defined by (6.15), as mean-square-error (MSE) networks.

We find it useful to collect in this example the details of the model for ease of reference whenever necessary. Thus, refer to Figure 6.5. The plot shows a

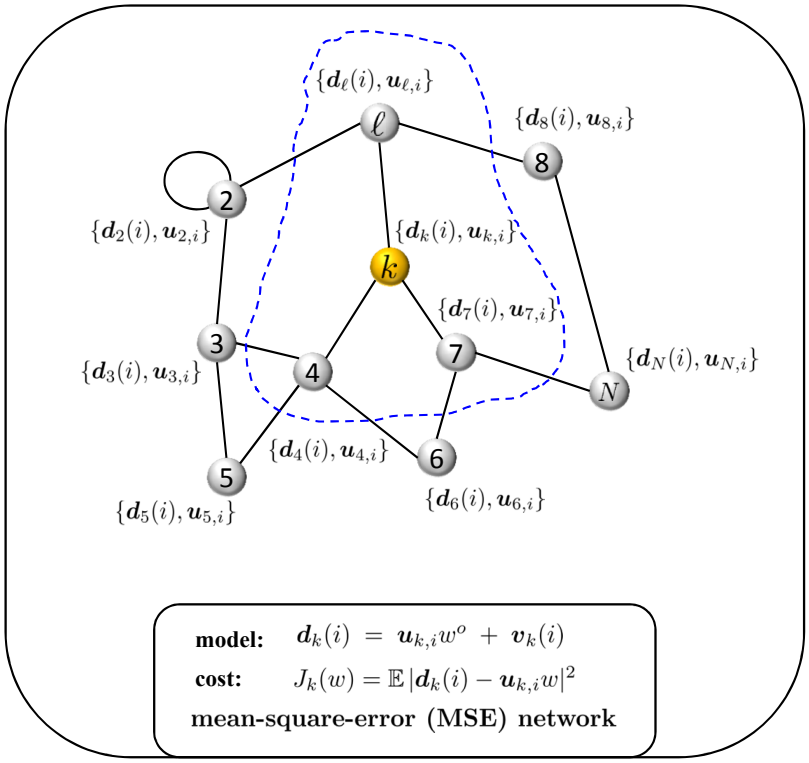

Figure 6.5: Illustration of mean-square-error (MSE) networks. The plot shows a strongly-connected network where each agent is subjected to streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ that satisfy the linear regression model (6.24). The cost associated with each agent is the mean-square-error cost defined by (6.25).

strongly-connected network where each agent is subjected to streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ that are assumed to satisfy the linear regression model:

$$\boldsymbol{d}_k(i) \;=\; \boldsymbol{u}_{k,i}w^o \;+\; \boldsymbol{v}_k(i), \;\; i \geq 0, \qquad k = 1, 2, \ldots, N \tag{6.24}$$

for some unknown $M \times 1$ vector w^o. A mean-square-error cost is associated with each agent k, namely,

$$J_k(w) \;=\; \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2, \qquad k = 1, 2, \ldots, N \tag{6.25}$$

The processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$ that appear in (6.24) are assumed to represent zero-mean jointly wide-sense stationary random processes that satisfy the following three conditions (these conditions help facilitate the

analysis of such networks):

(a) The regression data $\{\boldsymbol{u}_{k,i}\}$ are temporally white and independent over space with

$$\mathbb{E}\,\boldsymbol{u}_{k,i}^{*}\boldsymbol{u}_{\ell,j} \;\triangleq\; R_{u,k}\,\delta_{k,\ell}\,\delta_{i,j} \tag{6.26}$$

where $R_{u,k} > 0$ and the symbol $\delta_{m,n}$ denotes the Kronecker delta sequence: its value is equal to one when $m = n$ and its value is equal to zero otherwise.

(b) The noise process $\{\boldsymbol{v}_k(i)\}$ is temporally white and independent over space with variance

$$\mathbb{E}\,\boldsymbol{v}_k(i)\boldsymbol{v}_\ell^{*}(j) \;\triangleq\; \sigma_{v,k}^2\,\delta_{k,\ell}\,\delta_{i,j} \tag{6.27}$$

(c) The regression and noise processes $\{\boldsymbol{u}_{\ell,j}, \boldsymbol{v}_k(i)\}$ are independent of each other for all k, ℓ, i, j.

■

7

Multi-Agent Distributed Strategies

There are several distributed strategies that can be used to seek the minimizer of (6.12), namely,

$$w^o \triangleq \arg\min_{w} \sum_{k=1}^{N} J_k(w) \tag{7.1}$$

In this chapter, we describe three prominent strategies, namely,

(a) incremental strategies — see, e.g., [30, 31, 38, 55, 109, 129, 156, 161, 172, 193, 194, 209, 210];

(b) consensus strategies — see, e.g., [18, 26, 32, 46, 84, 87, 128, 137, 138, 174, 175, 185, 198, 204, 208, 214, 224, 241, 242, 265, 267];

(c) diffusion strategies — see, e.g., [62, 66, 69, 70, 86, 152, 163, 207, 208, 211, 214, 232, 238, 248, 276, 277].

While these algorithms can be motivated in alternative ways, some more formal than others, we opt to present them by using the centralized implementation (5.22) as a starting point, which we repeat below for ease of reference:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \frac{\mu}{N} \sum_{k=1}^{N} \widehat{\nabla_{w^*} J_k}(\boldsymbol{w}_{i-1}), \quad i \geq 0 \tag{7.2}$$

7.1 Incremental Strategy

We start with the incremental strategy. The centralized algorithm (7.2) is obviously not distributed since it requires that all information from the agents be aggregated at the fusion center to compute the sum of the gradient approximations. We can rewrite the algorithm in an equivalent manner that will motivate a particular distributed solution as follows.

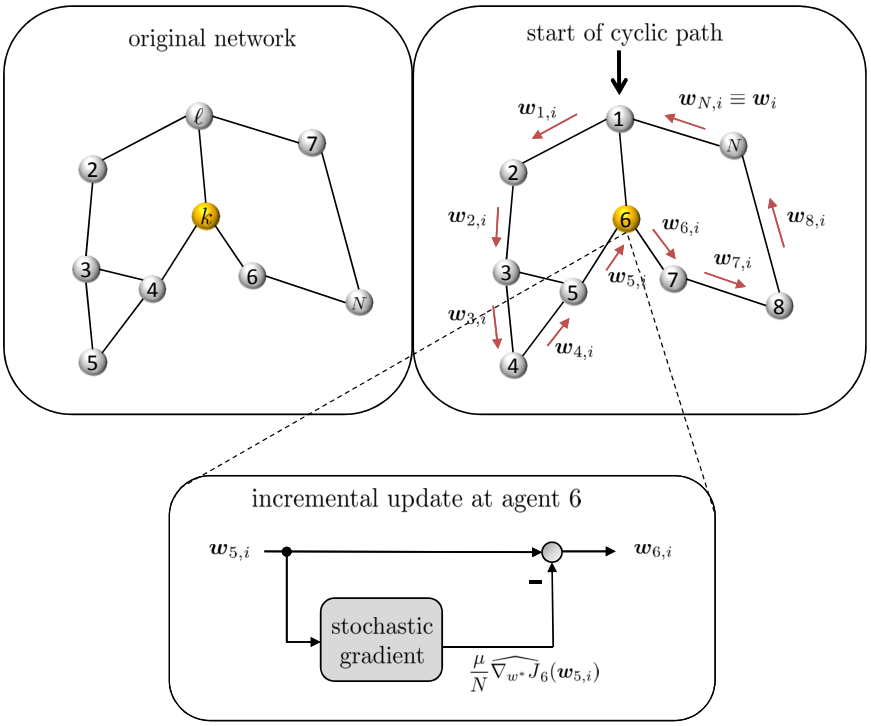

Figure 7.1: Starting from the given network on the left, a cyclic path is defined that visits all agents and is shown on the right. The agents are then re-numbered with agent 1 referring to the start of the cyclic path and agent N referring to its end. The diagram in the bottom illustrates the incremental calculations that are carried out by agent 6.

Referring to Figure 7.1, starting from a given network topology, we first determine a *cyclic* trajectory that covers all agents in the network in succession, one after the other. To facilitate the description of this construction, once a cycle has been selected, we re-number the agents along the trajectory from 1 to N with $\#1$ designating the agent at the start of the trajectory and $\#N$ designating the agent at the end of the trajectory. Then, at each iteration i, the centralized update (7.2) can be split into N consecutive *incremental* steps, with each step performed locally at one of the agents:

$$\left\{\begin{array}{rcrcl} \boldsymbol{w}_{1,i} & = & \boxed{\boldsymbol{w}_{i-1}} & - & \frac{\mu}{N}\widehat{\nabla_{w^*}J}_1(\boldsymbol{w}_{i-1}) \\ \boldsymbol{w}_{2,i} & = & \boldsymbol{w}_{1,i} & - & \frac{\mu}{N}\widehat{\nabla_{w^*}J}_2(\boldsymbol{w}_{i-1}) \\ \boldsymbol{w}_{3,i} & = & \boldsymbol{w}_{2,i} & - & \frac{\mu}{N}\widehat{\nabla_{w^*}J}_3(\boldsymbol{w}_{i-1}) \\ \vdots & = & \vdots & & \\ \boxed{\boldsymbol{w}_i} & = & \boldsymbol{w}_{N-1,i} & - & \frac{\mu}{N}\widehat{\nabla_{w^*}J}_N(\boldsymbol{w}_{i-1}) \end{array}\right. \tag{7.3}$$

In this implementation, information is passed from one agent to the next over the cyclic path until all agents are visited and the process is then repeated. Agent 1 starts with the existing iterate $\boldsymbol{w}_{i-1}$ and updates it to $\boldsymbol{w}_{1,i}$ using its approximation for its own gradient vector. Agent 2 then receives the updated iterate $\boldsymbol{w}_{1,i}$ from agent 1 and updates it to $\boldsymbol{w}_{2,i}$ using its approximate gradient vector, and so on. More generally, each agent k receives an intermediate variable, denoted by $\boldsymbol{w}_{k-1,i}$, from its predecessor agent $k-1$, incrementally adds one term from the gradient sum in (7.2) to this variable, and then computes its iterate, $\boldsymbol{w}_{k,i}$:

$$\boldsymbol{w}_{k,i} \;=\; \boldsymbol{w}_{k-1,i} \;-\; \frac{\mu}{N}\widehat{\nabla_{w^*}J}_k(\boldsymbol{w}_{i-1}) \tag{7.4}$$

At the end of the cycle of $N-$steps in (7.3), the iterate $\boldsymbol{w}_{N,i}$ at agent N coincides with the iterate $\boldsymbol{w}_i$ that would have resulted from (7.2).

Although recursion (7.3) is cooperative in nature, in that each agent is using some information from its preceding neighbor, this implementation still requires all agents to have access to one *global* piece of information represented by the vector $\boldsymbol{w}_{i-1}$. This is because this vector is used by all agents to evaluate the approximate gradient vectors in (7.3).

Consequently, implementation (7.3) is still *not* distributed. A fully distributed solution can only involve sharing of, and access to, information from local neighbors. At this point, we resort to a useful incremental construction, which has been widely studied in the literature (see, e.g., [30, 31, 38, 55, 109, 129, 156, 161, 172, 193, 194, 209, 210]). According to this construction, each agent k replaces the unavailable global variable $\boldsymbol{w}_{i-1}$ in (7.3) by the incremental variable it receives from its predecessor, which we denoted by $\boldsymbol{w}_{k-1,i}$. The approximate gradient vector is then evaluated at this intermediate variable, $\boldsymbol{w}_{k-1,i}$ rather than at the global variable $\boldsymbol{w}_{i-1}$, namely, equation (7.4) is replaced by

$$\boldsymbol{w}_{k,i} \;=\; \boldsymbol{w}_{k-1,i} \;-\; \frac{\mu}{N}\widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k-1,i}) \tag{7.5}$$

Obviously, the factor $1/N$ can be absorbed into the step-size μ. We leave it explicit to enable comparisons later with other distributed strategies. The resulting incremental implementation is summarized as follows.

Incremental strategy for adaptation and learning

for each time instant $i \geq 0$:
  set the fictitious boundary condition at $\boldsymbol{w}_{0,i} \leftarrow \boldsymbol{w}_{i-1}$.
  cycle over agents $k = 1, 2, \ldots, N$:
    agent k receives $\boldsymbol{w}_{k-1,i}$ from its preceding neighbor $k-1$.
    agent k performs: $\boldsymbol{w}_{k,i} \;=\; \boldsymbol{w}_{k-1,i} \;-\; \frac{\mu}{N}\widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k-1,i})$
  end
  $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_{N,i}$
end

(7.6)

Example 7.1 (Incremental LMS networks)**.** For the MSE network of Example 6.3, once a cyclic path has been determined and the agents renumbered from 1 to N, the incremental strategy (7.6) reduces to the following incremental LMS algorithm from [55, 156, 161, 209]:

$$\boldsymbol{w}_{k,i} \;=\; \boldsymbol{w}_{k-1,i} \;+\; \frac{2\mu}{Nh}\boldsymbol{u}_{k,i}^{*}[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k-1,i}] \tag{7.7}$$

where $h = 1$ for real data and $h = 2$ for complex data. It is understood that when the data are real-valued, the complex-conjugate transposition appearing on $\boldsymbol{u}_{k,i}^{*}$ should be replaced by the standard transposition, $\boldsymbol{u}_{k,i}^{\mathsf{T}}$.

■

The incremental solution (7.6) suffers from a number of limitations for applications involving adaptation and learning from streaming data. First, the incremental strategy is sensitive to agent or link failures. If an agent or link over the cyclic path fails, then the information flow over the network is interrupted. Second, starting from an arbitrary topology, determining a cyclic path that visits all agents is generally an NP-hard problem [139]. Third, cooperation between agents is limited with each agent allowed to receive data from one preceding agent and to share data with one successor agent. Fourth, for every iteration i, it is necessary to perform N incremental steps and to cycle through all agents in order to update $\boldsymbol{w}_{i-1}$ to $\boldsymbol{w}_i$; this means that the processing at the agents needs to be fast enough so that the N update steps can be completed before the next cycle begins. For these reasons, we shall not comment further on incremental strategies in this work. Readers can refer to more detailed studies that appear, for example, in [30, 31, 38, 55, 109, 129, 156, 161, 172, 193, 194, 209, 210].

We move on to motivate two other distributed strategies based on consensus and diffusion techniques that do not suffer from these limitations. These techniques take advantage of the following flexibility: (a) First, there is no reason why agents should only receive information from one neighbor at a time and pass information to only one other neighbor; (b) second, there is also no reason why the global variable $\boldsymbol{w}_{i-1}$ in (7.4) cannot be replaced by some other choice, other than $\boldsymbol{w}_{k-1,i}$, to attain decentralization; and (c) third, there is no reason why agents cannot adapt and learn simultaneously with other agents rather than wait for each cycle to complete.

7.2 Consensus Strategy

Examining description (7.6) for the incremental solution, we observe that the two objectives of cooperation and decentralization are attained by means of two artifacts. First, each agent k receives the incremental variable $\boldsymbol{w}_{k-1,i}$ from its predecessor and updates it to $\boldsymbol{w}_{k,i}$ using its own gradient vector approximation. This step, although limited, enforces one form of cooperation between two adjacent neighbors. Second, each

agent uses the iterate $\boldsymbol{w}_{k-1,i}$ received from its neighbor to replace the global variable $\boldsymbol{w}_{i-1}$ appearing in (7.4) by $\boldsymbol{w}_{k-1,i}$. This step allows the implementation to become decentralized with agents now relying solely on local data that are available to them. We highlight these two factors by rewriting the incremental step (7.4) at agent k as follows:

$$\boldsymbol{w}_{k,i} \;=\; \underbrace{\boldsymbol{w}_{k-1,i}}_{\text{(coop)}} \;-\; \frac{\mu}{N}\widehat{\nabla_{w^*} J}_k\big(\underbrace{\boldsymbol{w}_{k-1,i}}_{\text{(decen)}}\big) \tag{7.8}$$

where the term marked by the letters (coop) assists with *cooperation* and the term marked by the letters (decen) assists with *decentralization*. Both terms involve the same iterate $\boldsymbol{w}_{k-1,i}$, which appears twice on the right-hand side of the incremental update (7.8).

In the consensus strategy, the first $\boldsymbol{w}_{k-1,i}$ that agent k uses as the cooperation factor (coop) is replaced by a convex combination of the iterates that are available at the neighbors of agent k — see the first term on the right-hand side of (7.9). With regards to the second $\boldsymbol{w}_{k-1,i}$ on the right-hand side of (7.8), it is replaced by $\boldsymbol{w}_{k,i-1}$; this quantity is the iterate that is already available at agent k. In this manner, the consensus iteration at each agent k is given by:

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1} \;-\; \mu_k\widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \tag{7.9}$$

where we are further replacing the step-size μ/N in the incremental implementation by μ_k in the consensus implementation and allowing it to be agent-dependent for generality. This is because, as we are going to see, each agent will now be able to run its update simultaneously with the other agents. Moreover, it can be verified that by employing μ/N for incremental (and centralized solutions) and $\mu_k \equiv \mu$ for consensus, the convergence rates of these strategies will be similar (see future expression (11.141) in Example 11.2. Observe that the consensus update (7.9) can also be motivated by starting instead from the non-cooperative step (5.76) and replacing the first iterate $\boldsymbol{w}_{k,i-1}$ by the convex combination used in (7.9).

The combination coefficients $\{a_{\ell k}\}$ that appear in (7.9) are nonnegative scalars that are chosen to satisfy the following conditions for each

agent $k = 1, 2, \ldots, N$:

$$a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad \text{and} \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{7.10}$$

Condition (7.10) means that for every agent k, the sum of the weights $\{a_{\ell k}\}$ on the edges that arrive at it from its neighbors is one: the scalar $a_{\ell k}$ represents the weight that agent k assigns to the iterate $\boldsymbol{w}_{\ell,i-1}$ that it receives from agent ℓ. The coefficients $\{a_{\ell k}\}$ are free weighting parameters that are chosen by the designer; obviously, their selection will influence the performance of the algorithm (see Chapter 11). If we collect the entries $\{a_{\ell k}\}$ into an $N \times N$ matrix A, such that the $k-$th column of A consists of $\{a_{\ell k},\ \ell = 1, 2, \ldots, N\}$, then the second condition in (7.10) translates into saying that the entries on each *column* of A add up to one, i.e.,

$$A^{\mathsf{T}}\mathbb{1} = \mathbb{1} \tag{7.11}$$

We say that A is a *left-stochastic* matrix. One useful property of left-stochastic matrices is that the spectral radius of every such matrix is equal to one (so that the magnitude of any of the eigenvalues of A is bounded by one), i.e., $\rho(A) = 1$ (see [27, 104, 113, 189, 208] and Lemma F.4 in the appendix).

Now observe the following important fact from the consensus update (7.9). The information that is used by agent k from its neighbors are the iterates $\{\boldsymbol{w}_{\ell,i-1}\}$ and these iterates are *already* available for use from the previous iteration $i-1$. As such, there is *no* need any longer to cycle through the agents. At every iteration i, all agents in the network can run their consensus update (7.9) *simultaneously* by using iterates that are available from iteration $i-1$ at their neighbors to update their weight vectors. Accordingly, the consensus strategy (7.9) can be applied to a given network topology using its existing agent numbering (or labeling) scheme without the need to select a cycle and to re-number the agents, as was the case with the incremental strategy.

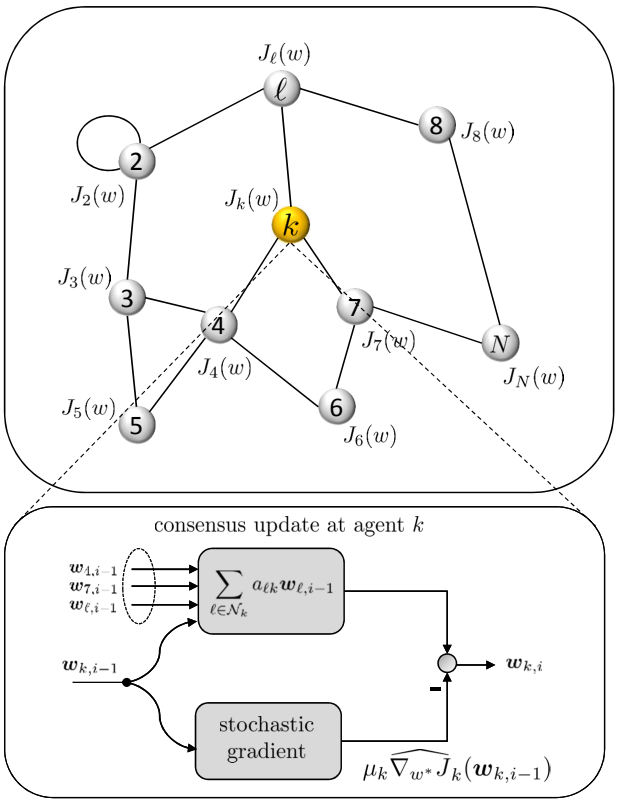

Figure 7.2: The diagram in the bottom shows the operations involved in the consensus implementation (7.9) at agent k, whose neighbors are agents are assumed to be $\{4, 7, \ell, k\}$.

Consensus strategy for adaptation and learning

for each time instant $i \geq 0$:

each agent $k = 1, 2, \ldots, N$ performs the update:

$$\begin{cases} \boldsymbol{\psi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu_k\, \widehat{\nabla_{w^*} J}_k\,(\boldsymbol{w}_{k,i-1}) \end{cases} \tag{7.12}$$

end

In the consensus implementation (7.9), at each iteration i, every agent k performs two steps: it aggregates the iterates from its neighbors and, subsequently, updates this aggregate value by the (negative of the conjugate) gradient vector evaluated at its existing iterate — see Figure 7.2. An equivalent representation that is useful for later analysis is to rewrite the consensus iteration (7.9) as shown in (7.12), where the intermediate iterate that results from the neighborhood combination is denoted by $\boldsymbol{\psi}_{k,i-1}$. Observe that the gradient vector in the consensus implementation (7.12) is evaluated at $\boldsymbol{w}_{k,i-1}$ and not $\boldsymbol{\psi}_{k,i-1}$.

Example 7.2 (Consensus LMS networks)**.** For the MSE network of Example 6.3, the consensus strategy (7.12) reduces to the following equivalent forms:

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1} \;+\; \frac{2\mu_k}{h}\boldsymbol{u}_{k,i}^{*}[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \tag{7.13}$$

or

$$\begin{cases} \boldsymbol{\psi}_{k,i-1} \;=\; \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} \;=\; \boldsymbol{\psi}_{k,i-1} + \dfrac{2\mu_k}{h}\boldsymbol{u}_{k,i}^{*}[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \end{cases} \tag{7.14}$$

where again $h = 1$ for real data and $h = 2$ for complex data. Moreover, when the data are real-valued, the complex-conjugate transposition appearing on $\boldsymbol{u}_{k,i}^{*}$ should be replaced by the standard transposition, $\boldsymbol{u}_{k,i}^{\mathsf{T}}$.

■

7.3 Diffusion Strategy

For ease of comparison, we repeat the incremental and consensus iterations (7.8) and (7.9) below:

$$\boldsymbol{w}_{k,i} \;=\; \underbrace{\boldsymbol{w}_{k-1,i}}_{\text{(coop)}} \;-\; \frac{\mu}{N}\widehat{\nabla_{w^*}J}_k\Big(\underbrace{\boldsymbol{w}_{k-1,i}}_{\text{(decen)}}\Big) \qquad \text{(incremental)} \tag{7.15}$$

$$\boldsymbol{w}_{k,i} \;=\; \underbrace{\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1}}_{\text{(coop)}} \;-\; \mu_k\widehat{\nabla_{w^*}J}_k\Big(\underbrace{\boldsymbol{w}_{k,i-1}}_{\text{(decen)}}\Big) \quad \text{(consensus)} \tag{7.16}$$

If we examine these updates, we observe that the cooperation and decentralization terms (coop) and (decen) in the incremental implementation (7.15) are identical to each other and equal to $\boldsymbol{w}_{k-1,i}$. On the other hand, the consensus construction (7.16) treats the factors "coop" and "decen" asymmetrically: the decentralization term (decen) is $\boldsymbol{w}_{k,i-1}$ while the cooperation term (coop) is different and involves a convex combination. This asymmetry is also clear from the equivalent form (7.12), where it is seen that the gradient vector in (7.12) is evaluated at $\boldsymbol{w}_{k,i-1}$ and not at the updated iterate $\boldsymbol{\psi}_{k,i-1}$. The *asymmetry* in the consensus update will be shown later in Sec. 10.6, and also in Examples 8.4 and 10.1, to be problematic when the strategy is used for adaptation and learning over networks. This is because the asymmetry can cause an unstable growth in the state of the network [248]. Diffusion strategies remove the asymmetry problem.

Combine-then-Adapt (CTA) Diffusion Strategy

There are several variations of the distributed diffusion strategy. The first diffusion variant can be motivated by requiring the *same* convex combination to be used for both the cooperation (coop) and decentralization (decen) factors. Doing so leads to the following algorithm known as the Combine-then-Adapt (CTA) diffusion strategy:

$$\boldsymbol{w}_{k,i} = \underbrace{\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1}}_{\text{(coop)}} - \mu_k \widehat{\nabla_{w^*} J}_k \underbrace{\left(\sum_{\ell\in\mathcal{N}_k} a_{\ell k} \boldsymbol{w}_{\ell,i-1} \right)}_{\text{(decen)}} \tag{7.17}$$

This implementation has *exactly the same* computational complexity as the consensus implementation (7.16). To see why, we rewrite (7.17) in a more revealing form in (7.18), where the convex combination term is first evaluated into an intermediate state variable, $\boldsymbol{\psi}_{k,i-1}$, and subsequently used to perform the gradient update — see Figure 7.3. Observe that in this form, and compared with (7.12), the gradient vector is now evaluated at $\boldsymbol{\psi}_{k,i-1}$.

Diffusion strategy for adaptation and learning (CTA)

for each time instant $i \geq 0$:

each agent $k = 1, 2, \ldots, N$ performs the update:

$$\begin{cases} \boldsymbol{\psi}_{k,i-1} \;=\; \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} \;=\; \boldsymbol{\psi}_{k,i-1} - \mu_k\, \widehat{\nabla_{w^*} J}_k\left(\boldsymbol{\psi}_{k,i-1}\right) \end{cases} \tag{7.18}$$

end

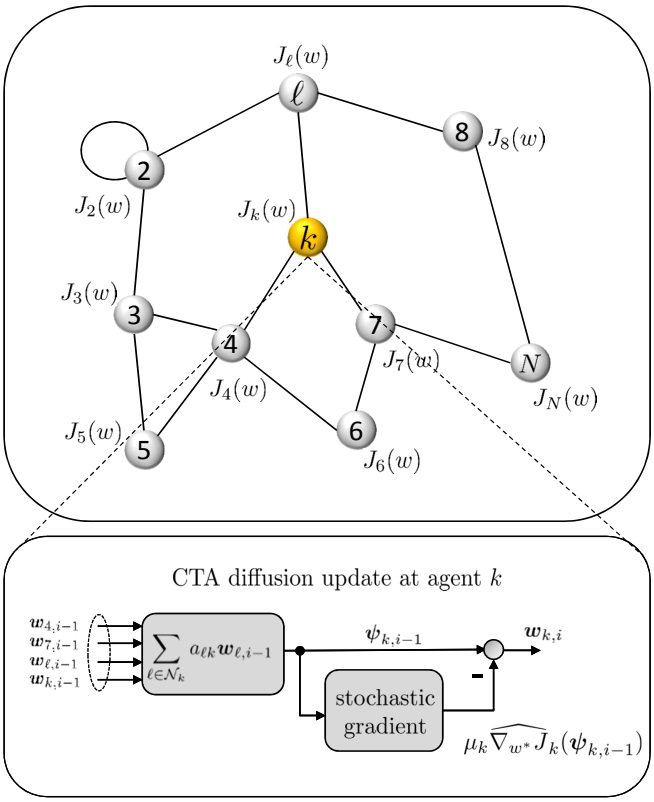

Figure 7.3: The diagram in the bottom shows the operations involved in the CTA diffusion implementation (7.18) at agent k, whose neighbors are agents $\{4, 7, \ell, k\}$.

At every iteration i, the strategy (7.18) performs two operations. The first operation is an aggregation step where agent k combines the existing iterates from its neighbors to obtain the intermediate iterate $\boldsymbol{\psi}_{k,i-1}$. All other agents in the network are *simultaneously* performing a similar step and aggregating the iterates of their neighbors. The second operation in (7.18) is an adaptation step where agent k approximates its gradient vector and uses it to update its intermediate iterate to $\boldsymbol{w}_{k,i}$. Again, all other agents in the network are *simultaneously* performing a similar information exchange step. The reason for the name "Combine-then-Adapt" (CTA) strategy is that the first step in (7.18) involves a combination step, while the second step involves an adaptation step. The reason for the qualification "diffusion" is that the use of the intermediate state $\boldsymbol{\psi}_{k,i-1}$ in both steps in (7.18) allows information to diffuse more thoroughly through the network. This is because information is not only being diffused through the aggregation of the neighborhood iterates, but also through the evaluation of the gradient vector at the aggregate state value.

Adapt-then-Combine (ATC) Diffusion Strategy

A similar implementation can be obtained by switching the order of the combination and adaptation steps in (7.18), as shown in the listing (7.19) — see Figure 7.4. The structure of the CTA and ATC strategies are fundamentally identical: the difference lies in which variable we choose to correspond to the updated iterate $\boldsymbol{w}_{k,i}$. In ATC, we choose the result of the *combination* step to be $\boldsymbol{w}_{k,i}$, whereas in CTA we choose the result of the *adaptation* step to be $\boldsymbol{w}_{k,i}$.

Diffusion strategy for adaptation and learning (ATC)

for each time instant $i \geq 0$:

each agent $k = 1, 2, \ldots, N$ performs the update:

$$\left\{\begin{array}{rcl} \boldsymbol{\psi}_{k,i} & = & \boldsymbol{w}_{k,i-1} - \mu_k \, \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} & = & \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \, \boldsymbol{\psi}_{\ell,i} \end{array}\right. \tag{7.19}$$

end

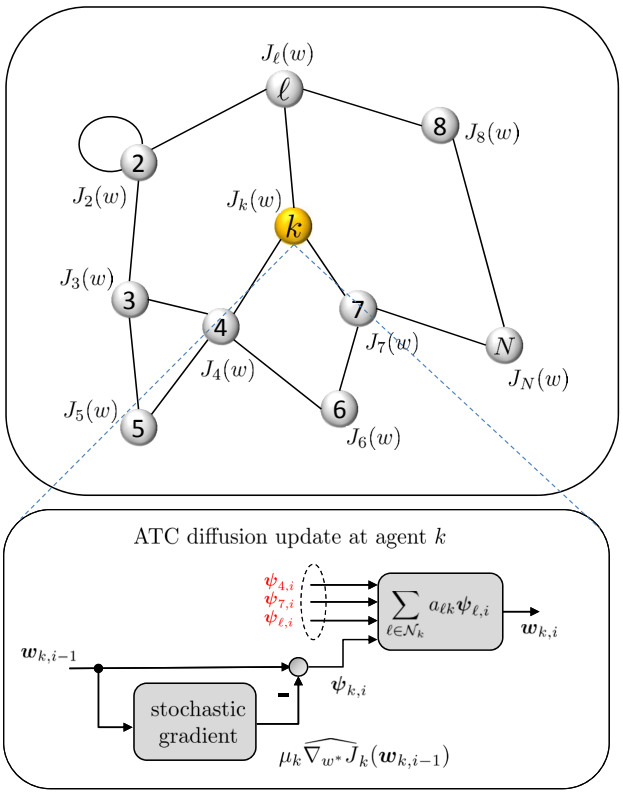

Figure 7.4: The diagram on the right shows the operations involved in the ATC diffusion implementation (7.19) at agent k, whose neighbors are agents $\{4, 7, \ell, k\}$.

In the ATC implementation, the first operation is the *adaptation* step where agent k uses its approximate gradient vector to update $\boldsymbol{w}_{k,i-1}$ to the intermediate state $\boldsymbol{\psi}_{k,i}$. All other agents in the network are performing a similar step simultaneously and updating their existing iterates $\{\boldsymbol{w}_{\ell,i-1}\}$ into intermediate iterates $\{\boldsymbol{\psi}_{\ell,i}\}$ by using information from their neighbors. The second step in (7.19) is an *aggregation* or consultation step where agent k combines the intermediate iterates from its neighbors to obtain its updated iterate $\boldsymbol{w}_{k,i}$. Again, all other agents in the network are simultaneously performing a similar step. The reason for the name "Adapt-then-Combine" (ATC) strategy is

that the first step (7.19) is an adaptation step, while the second step is a combination step. Again, this implementation has exactly the same computational complexity as the consensus implementation (7.16). If desired, both steps in (7.19) can be combined into a single update as:

$$\boldsymbol{w}_{k,i} \;=\; \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\left(\boldsymbol{w}_{\ell,i-1} \;-\; \mu_\ell\widehat{\nabla_{w^*}J}_\ell(\boldsymbol{w}_{\ell,i-1})\right) \tag{7.20}$$

or, equivalently,

$$\boldsymbol{w}_{k,i} \;=\; \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{w}_{\ell,i-1} \;-\; \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\mu_\ell\widehat{\nabla_{w^*}J}_\ell(\boldsymbol{w}_{\ell,i-1}) \tag{7.21}$$

where it is seen that the gradient vectors of the neighbors are also being combined by the ATC update, with each gradient evaluated at the respective iterate $\boldsymbol{w}_{\ell,i-1}$.

Example 7.3 (Diffusion LMS networks)**.** For the MSE network of Example 6.3, the CTA and ATC diffusion strategies (7.18) and (7.19) reduce to the following updates:

$$\left\{\begin{array}{rcl}\boldsymbol{\psi}_{k,i-1} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\;\boldsymbol{w}_{\ell,i-1}\\ \boldsymbol{w}_{k,i} &=& \boldsymbol{\psi}_{k,i-1}+\dfrac{2\mu_k}{h}\boldsymbol{u}_{k,i}^*\left[\boldsymbol{d}_k(i)-\boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k,i-1}\right]\end{array}\right. \qquad \text{(CTA)} \tag{7.22}$$

and

$$\left\{\begin{array}{rcl}\boldsymbol{\psi}_{k,i} &=& \boldsymbol{w}_{k,i-1}+\dfrac{2\mu_k}{h}\boldsymbol{u}_{k,i}^*\left[\boldsymbol{d}_k(i)-\boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}\right]\\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\;\boldsymbol{\psi}_{\ell,i}\end{array}\right. \qquad \text{(ATC)} \tag{7.23}$$

where for real data and $h = 2$ for complex data. Again, when the data are real-valued, the complex-conjugate transposition appearing on $\boldsymbol{u}_{k,i}^*$ should be replaced by the standard transposition, $\boldsymbol{u}_{k,i}^{\mathsf{T}}$.

■

Example 7.4 (Diffusion logistic network)**.** We reconsider the pattern classification problem from Example 3.2 where we now allow N agents to cooperate with each other over a connected network topology to solve the logistic regression problem — see Figure 7.5.

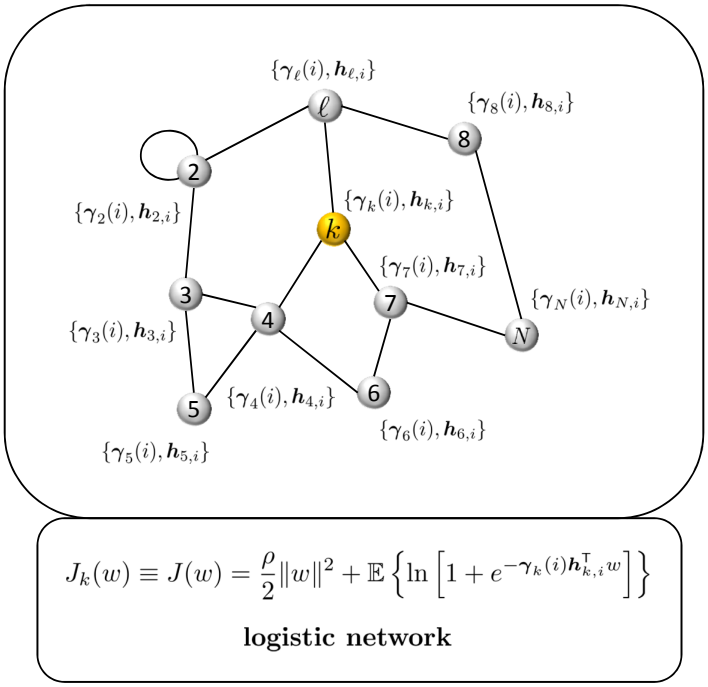

Figure 7.5: Each agent k receives streaming data $\{\boldsymbol{\gamma}_k(i), \boldsymbol{h}_{k,i}\}$. The agents cooperate to minimize the regularized logistic cost (7.24).

Each agent k is assumed to receive streaming data $\{\boldsymbol{\gamma}_k(i), \boldsymbol{h}_{k,i}\}$ at time i. The variable $\boldsymbol{\gamma}_k(i)$ assumes the values ± 1 and designates the class that feature vector $\boldsymbol{h}_{k,i}$ belongs to. The objective is to use the training data to determine the vector w^o that minimizes the regularized logistic cost under the assumption of joint wide-sense stationarity over the random data:

$$J(w) \;\triangleq\; \frac{\rho}{2}\|w\|^2 \;+\; \mathbb{E}\left\{\ln\left(1 + e^{-\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}} w}\right)\right\} \tag{7.24}$$

where $J(w)$ is the same for all agents. The corresponding loss function is

$$Q(w; \boldsymbol{\gamma}_k(i), \boldsymbol{h}_{k,i}) \;\triangleq\; \frac{\rho}{2}\|w\|^2 \;+\; \ln\left(1 + e^{-\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}} w}\right) \tag{7.25}$$

By using the gradient vector of $Q(\cdot)$ relative to w^{T} to approximate $\nabla_{w^{\mathsf{T}}} J(w)$, we arrive at the following ATC diffusion implementation of a distributed strat-

egy for solving the logistic regression problem cooperatively:

$$\begin{cases} \boldsymbol{\psi}_{k,i} &= (1-\rho\mu_k)\boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}\left(\dfrac{1}{1+e^{\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}}\boldsymbol{w}_{k,i-1}}}\right) \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{cases} \tag{7.26}$$

■

Diffusion Strategies with Enlarged Cooperation

Other forms of diffusion strategies are possible by allowing for enlarged cooperation and exchange of information among the agents, such as exchanging gradient vector approximations *in addition* to the iterates. For example, the following two forms of CTA and ATC employ an additional set of combination coefficients $\{c_{\ell k}\}$ to aggregate gradient information [62, 66, 208]:

$$\begin{cases} \boldsymbol{\psi}_{k,i-1} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} &= \boldsymbol{\psi}_{k,i-1} - \mu_k \displaystyle\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\widehat{\nabla_{w^*} J_\ell}(\boldsymbol{\psi}_{k,i-1}) \end{cases} \quad \text{(CTA)} \tag{7.27}$$

and

$$\begin{cases} \boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{k,i-1} - \mu_k \displaystyle\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\widehat{\nabla_{w^*} J_\ell}(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{cases} \quad \text{(ATC)} \tag{7.28}$$

where the $\{c_{\ell k}\}$ are nonnegative scalars that satisfy the following conditions for all agents $k = 1, 2, \ldots, N$:

$$c_{\ell k} \geq 0, \quad \sum_{k=1}^{N} c_{\ell k} = 1, \quad \text{and} \quad c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{7.29}$$

The coefficients $\{c_{\ell k}\}$ are free parameters that are chosen by the designer. If we collect the entries $\{c_{\ell k}\}$ into an $N \times N$ matrix C, so that the $\ell-$th row of C is formed of $\{c_{\ell k},\ k = 1, 2, \ldots, N\}$, then the second condition in (7.29) corresponds to the requirement that the entries on each *row* of C should add up to one, i.e.,

$$C\mathbb{1} = \mathbb{1} \tag{7.30}$$

We say that C is a *right-stochastic* matrix. Observe that the above enlarged diffusion strategies are equivalent to associating with each agent k the weighted neighborhood cost function:

$$J'_k(w) \triangleq \sum_{\ell \in \mathcal{N}_k} c_{\ell k} J_\ell(w) \tag{7.31}$$

and then applying (7.18) or (7.19). Our discussion in the sequel focuses on the case $C = I_N$. Additional details on the case $C \neq I_N$ appear in [62, 66, 208].

Discussion and Related Literature

As remarked in [207, 208], there has been extensive work on consensus techniques in the literature, starting with the foundational results by [26, 84], which were of a different nature and did not respond to streaming data arriving continuously at the agents, as is the case, for instance, with the continuous arrival of data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ in Examples 7.2–7.4. The original consensus formulation deals instead with the problem of computing averages over graphs. This can be explained as follows [26, 84, 241, 242]. Consider a collection of (scalar or vector) measurements denoted by $\{w_\ell,\ \ell = 1, 2, \ldots, N\}$ available at the vertices of a connected graph with N agents. The objective is to devise a distributed algorithm that enables every agent to determine the *average* value:

$$\overline{w} \triangleq \frac{1}{N} \sum_{k=1}^{N} w_k \tag{7.32}$$

by interacting solely with its neighbors. When this occurs, we say that the agents have reached consensus (or agreement) about $\overline{w}$. We select an $N \times N$ *doubly-stochastic* combination matrix $A = [a_{\ell k}]$; a doubly-stochastic matrix is one that has nonnegative elements and satisfies

$$A^{\mathsf{T}} \mathbb{1} = \mathbb{1}, \qquad A\mathbb{1} = \mathbb{1} \tag{7.33}$$

We assume the second largest-magnitude eigenvalue of A satisfies

$$|\lambda_2(A)| < 1 \tag{7.34}$$

Using the combination coefficients $\{a_{\ell k}\}$, each agent k then iterates *repeatedly* on the data of its neighbors:

$$w_{k,i} \;=\; \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1}, \quad i\geq 0, \quad k=1,2,\ldots,N \tag{7.35}$$

starting from the boundary conditions $w_{\ell,-1} = w_\ell$ for all $\ell \in \mathcal{N}_k$. The superscript i continues to denote the iteration index. Every agent k in the network performs the same calculation, which amounts to combining repeatedly, and in a convex manner, the state values of its neighbors. It can then be shown that (see [26, 84] and [208, App.E]):

$$\lim_{i\to\infty}\; w_{k,i} \;=\; \overline{w}, \qquad k=1,2,\ldots,N \tag{7.36}$$

In this way, through the localized iterative process (7.35), the agents are able to converge to the global average value, $\overline{w}$.

Motivated by this elegant result, several works in the literature (e.g., [8, 32, 52, 83, 128, 137, 138, 142, 174, 175, 179, 224, 242, 265]) proposed useful extensions of the original consensus construction (7.35) to minimize aggregate costs of the form (5.19) or to solve distributed estimation problems of the least-squares or Kalman filtering type. Some of the earlier extensions involved the use of *two* separate time-scales: one faster time-scale for performing multiple consensus iterations similar to (7.35) over the states of the neighbors, and a second slower time-scale for performing gradient vector updates or for updating the estimators by using the result of the consensus iterations (e.g., [52, 83, 128, 138, 142, 179, 265]). An example of a two-time scale implementation would be an algorithm of the following form:

$$\left\{\begin{array}{l}
\boldsymbol{w}_{\ell,i-1}^{(-1)} \;\longleftarrow\; \boldsymbol{w}_{\ell,i-1}, \;\; \text{for all agents } \ell \text{ at iteration } i-1 \\
\textbf{for } n=0,1,2,\ldots,J-1 \text{ iterate:} \\
\quad \boldsymbol{w}_{k,i-1}^{(n)} \;=\; \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{w}_{\ell,i-1}^{(n-1)}, \;\; \text{for all } k=1,2,\ldots,N \\
\textbf{end} \\
\boldsymbol{w}_{k,i} \;=\; \boldsymbol{w}_{k,i-1}^{(J-1)} \;-\; \mu_k\,\widehat{\nabla_{w^*}J}_k(\boldsymbol{w}_{k,i-1})
\end{array}\right. \tag{7.37}$$

If we compare the last equation in (7.37) with (7.9), we observe that the variable $\boldsymbol{w}_{k,i-1}^{(J-1)}$ that is used in (7.37) to obtain $\boldsymbol{w}_{k,i}$ is the result of J

repeated applications of a consensus operation of the form (7.35) on the iterates $\{\boldsymbol{w}_{\ell,i-1}\}$. The purpose of these repeated calculations is to approximate well the average of the iterates in the neighborhood of agent k. These J repeated averaging operations need to be completed before the availability of the gradient information for the last update step in (7.37). In other words, the J averaging operations need to performed at a faster rate than the last step in (7.37). Such two time-scale implementations are a hindrance for real-time adaptation from streaming data. The separate time-scales turn out to be unnecessary and this fact was one of the motivations for the introduction of the single time-scale diffusion strategies in [57, 58, 60, 61, 159, 160, 162, 163, 211].

Building upon a useful procedure for distributed optimization from [242, Eq. (2.1)] and [32, Eq. (7.1)], more recent works proposed single time-scale implementations for consensus strategies as well by using an implementation similar to (7.9) — see, e.g., [46, Eq. (3)], [174, Eq. (3)], [87, Eq. (19)], and [137, Eq.(9)]. These references, however, generally employ decaying step-sizes, $\mu_k(i) \to 0$, to ensure that the iterates $\{\boldsymbol{w}_{k,i}\}$ across all agents will converge almost-surely to the same value (thus, reaching agreement or consensus), namely, they employ recursions of the form:

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \;-\; \mu_k(i) \widehat{\nabla_{w^*} J_k}(\boldsymbol{w}_{k,i-1}) \tag{7.38}$$

or variations thereof, such as replacing $\mu_k(i)$ by some time-variant gain matrix sequence, say, $K_{k,i}$:

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \;-\; K_{k,i} \cdot \widehat{\nabla_{w^*} J_k}(\boldsymbol{w}_{k,i-1}) \tag{7.39}$$

As noted before, when diminishing step-sizes are used, adaptation is turned off over time, which is prejudicial for learning purposes. For this reason, we are instead setting the step-sizes to constant values in (7.9) in order to endow the consensus iteration with continuous adaptation and learning abilities (and to enhance the convergence rate). It turns out that some care is needed for consensus implementations when constant step-sizes are used. The main reason is that, as explained later in Sec. 10.6 and also Examples 8.4 and 10.1, and as alluded to

earlier, instability can occur in consensus networks due to an inherent asymmetry in the dynamics of the consensus iteration.

A second main reason for the introduction of cooperative strategies of the diffusion type (7.22) and (7.23) has been to show that single time-scale distributed learning from *streaming* data is possible, and that this objective can be achieved under *constant* step-size adaptation in a stable manner [60, 62, 69, 70, 159, 160, 162, 163, 211, 277] — see also Chapters 9–11 further ahead; the diffusion strategies further allow A to be left-stochastic and permit larger modes of cooperation than doubly-stochastic policies. The CTA diffusion strategy (7.22) was first introduced for mean-square-error estimation problems in [159, 160, 163, 211]. The ATC diffusion structure (7.23), with adaptation preceding combination, appeared in the work [57] on adaptive distributed least-squares schemes and also in the works [58, 60–62] on distributed mean-square-error and state-space estimation methods. The CTA structure (7.18) with an iteration dependent step-size that decays to zero, $\mu(i) \rightarrow 0$, was employed in [153, 196, 226] to solve distributed optimization problems that require all agents to reach agreement. The ATC form (7.23), also with an iteration dependent sequence $\mu(i)$ that decays to zero, was employed in [34, 227] to ensure almost-sure convergence and agreement among agents.

There has also been works on applying instead the alternating direction method of multipliers (ADMM) [44] to the design of consensus-type algorithms in [165, 216]. To enforce agreement among the agents, these last two references impose the requirement that the iterates at the agents should match each other. By doing so, the authors arrive at an implementation that necessitates the fine tuning of several parameters and whose performance is sensitive to the values of these parameters. Specifically, reference [216] considers networks where agents sense real-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ that are related via the regression model $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i)$. The individual cost associated with each agent is again the mean-square-error cost, $J_k(w) = \mathbb{E}\,(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w)^2$. The network model used in [216] is not homogeneous and assumes a special structure. The network is assumed to consist of two types of nodes. One type involves "regular" agents, indexed by k, where data

samples $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ arrive sequentially. The second type of nodes involves "bridge" agents, indexed by b, which do not receive data and their purpose is to connect the regular agents. The set of bridge nodes is denoted by $\mathcal{B}$. The two classes of nodes are required to be placed in a particular manner in the network, namely, (i) for every regular agent k, there should exist at least one bridge node $b \in \mathcal{B}$ such that $b \in \mathcal{N}_k$, and (ii) for every two bridge nodes, b_1 and b_2, there should exist a path connecting them that is devoid of edges that link two non-bridge nodes. Then, the problem of optimizing (7.1) is transformed into the following equivalent problem on this particular topology:

$$\begin{aligned} &\min_{\{w_k, w_b\}} \sum_{k=1}^{N} J_k(w) \\ &\text{subject to } w_k = w_b,\ b \in \mathcal{B},\ k \in \mathcal{N}_b \end{aligned} \tag{7.40}$$

This problem is subsequently solved using an augmented Lagrangian (or ADMM) technique and it leads to the following distributed algorithm, which involves the propagation of an additional dual variable, denoted here by $\boldsymbol{z}_{k,i}^b$:

$$\boldsymbol{y}_{k,i-1} = \mu\zeta|\mathcal{N}_k|\boldsymbol{w}_{k,i-1} + \mu \sum_{b\in\mathcal{N}_k} \left(\boldsymbol{z}_{k,i-1}^b - \zeta \boldsymbol{w}_{b,i-1}\right) \tag{7.41}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + 2\mu \boldsymbol{u}_{k,i}^{\mathsf{T}}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) - \boldsymbol{y}_{k,i-1} \tag{7.42}$$

$$\boldsymbol{w}_{b,i} = \frac{1}{\zeta|\mathcal{N}_b|} \sum_{k\in\mathcal{N}_b} (\boldsymbol{z}_{k,i-1}^b + \zeta \boldsymbol{w}_{k,i}) \tag{7.43}$$

$$\boldsymbol{z}_{k,i}^b = \boldsymbol{z}_{k,i-1}^b + \mu_b(\boldsymbol{w}_{k,i} - \boldsymbol{w}_{b,i}) \tag{7.44}$$

where $\{\mu, \mu_b, \zeta\}$ are step-size parameters and $|\mathcal{N}_k|$ denotes the cardinality of set $\mathcal{N}_k$. It is clear from the above equations that the structure of the resulting solution is more complex than the consensus and diffusion solutions from Examples 7.2 and 7.3. Observe in particular that the above algorithm requires the careful tuning of three parameters $\{\mu, \mu_b, \zeta\}$, as well as the propagation of several vectors, $\{\boldsymbol{y}_{k,i-1}, \boldsymbol{w}_{b,i}, \boldsymbol{z}_{k,i}^b, \boldsymbol{w}_{k,i}\}$. Moreover, the implementation requires a particular network structure with both regular and bridge nodes satisfying certain topological constraints. All these requirements are not needed in the consensus and diffusion solutions discussed earlier in Examples 7.2

and 7.3. More importantly, by explicitly incorporating the equality constraints (7.40) into the problem formulation, the resulting effect ends up limiting the learning abilities of the agents in general. This is because if data sensed by one agent is already reflecting drifts in the model while the data at the other agents is not, then by requiring the iterates to be matching can hinder the ability of the better informed agent to learn more thoroughly. One of the advantages of the consensus (7.9) and diffusion strategies (7.18)–(7.19) studied in this work is that, as the discussion in future chapters will reveal, they naturally lead to an equalization effect across the agents without added complexity — see, e.g., the explanation after future expression (11.138).

Finally, we remark that the distributed strategies described so far in this work are well-suited for cooperative networks where agents interact with each other to optimize an aggregate cost function. There are of course situations in which agents may behave in a selfish manner. In these cases, agents would participate in the collaborative process and share information with their neighbors only if cooperation is deemed beneficial to them (e.g., [102, 271]). We do not study this situation in the current work and focus instead on cooperative networks.

8

Evolution of Multi-Agent Networks

In this chapter we initiate our examination of the behavior and performance of multi-agent networks for adaptation, learning, and optimization. We divide the analysis in several consecutive chapters in order to emphasize in each chapter some relevant aspects that are unique to the networked solution. As the presentation will reveal, the study of the behavior of networked agents is more challenging than in the single-agent and centralized modes of operation due to at least two factors: (a) the coupling among interacting agents and (b) the fact that the networks are generally sparsely connected. When all is said and done, the results will help clarify the effect of network topology on performance and will present tools that enable the designer to compare various strategies against each other and against the centralized solution.

8.1 State Recursion for Network Errors

We pursue the performance analysis of networked solutions by examining how the error vectors across all agents evolve over time by means of a state recursion. We shall arrive at the *network* state evolution by collecting the error vectors from across all agents into

a single vector and by studying how the first, second, and fourth-order moments of this vector evolves over time. We shall carry out the analysis in a *unified* manner for both classes of consensus and diffusion algorithms by following the energy conservation arguments of [70, 71, 205, 206, 208, 277, 278]. We motivate the analysis by considering first, in this initial section, an illustrative example from [207, 208] dealing with MSE networks of the form described earlier in Example 6.3; these networks involve quadratic costs that share a common minimizer. Following the example, we extend the framework to more general costs in subsequent sections and chapters.

Example 8.1 (Error dynamics over MSE networks)**.** We consider the MSE network of Example 6.3, where each agent k observes realizations of zero-mean wide-sense jointly stationary data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$. The regression process $\boldsymbol{u}_{k,i}$ is $1 \times M$ and its covariance matrix is denoted by $R_{u,k} = \mathbb{E}\,\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i} > 0$. The measured data are assumed to be related to each other via the linear regression model:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i), \qquad k = 1, 2, \ldots, N \tag{8.1}$$

where $w^o \in \mathbb{C}^M$ is the unknown $M \times 1$ column vector that the agents wish to estimate. Moreover, the process $\boldsymbol{v}_k(i)$ is a zero-mean wide-sense stationary noise process with power $\sigma_{v,k}^2$ and assumed to be independent of $\boldsymbol{u}_{\ell,j}$ for all i, j, k, and ℓ. We associate with each agent the mean-square-error (quadratic) cost

$$J_k(w) = \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2 \tag{8.2}$$

We explained in Example 6.1 that this case corresponds to a situation where all individual costs, $J_k(w)$, have the same minimizer, which occurs at the location

$$w_k^o \;=\; w^o \;=\; R_{u,k}^{-1}r_{du,k} \tag{8.3}$$

Moreover, the Hessian matrix of each $J_k(w)$ is *block diagonal* and given by

$$\nabla_w^2 J_k(w) \;=\; \begin{bmatrix} R_{u,k} & 0 \\ 0 & R_{u,k}^{\mathsf{T}} \end{bmatrix} \tag{8.4}$$

We shall comment on the significance of this *block diagonal* structure after the example when we explain how to handle situations involving more general cost functions with Hessian matrices that are not necessarily block diagonal (or even independent of w, as is the case with (8.4)).

The update equations for the non-cooperative, consensus, and diffusion strategies are given by (3.13), (7.13), and (7.22)–(7.23). We list them in Table 8.1 for ease of reference.

Table 8.1: Update equations for non-cooperative, diffusion, and consensus strategies over MSE networks.

algorithm	**update equations**
non-cooperative	$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^{*}[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}]$
consensus	$\left\{ \begin{array}{rcl} \boldsymbol{\psi}_{k,i-1} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} &=& \boldsymbol{\psi}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^{*}[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \end{array} \right.$
CTA diffusion	$\left\{ \begin{array}{rcl} \boldsymbol{\psi}_{k,i-1} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} &=& \boldsymbol{\psi}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^{*}[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k,i-1}] \end{array} \right.$
ATC diffusion	$\left\{ \begin{array}{rcl} \boldsymbol{\psi}_{k,i} &=& \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^{*}[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{\psi}_{\ell,i} \end{array} \right.$

We capture the various strategies by a single *unifying* description by considering the following general algorithmic structure in terms of three sets of combination coefficients denoted by $\{a_{o,\ell k}, a_{1,\ell k}, a_{2,\ell k}\}$:

$$\left\{ \begin{array}{rcl} \boldsymbol{\phi}_{k,i-1} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{\psi}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\, \boldsymbol{\phi}_{\ell,i-1} \;+\; \mu_k \boldsymbol{u}_{k,i}^{*}\left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\phi}_{k,i-1}\right] \\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\boldsymbol{\psi}_{\ell,i} \end{array} \right. \tag{8.5}$$

In (8.5), the quantities $\{\boldsymbol{\phi}_{k,i-1}, \boldsymbol{\psi}_{k,i}\}$ denote $M \times 1$ intermediate variables, while the nonnegative entries of the $N \times N$ matrices:

$$A_o \triangleq [a_{o,\ell k}], \qquad A_1 \triangleq [a_{1,\ell k}], \qquad A_2 \triangleq [a_{2,\ell k}] \tag{8.6}$$

are assumed to satisfy the same conditions (7.10) and, hence, the matrices $\{A_o, A_1, A_2\}$ are *left-stochastic*. Any of the combination weights

$\{a_{o,\ell k}, a_{1,\ell k}, a_{2,\ell k}\}$ is zero whenever $\ell \notin \mathcal{N}_k$, where $\mathcal{N}_k$ denotes the set of neighbors of agent k. Different choices for $\{A_o, A_1, A_2\}$ correspond to different strategies, as the following list reveals and where we are introducing the matrix product $P = A_1 A_o A_2$:

$$\begin{aligned}
&\text{non-cooperative:} && A_1 = A_o = A_2 = I_N && \longrightarrow && P = I_N && (8.7)\\
&\text{consensus:} && A_o = A,\ \ A_1 = I_N = A_2 && \longrightarrow && P = A && (8.8)\\
&\text{CTA diffusion:} && A_1 = A,\ \ A_2 = I_N = A_o && \longrightarrow && P = A && (8.9)\\
&\text{ATC diffusion:} && A_2 = A,\ \ A_1 = I_N = A_o && \longrightarrow && P = A && (8.10)
\end{aligned}$$

We associate with each agent k the following three errors:

$$\widetilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i} \tag{8.11}$$

$$\widetilde{\boldsymbol{\psi}}_{k,i} \triangleq w^o - \boldsymbol{\psi}_{k,i} \tag{8.12}$$

$$\widetilde{\boldsymbol{\phi}}_{k,i-1} \triangleq w^o - \boldsymbol{\phi}_{k,i-1} \tag{8.13}$$

which measure the deviations from the desired solution w^o. Subtracting w^o from both sides of the equations in (8.5) and using (8.1) we get

$$\left\{\begin{aligned}
\widetilde{\boldsymbol{\phi}}_{k,i-1} &= \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\ \widetilde{\boldsymbol{w}}_{\ell,i-1}\\
\widetilde{\boldsymbol{\psi}}_{k,i} &= \sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\ \widetilde{\boldsymbol{\phi}}_{\ell,i-1} - \mu_k \boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i} \widetilde{\boldsymbol{\phi}}_{k,i-1} - \mu_k \boldsymbol{u}_{k,i}^* \boldsymbol{v}_k(i)\\
\widetilde{\boldsymbol{w}}_{k,i} &= \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\ \widetilde{\boldsymbol{\psi}}_{\ell,i}
\end{aligned}\right. \tag{8.14}$$

In a manner similar to (3.126), the gradient noise process at each agent k is given by

$$\boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1}) = \big(R_{u,k} - \boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i}\big)\,\widetilde{\boldsymbol{\phi}}_{k,i-1} - \boldsymbol{u}_{k,i}^* \boldsymbol{v}_k(i) \tag{8.15}$$

In order to examine the evolution of the error dynamics across the entire network, we collect the error vectors from all agents into $N \times 1$ block error vectors (whose individual entries are of size $M \times 1$ each):

$$\widetilde{\boldsymbol{w}}_i \triangleq \begin{bmatrix} \widetilde{\boldsymbol{w}}_{1,i} \\ \widetilde{\boldsymbol{w}}_{2,i} \\ \vdots \\ \widetilde{\boldsymbol{w}}_{N,i} \end{bmatrix}, \quad \widetilde{\boldsymbol{\psi}}_i \triangleq \begin{bmatrix} \widetilde{\boldsymbol{\psi}}_{1,i} \\ \widetilde{\boldsymbol{\psi}}_{2,i} \\ \vdots \\ \widetilde{\boldsymbol{\psi}}_{N,i} \end{bmatrix}, \quad \widetilde{\boldsymbol{\phi}}_{i-1} \triangleq \begin{bmatrix} \widetilde{\boldsymbol{\phi}}_{1,i-1} \\ \widetilde{\boldsymbol{\phi}}_{2,i-1} \\ \vdots \\ \widetilde{\boldsymbol{\phi}}_{N,i-1} \end{bmatrix} \tag{8.16}$$

The block quantities $\{\widetilde{\boldsymbol{\psi}}_i, \widetilde{\boldsymbol{\phi}}_{i-1}, \widetilde{\boldsymbol{w}}_i\}$ represent the state of the errors across the network at time i. Motivated by the last term in the second equation in

(8.14), and by the gradient noise terms (8.15), we also introduce the following $N \times 1$ column vectors whose entries are of size $M \times 1$ each:

$$\boldsymbol{z}_i \triangleq \begin{bmatrix} \boldsymbol{u}_{1,i}^{*}\boldsymbol{v}_1(i) \\ \boldsymbol{u}_{2,i}^{*}\boldsymbol{v}_2(i) \\ \vdots \\ \boldsymbol{u}_{N,i}^{*}\boldsymbol{v}_N(i) \end{bmatrix}, \qquad \boldsymbol{s}_i \triangleq \begin{bmatrix} \boldsymbol{s}_{1,i}(\boldsymbol{\phi}_{1,i-1}) \\ \boldsymbol{s}_{2,i}(\boldsymbol{\phi}_{2,i-1}) \\ \vdots \\ \boldsymbol{s}_{N,i}(\boldsymbol{\phi}_{N,i-1}) \end{bmatrix} \tag{8.17}$$

We further introduce the Kronecker products

$$\mathcal{A}_o \triangleq A_o \otimes I_M, \qquad \mathcal{A}_1 \triangleq A_1 \otimes I_M, \qquad \mathcal{A}_2 \triangleq A_2 \otimes I_M \tag{8.18}$$

The matrix $\mathcal{A}_o$ is an $N \times N$ *block* matrix whose $(\ell, k)-$th block is equal to $a_{o,\ell k} I_M$. Likewise, for $\mathcal{A}_1$ and $\mathcal{A}_2$. In other words, the Kronecker product transformations defined by (8.18) simply replace the matrices $\{A_o, A_1, A_2\}$ by block matrices $\{\mathcal{A}_o, \mathcal{A}_1, \mathcal{A}_2\}$ where each entry $\{a_{o,\ell k}, a_{1,\ell k}, a_{2,\ell k}\}$ in the original matrices is replaced by the diagonal matrices $\{a_{o,\ell k} I_M, a_{1,\ell k} I_M,\ a_{2,\ell k} I_M\}$.

We also introduce the following $N \times N$ *block* diagonal matrices, whose individual entries are of size $M \times M$ each:

$$\mathcal{M} \triangleq \mathrm{diag}\{\ \mu_1 I_M,\ \mu_2 I_M,\ \ldots,\ \mu_N I_M\ \} \tag{8.19}$$

$$\boldsymbol{\mathcal{R}}_i \triangleq \mathrm{diag}\left\{\ \boldsymbol{u}_{1,i}^{*}\boldsymbol{u}_{1,i},\ \boldsymbol{u}_{2,i}^{*}\boldsymbol{u}_{2,i},\ \ldots,\ \boldsymbol{u}_{N,i}^{*}\boldsymbol{u}_{N,i}\ \right\} \tag{8.20}$$

From (8.14), we can easily conclude that the block network variables (8.16) satisfy the relations:

$$\left\{ \begin{array}{rcl} \widetilde{\boldsymbol{\phi}}_{i-1} &=& \mathcal{A}_1^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1} \\ \widetilde{\boldsymbol{\psi}}_i &=& \left[\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right]\widetilde{\boldsymbol{\phi}}_{i-1} - \mathcal{M}\boldsymbol{z}_i \\ \widetilde{\boldsymbol{w}}_i &=& \mathcal{A}_2^{\mathsf{T}}\widetilde{\boldsymbol{\psi}}_i \end{array} \right. \tag{8.21}$$

so that the network weight error vector, $\widetilde{\boldsymbol{w}}_i$, ends up evolving according to the following *stochastic* state-space recursion:

$$\widetilde{\boldsymbol{w}}_i = \mathcal{A}_2^{\mathsf{T}}\left(\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right)\mathcal{A}_1^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{z}_i,\ \ i \geq 0 \ \ \text{(distributed)} \tag{8.22}$$

For comparison purposes, if each agent operates individually and uses the non-cooperative strategy (3.13), then the weight error vector across all N agents would instead evolve according to the following recursion:

$$\widetilde{\boldsymbol{w}}_i = \left(I_{MN} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right)\widetilde{\boldsymbol{w}}_{i-1} - \mathcal{M}\boldsymbol{z}_i,\ \ i \geq 0 \qquad \text{(non-cooperative)} \tag{8.23}$$

where the matrices $\{\mathcal{A}_o, \mathcal{A}_1, \mathcal{A}_2\}$ do not appear any longer, and with a block diagonal coefficient matrix $(I_{MN} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)$.

For later reference, it is straightforward to verify from (8.15) that

$$\boldsymbol{s}_i \;=\; (\mathcal{R}-\boldsymbol{\mathcal{R}}_i)\widetilde{\boldsymbol{\phi}}_{i-1} \;-\; \boldsymbol{z}_i \tag{8.24}$$

so that recursion (8.22) can be equivalently rewritten in the following form in terms of the gradient noise vector, $\boldsymbol{s}_i$, defined by (8.17):

$$\widetilde{\boldsymbol{w}}_i \;=\; \mathcal{B}\,\widetilde{\boldsymbol{w}}_{i-1} \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i \tag{8.25}$$

where we introduced the constant matrices

$$\mathcal{B} \;\triangleq\; \mathcal{A}_2^{\mathsf{T}}\left(\mathcal{A}_o^{\mathsf{T}}-\mathcal{M}\mathcal{R}\right)\mathcal{A}_1^{\mathsf{T}} \tag{8.26}$$

$$\mathcal{R} \;\triangleq\; \mathbb{E}\,\boldsymbol{\mathcal{R}}_i \;=\; \mathrm{diag}\{R_{u,1},\;R_{u,2},\;\ldots,\;R_{u,N}\} \tag{8.27}$$

■

Example 8.2 (Mean error behavior)**.** We continue with the formulation of Example 8.1. In mean-square-error analysis, we are interested in examining how the mean and variance of the weight-error vector evolve over time, namely, the quantities $\mathbb{E}\,\widetilde{\boldsymbol{w}}_i$ and $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$. If we refer back to the MSE data model described in Example 6.3, where the regression data $\{\boldsymbol{u}_{k,i}\}$ were assumed to be temporally white and independent over space, then the stochastic matrix $\boldsymbol{\mathcal{R}}_i$ appearing in (8.22)–(8.23) becomes statistically independent of $\widetilde{\boldsymbol{w}}_{i-1}$. Therefore, taking expectations of both sides of these recursions, and invoking the fact that $\boldsymbol{u}_{k,i}$ and $\boldsymbol{v}_k(i)$ are also independent of each other and have zero means (so that $\mathbb{E}\,\boldsymbol{z}_i = 0$), we conclude that the mean-error vectors evolve according to the following recursions [207]:

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i \;=\; \mathcal{B}\;(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}) \qquad \text{(distributed)} \tag{8.28}$$

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i \;=\; (I_{MN}-\mathcal{M}\mathcal{R})\;(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}) \qquad \text{(non-cooperative)} \tag{8.29}$$

The matrix $\mathcal{B}$ controls the dynamics of the mean weight-error vector for the distributed strategies. Observe, in particular, from (8.7)–(8.10) that $\mathcal{B}$ reduces to the following forms for the various strategies (non-cooperative (3.13), consensus (7.13), CTA diffusion (7.22), and ATC diffusion (7.23)):

$$\mathcal{B}_{\mathrm{ncop}} \;=\; I_{MN}-\mathcal{M}\mathcal{R} \tag{8.30}$$

$$\mathcal{B}_{\mathrm{cons}} \;=\; \mathcal{A}^{\mathsf{T}}-\mathcal{M}\mathcal{R} \tag{8.31}$$

$$\mathcal{B}_{\mathrm{atc}} \;=\; \mathcal{A}^{\mathsf{T}}\,(I_{MN}-\mathcal{M}\mathcal{R}) \tag{8.32}$$

$$\mathcal{B}_{\mathrm{cta}} \;=\; (I_{MN}-\mathcal{M}\mathcal{R})\,\mathcal{A}^{\mathsf{T}} \tag{8.33}$$

where $\mathcal{A} = A \otimes I_M$. ■

Example 8.3 (MSE networks with uniform agents)**.** We continue with Example 8.2 and show how the results simplify when all agents employ the same step-size, $\mu_k \equiv \mu$, and observe regression data with the same covariance matrix, $R_{u,k} \equiv R_u$. Note first that, in this case, we can express $\mathcal{M}$ and $\mathcal{R}$ from (8.19) and (8.27) in Kronecker product form as follows:

$$\mathcal{M} = \mu I_N \otimes I_M, \qquad \mathcal{R} = I_N \otimes R_u \tag{8.34}$$

so that expressions (8.30)–(8.33) reduce to

$$\left\{ \begin{array}{rcl} \mathcal{B}_{\rm ncop} &=& I_N \otimes (I_M \;-\; \mu R_u) \\ \mathcal{B}_{\rm cons} &=& A^{\mathsf{T}} \otimes I_M \;-\; \mu(I_M \otimes R_u) \\ \mathcal{B}_{\rm atc} &=& A^{\mathsf{T}} \otimes (I_M - \mu R_u) \\ \mathcal{B}_{\rm cta} &=& A^{\mathsf{T}} \otimes (I_M - \mu R_u) \end{array} \right. \tag{8.35}$$

For example, starting from (8.32) we have

$$\begin{aligned} \mathcal{B}_{\rm atc} &= \mathcal{A}^{\mathsf{T}} \left(I_{MN} - \mathcal{M}\mathcal{R}\right) \\ &= (A \otimes I_M)^{\mathsf{T}} \left[(I_N \otimes I_M) \;-\; (\mu I_N \otimes I_M)(I_N \otimes R_u)\right] \\ &= (A \otimes I_M)^{\mathsf{T}} \left[(I_N \otimes I_M) \;-\; \mu(I_N \otimes I_M)(I_N \otimes R_u)\right] \\ &= (A \otimes I_M)^{\mathsf{T}} \left[(I_N \otimes I_M) \;-\; \mu(I_N \otimes R_u)\right] \\ &= (A^{\mathsf{T}} \otimes I_M) \left[I_N \otimes (I_M - \mu R_u)\right] \\ &= A^{\mathsf{T}} \otimes (I_M - \mu R_u) \end{aligned} \tag{8.36}$$

where we used properties of the Kronecker product operation from Table F.1 in the appendix. Observe from (8.35) that $\mathcal{B}_{\rm atc} = \mathcal{B}_{\rm cta}$, so we denote these matrices by $\mathcal{B}_{\rm diff}$ whenever appropriate. Furthermore, using properties of the eigenvalues of Kronecker products of matrices, it can be verified that the MN eigenvalues of the above $\mathcal{B}$ matrices are given by the following expressions in terms of the eigenvalues of the component matrices $\{A, R_u\}$ for $k = 1, 2, \ldots N$ and $m = 1, 2, \ldots, M$:

$$\lambda(\mathcal{B}_{\rm diff}) = \lambda_k(A)\,[1 - \mu\lambda_m(R_u)] \tag{8.37}$$

$$\lambda(\mathcal{B}_{\rm cons}) = \lambda_k(A) - \mu\lambda_m(R_u) \tag{8.38}$$

$$\lambda(\mathcal{B}_{\rm ncop}) = 1 - \mu\lambda_m(R_u) \tag{8.39}$$

The expressions for $\lambda(\mathcal{B}_{\rm diff})$ and $\lambda(\mathcal{B}_{\rm ncop})$ follow directly from the properties of Kronecker products — see Table F.1. The expression for $\lambda(\mathcal{B}_{\rm cons})$ can be justified as follows. Let x_k and y_m denote right eigenvectors for A^{T} and R_u corresponding to the eigenvalues $\lambda_k(A)$ and $\lambda_m(R_u)$, respectively. Then, we again invoke properties of Kronecker products from Table F.1 in the appendix

to note that

$$\begin{aligned}\mathcal{B}_{\rm cons}(x_k\otimes y_m) &= \left[A^{\mathsf{T}}\otimes I_M \;-\; \mu(I_M\otimes R_u)\right](x_k\otimes y_m)\\ &= (A^{\mathsf{T}}x_k\otimes y_m) \;-\; \mu(x_k\otimes R_u y_m)\\ &= (\lambda_k(A)x_k\otimes y_m) \;-\; \mu(x_k\otimes \lambda_m(R_u)y_m)\\ &= \lambda_k(A)(x_k\otimes y_m) \;-\; \mu\lambda_m(R_u)(x_k\otimes y_m)\\ &= (\lambda_k(A)-\mu\lambda_m(R_u))(x_k\otimes y_m) \end{aligned} \tag{8.40}$$

so that $x_k\otimes y_m$ is an eigenvector for $\mathcal{B}_{\rm cons}$ with eigenvalue $\lambda_k(A)-\mu\lambda_m(R_u)$, as claimed.

■

Example 8.4 (Potential mean instability of consensus networks)**.** Consensus strategies can become unstable when used for adaptation purposes [207, 248]. This undesirable effect is already reflected in expressions (8.37)–(8.39). In particular, observe that the eigenvalues of A appear multiplying $(1-\mu\lambda_m(R_u))$ in expression (8.37) for diffusion. As such, and since $\rho(A)=1$ for any left-stochastic matrix, we conclude for this case of uniform agents that

$$\rho(\mathcal{B}_{\rm diff}) = \rho(\mathcal{B}_{\rm ncop}) \tag{8.41}$$

It follows that, regardless of the choice of the combination policy A, the diffusion strategies will be stable in the mean (i.e., $\mathbb{E}\,\widetilde{\boldsymbol{w}}_i$ will converge asymptotically to zero) whenever the individual non-cooperative agents are stable in the mean:

$$\text{individual agents stable} \implies \text{diffusion networks stable} \tag{8.42}$$

The same conclusion is not true for consensus networks; the individual agents can be stable and yet the consensus network can become unstable. This is because $\lambda_k(A)$ appears as an additive (rather than multiplicative) term in (8.38) (see [214, 248] and also future Examples 10.1 and 10.2):

$$\text{individual agents stable} \not\Rightarrow \text{consensus networks stable} \tag{8.43}$$

The fact that the combination matrix $\mathcal{A}^{\mathsf{T}}$ appears in an additive form in (8.31) is the result of the asymmetry that was mentioned earlier following (7.16) in the update equation for the consensus strategy. In contrast, the update equations for the diffusion strategies lead to $\mathcal{A}^{\mathsf{T}}$ appearing in a multiplicative form in (8.32)–(8.33). A more detailed example with a supporting simulation is discussed later in Example 10.2.

■

8.2 Network Limit Point and Pareto Optimality

Motivated by the discussion in the previous section on MSE networks, we now examine the evolution of distributed networks for the minimization of aggregate costs of the form

$$J^{\rm glob}(w) \triangleq \sum_{k=1}^{N} J_k(w) \tag{8.44}$$

where the individual costs, $J_k(w)$, and the aggregate cost are assumed to satisfy the conditions stated earlier in Assumption 6.1. We denote the unique minimizer of $J^{\rm glob}(w)$ by w^o; it is the unique solution to the algebraic equation:

$$\nabla_w J^{\rm glob}(w^o) = 0 \iff \sum_{k=1}^{N} \nabla_w J_k(w^o) = 0 \tag{8.45}$$

In the general case when the $J_k(w)$ are not necessarily quadratic in w, the Hessian matrices, $\nabla_w^2 J_k(w)$, need not be block diagonal anymore, as was the case with (8.4). Moreover, minimizers, w_k^o, of the individual costs, $J_k(w)$, need not agree with the global minimizer, w^o. Two complications arise as a result of these facts and they will need to be addressed. First, because the Hessian matrices are not generally block diagonal, it will turn out that the error quantities $\{\widetilde{\boldsymbol{w}}_{k,i}, \widetilde{\boldsymbol{\psi}}_{k,i}, \widetilde{\boldsymbol{\phi}}_{k,i-1}\}$, which were introduced in Example 8.1 and used to arrive at the state-space recursion (8.22), will not be sufficient anymore to fully capture the dynamics of the network in the general case for *complex data*. Extended versions of these vectors will need to be introduced. Second, and because the individual minimizers and the global minimizer are generally different, the distributed strategies will not converge to w^o but to another limit point, which we shall denote by $w^\star$ and whose value will be seen to be dependent on the network topology in an interesting way. We will identify $w^\star$ and explain under what conditions $w^\star$ and w^o agree with each other.

Unified Description

To begin with, and for ease of reference, we collect in Table 8.2 the

Table 8.2: Update equations for non-cooperative, diffusion, and consensus strategies.

algorithm	**update equations**
non-cooperative	$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \widehat{\nabla_{w^*} J}_k\,(\boldsymbol{w}_{k,i-1})$
consensus	$\begin{cases} \boldsymbol{\psi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu_k \widehat{\nabla_{w^*} J}_k\,(\boldsymbol{w}_{k,i-1}) \end{cases}$
CTA diffusion	$\begin{cases} \boldsymbol{\psi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu_k \widehat{\nabla_{w^*} J}_k\,(\boldsymbol{\psi}_{k,i-1}) \end{cases}$
ATC diffusion	$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{cases}$

equations that describe the non-cooperative (5.76), consensus (7.9), and diffusion strategies (7.18) and (7.19).

In a manner similar to (8.5), we can again describe these strategies by means of a single unifying description as follows:

$$\begin{cases} \boldsymbol{\phi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\,\boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{\psi}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\,\boldsymbol{\phi}_{\ell,i-1} - \mu_k \widehat{\nabla_{w^*} J}_k\left(\boldsymbol{\phi}_{k,i-1}\right) \\ \boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{cases} \tag{8.46}$$

where $\{\boldsymbol{\phi}_{k,i-1}, \boldsymbol{\psi}_{k,i}\}$ denote $M \times 1$ intermediate variables, while the nonnegative entries of the $N \times N$ matrices $A_o = [a_{o,\ell k}]$, $A_1 = [a_{1,\ell k}]$, and $A_2 = [a_{2,\ell k}]$ satisfy the same conditions (7.10) and, hence, the matrices $\{A_o, A_1, A_2\}$ are left-stochastic

$$A_o^{\mathsf{T}}\mathbb{1} = \mathbb{1}, \quad A_1^{\mathsf{T}}\mathbb{1} = \mathbb{1}, \quad A_2^{\mathsf{T}}\mathbb{1} = \mathbb{1} \tag{8.47}$$

We assume that each of these combination matrices defines an underly-

ing *connected* network topology so that none of their rows are identically zero. Again, different choices for $\{A_o, A_1, A_2\}$ correspond to different distributed strategies, as indicated earlier by (8.7)–(8.10), and where the left-stochastic matrix P represents the product:

$$P \triangleq A_1 A_o A_2 \tag{8.48}$$

Perron Eigenvector

We assume that P is a *primitive* matrix. For example, this condition is automatically guaranteed if the combination matrix A in the selections (8.8)–(8.10) is primitive, which in turn is guaranteed for strongly-connected networks. It then follows from the Perron-Frobenius Theorem [27, 113, 189] that we can characterize the eigen-structure of P in the following manner — see Lemma F.4 in the appendix:

(a) The matrix P has a *single* eigenvalue at one.

(b) All other eigenvalues of P are strictly inside the unit circle so that $\rho(P) = 1$.

(c) With proper sign scaling, all entries of the right-eigenvector of P corresponding to the single eigenvalue at one are *positive*. Let p denote this right-eigenvector, with its entries $\{p_k\}$ normalized to add up to one, i.e.,

$$Pp = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0, \quad k = 1, 2, \ldots, N \tag{8.49}$$

We refer to p as the *Perron eigenvector* of P.

Weighted Aggregate Cost

Following [68–70], we next introduce the vector:

$$q \triangleq \mathrm{diag}\{\mu_1, \mu_2, \ldots, \mu_N\} A_2 p \tag{8.50}$$

It is clear that all entries of q are strictly positive since each $\mu_k > 0$ and the entries of $A_2 p$ are all positive. The latter statement follows from the fact that each entry of $A_2 p$ is a linear combination of the positive

entries of p. Therefore, if we denote the individual entries of the vector q by $\{q_k\}$, then it holds that

$$q_k > 0, \qquad k = 1, 2, \ldots, N \tag{8.51}$$

We also represent the step-sizes as scaled multiples of the same factor $\mu_{\max}$, namely,

$$\mu_k \triangleq \tau_k \, \mu_{\max}, \qquad k = 1, 2, \ldots, N \tag{8.52}$$

where $0 < \tau_k \leq 1$. In this way, it becomes clear that all step-sizes become smaller as $\mu_{\max}$ is reduced in size.

We further introduce the weighted aggregate cost

$$J^{\text{glob},\star}(w) \triangleq \sum_{k=1}^{N} q_k J_k(w) \tag{8.53}$$

Since all the $J_k(w)$ are convex in w, then the strong convexity of $J^{\text{glob}}(w)$ guarantees the strong convexity of $J^{\text{glob},\star}(w)$. Indeed, note that

$$\begin{aligned} \nabla_w^2 \, J^{\text{glob},\star}(w) &= \sum_{k=1}^{N} q_k \nabla_w^2 J_k(w) \\ &\geq q_{\min} \cdot \left(\sum_{k=1}^{N} \nabla_w^2 J_k(w) \right) \\ &\overset{(6.13)}{\geq} q_{\min} \, \frac{\nu_d}{h} \, I_{hM} > 0 \end{aligned} \tag{8.54}$$

where $q_{\min}$ is the smallest entry of q and is strictly positive; moreover, $h = 1$ for real data and $h = 2$ for complex data. It follows that $J^{\text{glob},\star}(w)$ will have a unique global minimum, which we denote by $w^\star$ and it satisfies:

$$\nabla_w \, J^{\text{glob},\star}(w^\star) = 0 \iff \sum_{k=1}^{N} q_k \nabla_w \, J_k(w^\star) = 0 \tag{8.55}$$

In general, the minimizers $\{w^o, w^\star\}$ of $J^{\text{glob}}(w)$ and $J^{\text{glob},\star}(w)$, respectively, are different. However, they will coincide in some important cases such as:

(a) When the $\{q_k\}$ are equal to each other. This situation occurs, for example, when $\mu_k \equiv \mu$ across all agents and the matrices $\{A_o, A_1, A_2\}$ are doubly-stochastic (in which case the Perron eigenvector is given by $p = \mathbb{1}/N$). A second situation is discussed in Example 8.10.

(b) When the individual costs, $J_k(w)$, are all minimized at the *same* location, as was the case with the MSE networks of Example 8.1.

The arguments in future chapters will establish that the location $w^\star$ serves as the limit point for the networked solution in the mean-square-error sense. Specifically, if we now measure (or define) the errors relative to $w^\star$, say, as:

$$\widetilde{\boldsymbol{w}}_{k,i} \triangleq w^\star - \boldsymbol{w}_{k,i}, \quad k = 1, 2, \ldots, N \tag{8.56}$$

then we will be arguing later (see future expression (9.11)) that:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \;=\; O(\mu_{\max}) \tag{8.57}$$

so that the size of the (variance of the) error is in the order of $\mu_{\max}$ and can be made arbitrarily small for smaller step-sizes. In particular, by calling upon Markov's inequality and using an argument similar to (4.53), we would be able to conclude that each $\boldsymbol{w}_{k,i}$ approaches $w^\star$ asymptotically with high probability for sufficiently small step-sizes.

Example 8.5 (Normalization of weights in aggregate cost)**.** If desired, we may normalize the positive weighting coefficients $\{q_k\}$ defined by (8.50) to have their sum add up to one, say, by introducing instead the coefficients:

$$\bar{q}_k \triangleq q_k / \sum_{k=1}^{N} q_k \tag{8.58}$$

and replacing (8.53) by the convex combination:

$$\bar{J}^{\text{glob},\star}(w) \triangleq \sum_{k=1}^{N} \bar{q}_k J_k(w) \tag{8.59}$$

Clearly, both aggregate functions, $J^{\text{glob},\star}(w)$ defined by (8.53) and $\bar{J}^{\text{glob},\star}(w)$, are scaled multiples of each other and, hence, their unique minimizers occur

at the same location $w^\star$. One advantage of working with the normalized aggregate cost (8.59) is that when all individual costs happen to coincide, say, $J_k(w) \equiv J(w)$, then expression (8.59) reduces to

$$\bar{J}^{\text{glob},\star}(w) = J(w) \tag{8.60}$$

whereas $J^{\text{glob},\star}(w)$ will be a scaled multiple of $J(w)$.

Since $J^{\text{glob},\star}(w)$ and $\bar{J}^{\text{glob},\star}(w)$ have the same global minimizer $w^\star$, we will continue to work with the un-normalized definition (8.53) for the remainder of this chapter, and also in Chapters 9 and 10 where we examine the stability of multi-agent networks and the convergence of their iterates towards $w^\star$. We will find it more convenient to employ the normalized representation (8.59) in Chapter 11 when we examine the excess-risk performance of these networks. ■

Example 8.6 (Weighted aggregate cost for consensus and diffusion)**.** The expression for q simplifies for the particular choices of $\{A_o, A_1, A_2\}$ shown in (8.7)–(8.10) for consensus and diffusion, which involve a single left-stochastic and primitive combination matrix A. In all three cases we obtain $P = A$ so that the vector p is the Perron eigenvector that is associated with A:

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0 \tag{8.61}$$

Moreover, expression (8.50) reduces to

$$q_k \triangleq \mu_k p_k > 0, \quad k = 1, 2, \ldots, N \tag{8.62}$$

so that each q_k is simply a scaled multiple of the corresponding p_k. The *weighted* aggregate cost (8.53) then becomes

$$J^{\text{glob},\star}(w) \triangleq \sum_{k=1}^{N} \mu_k p_k J_k(w) \tag{8.63}$$

When A is doubly stochastic so that $p_k = 1/N$, we obtain

$$J^{\text{glob},\star}(w) \triangleq \frac{\mu_{\max}}{N} \left(\sum_{k=1}^{N} \tau_k J_k(w) \right) \tag{8.64}$$

where we used $\mu_k = \tau_k \, \mu_{\max}$. It is seen that even the use of different step-sizes across the agents is sufficient to steer the limit point away from w^o. ■

Interpretation as Pareto Solution

As already explained in [67, 69], the unique vector $w^\star$ that solves (8.55)

can be interpreted as corresponding to a Pareto optimal solution for the collection of convex functions $\{J_k(w)\}$. To explain why this is the case, let us first review briefly the concept of Pareto optimality.

Recall that we are denoting by w_k^o the minimizers for the individual costs, $J_k(w)$. In general, the minimizers $\{w_k^o,\ k = 1, 2, \ldots, N\}$ are distinct from each other. In order for cooperation among the agents to be meaningful, we need to seek some solution vector $w^\star$ that is "optimal" in some sense for the entire network. One useful concept of optimality is the one known as *Pareto optimality* (see, e.g., [45, 120, 272]). A solution $w^\star$ is said to be Pareto optimal for all N agents if there does not exist any other vector, $w^\bullet$, that dominates $w^\star$, i.e., that satisfies the following two conditions:

$$J_k(w^\bullet) \leq J_k(w^\star), \quad \text{for all } k \in \{1, 2, \ldots, N\} \tag{8.65}$$

$$J_{k^o}(w^\bullet) < J_{k^o}(w^\star), \quad \text{for at least one } k^o \in \{1, 2, \ldots, N\} \tag{8.66}$$

In other words, any other vector $w^\bullet$ that improves one of the costs, say, $J_{k^o}(w^\bullet) < J_{k^o}(w^\star)$, will necessarily degrade the performance of some other cost, i.e., $J_k(w^\bullet) > J_k(w^\star)$ for some $k \neq k^o$. In this way, solutions $w^\star$ that are Pareto optimal are such that no agent in the cooperative network can have its performance improved by moving away from $w^\star$ without degrading the performance of some other agent.

To illustrate this concept, let us consider an example from [69] corresponding to $N = 2$ agents with the argument $w \in \mathbb{R}$ being real-valued and scalar. Let the set

$$\mathcal{S} \triangleq \{\ J_1(w),\ J_2(w)\ \} \subset \mathbb{R}^2 \tag{8.67}$$

denote the achievable cost values over all feasible choices of $w \in \mathbb{R}$; each point $S \in \mathcal{S}$ belongs to the two-dimensional space $\mathbb{R}^2$ and represents values attained by the cost functions $\{J_1(w), J_2(w)\}$ for a particular w. The shaded areas in Figure 8.1 represent the set $\mathcal{S}$ for two situations of interest. The plot on the left represents the situation in which the two cost functions $J_1(w)$ and $J_2(w)$ achieve their minima at the *same* location, namely, $w_1^o = w_2^o$. This location is indicated by the point $S_o = \{J_1(w^o); J_2(w^o)\}$ in the figure, where w^o denotes the common minimizer. In comparison, the plot on the right represents the

situation in which the two cost functions $J_1(w)$ and $J_2(w)$ achieve their minima at two distinct locations, w_1^o and w_2^o. Point S_1 in the figure indicates the location where $J_1(w)$ attains its minimum value, while point S_2 indicates the location where $J_2(w)$ attains its minimum value. In this case, the two cost functions do not have a common minimizer. It is easy to verify that all points that lie on the heavy curve between points S_1 and S_2 are Pareto optimal solutions for $\{J_1(w), J_2(w)\}$. For example, starting at some arbitrary point B on the curve, if we want to reduce the value of $J_1(w)$ without increasing the value of $J_2(w)$, then we will need to move out of the achievable set $\mathcal{S}$ towards point C, which is not feasible. The alternative choice to reducing the value of $J_1(w)$ is to move from B on the curve to another Pareto optimal point, such as point D. This move, while feasible, it would increase the value of $J_2(w)$. In this way, we would need to trade the value of $J_2(w)$ for $J_1(w)$. For this reason, the curve from S_1 to S_2 is called the optimal tradeoff curve (or optimal tradeoff surface when $N > 2$) [45, p.183].

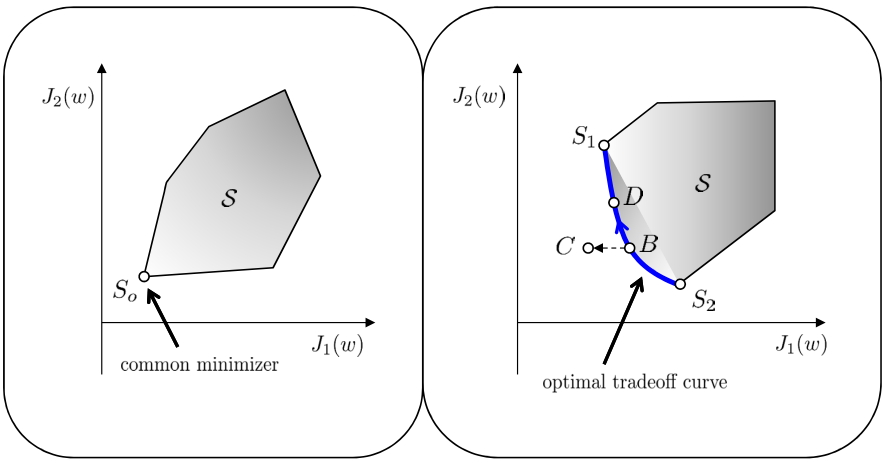

Figure 8.1: Pareto optimal points for the case $N = 2$. In the figure on the left, point S denotes the optimal point where both cost functions are minimized simultaneously. In the figure on the right, all points that lie on the heavy boundary curve are Pareto optimal solutions.

As we see from the tradeoff curve in Figure 8.1, Pareto optimal solutions are generally non-unique. One useful method to determine a Pareto optimal solution is a *scalarization* technique, whereby an *aggregate* cost function is first formed as the weighted sum of the component convex cost functions as follows [45, 272]:

$$J^{\text{glob},\pi}(w) \triangleq \sum_{k=1}^{N} \pi_k \, J_k(w) \tag{8.68}$$

where the $\{\pi_k\}$ are positive scalars. It is shown in [45, p.183] that the unique minimizer, which we denote by w^π, for the above aggregate cost corresponds to a Pareto optimal solution for the collection of convex costs $\{J_k(w), \;\; k = 1, 2, \ldots, N\}$. Moreover, by varying the values of the $\{\pi_k\}$, we are able to determine different Pareto optimal solutions from the tradeoff curve. If we now compare expression (8.68) with the earlier aggregate cost (8.53), we conclude that the solution $w^\star$ can be interpreted as the Pareto optimal solution that corresponds to selecting the parameters $\pi_k = q_k$.

Example 8.7 (Pareto optimal solutions for mean-square-error costs)**.** We illustrate the concept of Pareto optimality for quadratic cost functions of the form:

$$J_k(w) \;=\; \sigma_d^2 \;-\; r_{du,k}^* w - w^* r_{du,k} + w^* R_{u,k} w, \quad k = 1, 2, \ldots, N \tag{8.69}$$

where $w \in \mathbb{C}^M$, $R_{u,k} > 0$, and $r_{du,k} \in \mathbb{C}^M$. By setting $\nabla_w J_k(w) = 0$, we find that the minimizer of each $J_k(w)$ occurs at the vector location

$$w_k^o \;=\; R_{u,k}^{-1} r_{du,k} \tag{8.70}$$

Since the moments $\{r_{du,k}, R_{u,k}\}$ can differ across the agents, these individual minimizers need not coincide. Pareto optimal solutions can be found by minimizing the aggregate cost function (8.68) for any collection of weights $\{\pi_k > 0\}$. Setting the gradient vector of $J^{\text{glob},\pi}(w)$ to zero we arrive at the following expression for Pareto optimal solutions in this case:

$$w^\pi \;=\; \left(\sum_{k=1}^{N} \pi_k R_{u,k}\right)^{-1} \left(\sum_{k=1}^{N} \pi_k \, r_{du,k}\right) \tag{8.71}$$

Using (8.70), the above Pareto optimal solution can be expressed as the combination:

$$w^{\pi} = \sum_{k=1}^{N} B_k \, w_k^o \tag{8.72}$$

where

$$B_k \triangleq \left(\sum_{\ell=1}^{N} \pi_\ell R_{u,\ell} \right)^{-1} (\pi_k R_{u,k}), \quad k = 1, 2, \ldots, N \tag{8.73}$$

Observe that the matrix coefficients $\{B_k\}$ satisfy:

$$B_k > 0, \qquad \sum_{k=1}^{N} B_k = I_M \tag{8.74}$$

so that expression (8.72) amounts to a convex combination calculation.

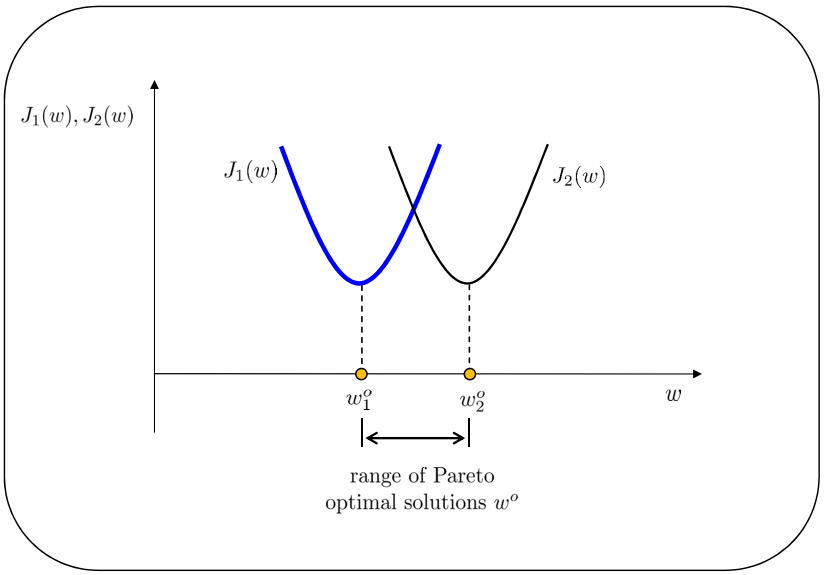

Figure 8.2: Two quadratic cost functions of a scalar real parameter w with minima at locations $w = w_1^o$ and $w = w_2^o$. As shown by (8.75), the set of Pareto optimal solutions in this case consists of all parameters w within the interval $w \in (w_1^o, w_2^o)$.

Figure 8.2 illustrates this conclusion for the case of two cost functions ($N = 2$) and a scalar parameter $w \in \mathbb{R}$. In this case, we denote the covariance

matrices $\{R_{u,1}, R_{u,2}\}$ by the positive scalars $\{\sigma_{u,1}^2, \sigma_{u,2}^2\}$ so that expression (8.72) becomes

$$w^{\pi} \;=\; \left(\frac{\pi_1\sigma_{u,1}^2}{\pi_1\sigma_{u,1}^2+\pi_2\sigma_{u,2}^2}\right) w_1^o \;+\; \left(\frac{\pi_2\sigma_{u,2}^2}{\pi_1\sigma_{u,1}^2+\pi_2\sigma_{u,2}^2}\right) w_2^o \tag{8.75}$$

Observe that the set of Pareto optimal solutions defined by (8.75) consists of convex combinations of $\{w_1^o, w_2^o\}$.

■

Example 8.8 (Pareto optimal solutions for MSE networks)**.** Let us consider a *variation* of the MSE networks defined in Example 6.3 where the data model at each agent is now assumed to be given by:

$$\boldsymbol{d}_k(i) \;=\; \boldsymbol{u}_{k,i} w_k^o \;+\; \boldsymbol{v}_k(i) \tag{8.76}$$

with the model vector w_k^o being possibly different at the various agents. If we multiply both sides of the above equation by $\boldsymbol{u}_{k,i}^*$ and take expectations, we find that w_k^o satisfies

$$r_{du,k} = R_{u,k} w_k^o, \quad k = 1, 2, \ldots, N \tag{8.77}$$

in terms of the second-order moments:

$$r_{du,k} = \mathbb{E}\, \boldsymbol{d}_k(i)\boldsymbol{u}_{k,i}^*, \qquad R_{u,k} = \mathbb{E}\, \boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i} \tag{8.78}$$

The individual cost function associated with each agent k continues to be the mean-square-error cost, $J_k(w) = \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w|^2$, so that

$$\begin{aligned} \nabla_w J_k(w) \;&=\; R_{u,k} w - r_{du,k} \\ &\overset{(8.77)}{=}\; R_{u,k}(w - w_k^o) \end{aligned} \tag{8.79}$$

We assume that all agents in the network are running either the consensus strategy (7.14) or the diffusion strategy (7.22) or (7.23). These strategies correspond to the choices $\{A_o, A_1, A_2\}$ shown earlier in (8.7)–(8.10) in terms of a single combination matrix A, namely,

$$\text{consensus:} \qquad A_o = A, \;\; A_1 = I_N = A_2 \tag{8.80}$$

$$\text{CTA diffusion:} \qquad A_1 = A, \;\; A_2 = I_N = A_o \tag{8.81}$$

$$\text{ATC diffusion:} \qquad A_2 = A, \;\; A_1 = I_N = A_o \tag{8.82}$$

In these cases, the Perron eigenvector p defined by (8.49) will correspond to the Perron eigenvector associated with A:

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0 \tag{8.83}$$

Consequently, the entries q_k defined by (8.50) will reduce to

$$q_k \;=\; \mu_k p_k \tag{8.84}$$

The resulting Pareto optimal solution, $w^\star$, is given by the unique solution to (8.55), which reduces to the following expression in the current scenario:

$$\sum_{k=1}^{N} \mu_k p_k R_{u,k}(w^\star - w_k^o) \;=\; 0 \tag{8.85}$$

or, equivalently,

$$w^\star \;=\; \left(\sum_{k=1}^{N} \mu_k p_k R_{u,k}\right)^{-1} \left(\sum_{k=1}^{N} \mu_k p_k R_{u,k} w_k^o\right) \tag{8.86}$$

If we assume that the regression covariance matrices are of the form $R_{u,k} = \sigma_{u,k}^2 I_M$, for some variances $\sigma_{u,k}^2 > 0$, then the above expression simplifies to the convex combination:

$$w^\star \;=\; \sum_{k=1}^{N} \pi_k w_k^o \tag{8.87}$$

where the scalar combination coefficients, $\{\pi_k\}$, are nonnegative, add up to one, and are given by:

$$\pi_k \;\triangleq\; \mu_k p_k \sigma_{u,k}^2 \left(\sum_{k=1}^{N} \mu_k p_k \sigma_{u,k}^2\right)^{-1}, \quad k = 1, 2, \ldots, N \tag{8.88}$$

We illustrate these results numerically for the case of the averaging (uniform) combination policy with uniform step-sizes across the agents, $\mu_k \equiv \mu$. In the uniform policy, the combination weights $\{a_{\ell k}\}$ are selected according to the averaging rule:

$$a_{\ell k} \;=\; \begin{cases} 1/n_k, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{8.89}$$

where

$$n_k \;\triangleq\; |\mathcal{N}_k| \tag{8.90}$$

denotes the size of the neighborhood of agent k (or its degree). In this case, all neighbors of agent k are assigned the same weight, $1/n_k$, and the matrix A will be left-stochastic. The entries of the corresponding Perron eigenvector can be verified to be

$$p_k \;=\; n_k \left(\sum_{m=1}^{N} n_m\right)^{-1} \tag{8.91}$$

Then, expression (8.88) gives

$$\pi_k \triangleq n_k \sigma_{u,k}^2 \left(\sum_{k=1}^{N} n_k \sigma_{u,k}^2 \right)^{-1}, \quad k = 1, 2, \ldots, N \tag{8.92}$$

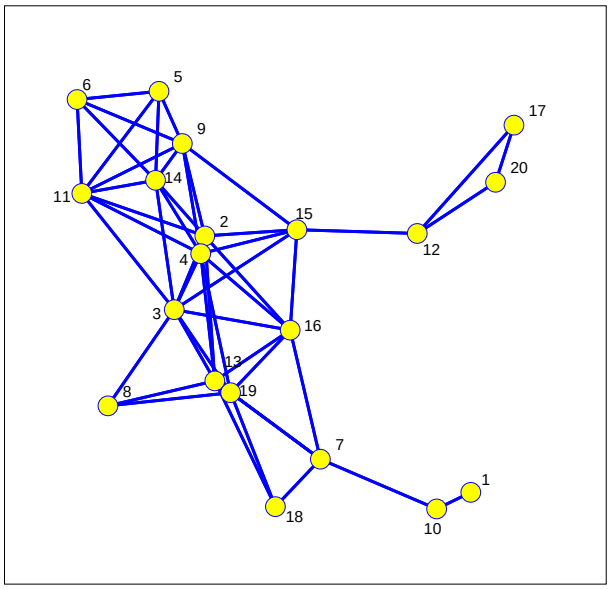

Figure 8.3: A connected network topology consisting of $N = 20$ agents employing the averaging rule (8.89). Each agent k is assumed to belong its neighborhood $\mathcal{N}_k$. It follows that the network is strongly-connected.

Figure 8.3 shows the connected network topology with $N = 20$ agents used for this simulation, with the measurement noise variances, $\{\sigma_{v,k}^2\}$, and the power of the regression data, $\{\sigma_{u,k}^2 I_M\}$, shown in the right and left plots of Figure 8.4, respectively. All agents are assumed to have a non-trivial self-loop so that the neighborhood of each agent includes the agent itself as well. The resulting network is therefore strongly-connected.

Figure 8.5 plots the evolution of the ensemble-average learning curves, $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, relative to the Pareto optimal solution $w^\star$ defined by (8.87) and (8.92), for consensus, ATC diffusion, and CTA diffusion using $\mu = 0.001$. The measure $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ corresponds to the average mean-square-deviation (MSD)

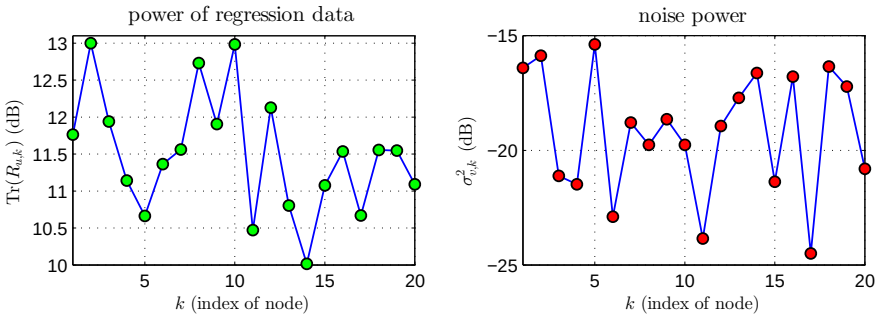

Figure 8.4: Measurement noise profile (right) and regression data power (left) across all agents in the network. The covariance matrices are assumed to be of the form $R_{u,k} = \sigma^2_{u,k} I_M$, and the noise and regression data are Gaussian distributed in this simulation.

across all agents at time i since

$$\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \tag{8.93}$$

and $\widetilde{\boldsymbol{w}}_{k,i} = w^\star - \boldsymbol{w}_{k,i}$. The learning curves are obtained by averaging the trajectories $\{\frac{1}{N}\|\widetilde{\boldsymbol{w}}_i\|^2\}$ over 200 repeated experiments. The label on the vertical axis in the figure refers to the learning curves $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ by writing $\mathrm{MSD}_{\mathrm{dist,av}}(i)$, with an iteration index i and where the subscripts "dist" and "av" are meant to indicate that this is an average performance measure for a distributed solution. Each experiment in this simulation involves running the consensus (7.14) or diffusion (7.22)–(7.23) LMS recursions with $h = 2$ on complex-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ generated according to the model $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o_k + \boldsymbol{v}_k(i)$, with $M = 10$. The unknown vectors $\{w^o_k\}$ are generated randomly and their norms are normalized to one. It is observed in the figure that the learning curves tend to the MSD value predicted by future expression (11.175).

■

Example 8.9 (Controlling the limit point — Hastings rule)**.** We observe from (8.55) that the limit point $w^\star$ is dependent on the scaling coefficients $\{q_k\}$, which in turn depend on the choice of the combination matrices $\{A_o, A_1, A_2\}$ through their dependence on the Perron eigenvector, p. Therefore, once the combination policies are selected, the limit point for the network is fixed at the unique solution $w^\star$ of (8.53).

Let us illustrate the reverse direction in which it is desirable to select the

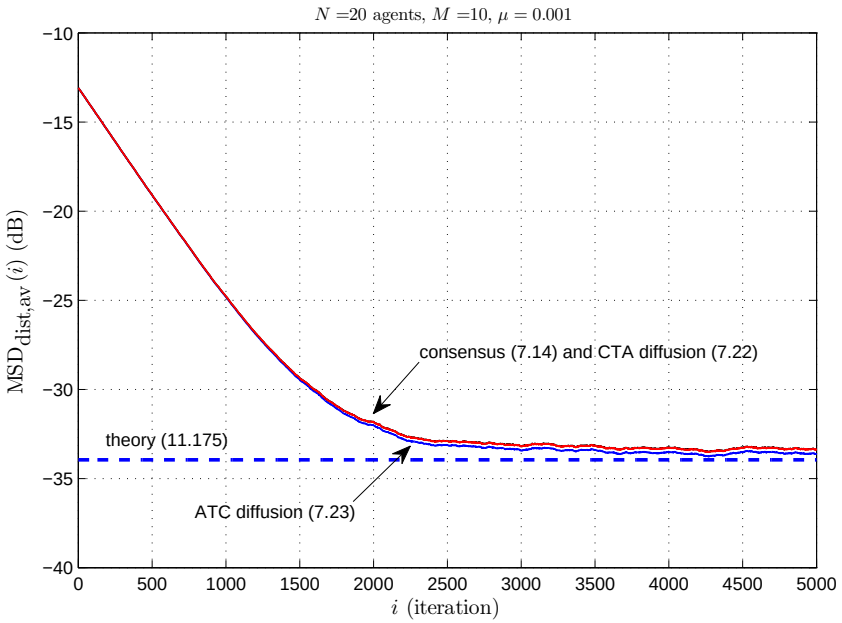

Figure 8.5: Evolution of the learning curves for three strategies: consensus (7.14), CTA diffusion (7.22), and ATC diffusion (7.23), with all agents employing the same step-size $\mu = 0.001$ and the averaging combination policy.

combination policy to attain a particular Pareto optimal solution. We illustrate the construction for the case of consensus and diffusion strategies, which correspond to the choices $\{A_o, A_1, A_2\}$ shown earlier in (8.7)–(8.10). Again, in these cases, the Perron eigenvector p defined by (8.49) will correspond to the Perron eigenvector associated with A:

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0 \tag{8.94}$$

Consequently, the entries q_k defined by (8.50) will reduce to

$$q_k \;=\; \mu_k p_k \tag{8.95}$$

Now assume we are given a collection of positive scaling coefficients $\{q'_k\}$. These coefficients define a unique solution, $w^\star$, to the algebraic equation (8.55) defined in terms of these $\{q'_k\}$. Assume further that we are given a connected network topology and we would like to determine a left-stochastic combination matrix, A, that would lead to the coefficients $\{q'_k\}$, or to some scaled multiples of them. That is, we would like to determine A such that the $\{q_k\}$ that result

from the construction (8.94)–(8.95) would coincide with, or be multiples of, the given $\{q'_k\}$. To answer this question, we call upon the following useful result. Given a set of positive scalars $\{q'_k,\ k=1,2,\ldots,N\}$ and a connected network with N agents, it is explained in [68, 276], using a construction procedure from [35, 42, 106], that one way to construct a left-stochastic matrix A that leads to (a scaled multiple of) the given coefficients $\{q'_k\}$ is as follows (we refer to the resulting matrix A as the Hastings combination rule) — see also future Lemma 12.2:

$$a_{\ell k} \;=\; \begin{cases} \dfrac{\mu_k/q'_k}{\max\{\ n_k\mu_k/q'_k,\ n_\ell\mu_\ell/q'_\ell\ \}}, & \ell\in\mathcal{N}_k\backslash\{k\} \\ 1-\displaystyle\sum_{m\in\mathcal{N}_k\backslash\{k\}} a_{mk}, & \ell=k \end{cases} \tag{8.96}$$

where the $\{\mu_k\}$ represent step-size parameters, and the scalar n_k in (8.96) denotes the cardinality of $\mathcal{N}_k$ (also called the degree of agent k and is equal to the number of neighbors that k has):

$$n_k \;\triangleq\; |\mathcal{N}_k| \tag{8.97}$$

It can be verified that the entries of the Perron eigenvector, p, of this matrix A are given by — see the proof of Lemma 12.2:

$$p_k \;=\; \frac{q'_k}{\mu_k}\left(\sum_{\ell=1}^{N}\frac{q'_\ell}{\mu_\ell}\right)^{-1} \tag{8.98}$$

so that the products $\mu_k p_k$ are proportional to the given q'_k, as desired.

A particular case of interest is when we want to determine a combination matrix A that leads to a uniform value for the $\{q'_k\}$, i.e., $q'_k\equiv q'$ for $k=1,2,\ldots,N$. In this case, the minimizers of $J^{\rm glob}(w)$ and $J^{{\rm glob},\star}(w)$ defined by (8.44) and (8.53) will coincide, namely, $w^\star=w^o$, and construction (8.96) will reduce to

$$a_{\ell k} \;=\; \begin{cases} \dfrac{\mu_k}{\max\{\ n_k\mu_k,\ n_\ell\mu_\ell\ \}}, & \ell\in\mathcal{N}_k\backslash\{k\} \\ 1-\displaystyle\sum_{m\in\mathcal{N}_k\backslash\{k\}} a_{mk}, & \ell=k \end{cases} \tag{8.99}$$

In the special case when the step-sizes are uniform across all agents, $\mu_k\equiv\mu$ for $k=1,2,\ldots,N$, then the step-sizes disappear from (8.99) and the above expression reduces to the so-called Metropolis rule (e.g., [106, 167, 265]), which

is doubly-stochastic:

$$a_{\ell k} = \begin{cases} \dfrac{1}{\max\{n_k, n_\ell\}}, & \ell \in \mathcal{N}_k \backslash \{k\} \\ 1 - \displaystyle\sum_{m \in \mathcal{N}_k \backslash \{k\}} a_{mk}, & \ell = k \end{cases} \tag{8.100}$$

■

Example 8.10 (Controlling the limit point — power iteration)**.** We continue with the setting of Example 8.9 for consensus or diffusion strategies, which correspond to the choices $\{A_o, A_1, A_2\}$ shown earlier in (8.7)–(8.10). Example 8.9 showed one method to select the combination policy A according to the Hastings rule (8.99)–(8.100) in order to ensure that the distributed implementation (8.46) will converge towards the minimizer, w^o, of the original aggregate cost (8.44) and not towards the limit point $w^\star$ from (8.55). This method, however, assumes that the designer is free to select the combination policy, A.

If, on the other hand, we are already given a combination policy that cannot be modified, then we can resort to an alternative method that relies on selecting the step-size parameters, μ_k [72]. Specifically, from (8.95) we observe that the $\{q_k\}$ can be made uniform by selecting

$$\mu_k = \frac{\mu_o}{p_k}, \quad k = 1, 2 \ldots, N \tag{8.101}$$

where $\mu_o > 0$ is some positive scaling parameter. This construction results in $q_k \equiv \mu_o$. Consequently, under (8.101), recursion (8.46) for ATC diffusion becomes (similarly, for CTA diffusion or consensus):

$$\begin{cases} \boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{k,i-1} - \frac{\mu_o}{p_k} \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \, \boldsymbol{\psi}_{\ell,i} \end{cases} \tag{8.102}$$

By doing so, the above distributed solution will now converge in the mean-square-error sense towards the minimizer of the weighted aggregate cost (8.53) that results from replacing q_k by μ_o so that

$$J^{\text{glob},\star}(w) = \mu_o \left(\sum_{k=1}^{N} J_k(w) \right) = \mu_o \cdot J^{\text{glob}}(w) \tag{8.103}$$

and, hence, $w^\star = w^o$, as desired.

The challenge in running (8.102) is that the implementation requires knowledge of the Perron entries, $\{p_k\}$. For some combination policies, this

information is readily available. For example, for the averaging rule (8.89), we can use expression (8.91) for p_k to conclude that we can run the above algorithm by using μ_o/n_k instead of μ_o/p_k, where n_k is the degree of agent k. The factor that appears in the denominator of p_k in (8.89) is common to all agents and can be incorporated into μ_o (in this way, recursion (8.102) can run with knowledge of only the local information n_k). For more general left-stochastic combination matrices A, one can run a power iteration [104] in parallel with the distributed implementation (8.102) in order to estimate the entries p_k. The power iteration involves a recursion of the following form:

$$r_i = Ar_{i-1}, \quad r_{-1} \neq 0, \quad i \geq 0 \tag{8.104}$$

with coefficient matrix equal to A and with an initial nonzero vector r_{-1} that is selected randomly. We denote the entries of r_i by $\{r_k(i)\}$ for $k = 1, 2 \ldots, N$.

Since we are assuming A to be primitive, then it has a unique eigenvalue at one and, moreover, this eigenvalue is dominant (i.e., its magnitude is strictly larger than the magnitude of each of the other eigenvalues of A). Then, the power iteration is known to converge towards a right-eigenvector of A that corresponds to its largest-magnitude eigenvalue, which is the eigenvalue at one [104, 263]. That is, the entries $\{r_k(i)\}$ converge towards a constant multiple of the corresponding entries $\{p_k\}$. Therefore, we may replace the scalars $\{p_k\}$ in (8.102) by the values $\{r_k(i)\}$ estimated recursively and in a distributed manner, as shown in the following listing for each agent k (the constant scaling between the values of $r_k(i)$ and p_k is incorporated into μ_o since the scaling is common to all agents):

$$\left\{ \begin{array}{rcl} r_k(i) & = & \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{k\ell}\, r_k(i-1) \\ \boldsymbol{\psi}_{k,i} & = & \boldsymbol{w}_{k,i-1} \; - \; \frac{\mu_o}{r_k(i)} \, \widehat{\nabla_{w^*} J}_k \left(\boldsymbol{w}_{k,i-1} \right) \\ \boldsymbol{w}_{k,i} & = & \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{\psi}_{\ell,i} \end{array} \right. \tag{8.105}$$

Observe that implementation (8.105) employs two sets of coefficients: $\{a_{k\ell}\}$ in the first line and $\{a_{\ell k}\}$ in the last line. The first set corresponds to the entries on the $k-$th row of A, while the second set corresponds to the entries on the $k-$th column of A; these latter entries add up to one and perform a convex combination operation. Therefore, this second method assumes that each agent k has access to both sets of coefficients $\{a_{\ell k}, a_{k\ell}\}$, which is feasible for undirected graphs. This construction is related to, albeit different from, a push-sum protocol used for computing the average value of distributed measurements over directed graphs in, e.g., [23, 78, 140, 173, 240].

■

8.3 Gradient Noise Model

From this point onwards, we shall therefore measure the performance of the distributed strategy (8.46) by using $w^\star$ as the reference vector (instead of w^o) and define the error vectors as:

$$\widetilde{\boldsymbol{w}}_{k,i} \triangleq w^\star - \boldsymbol{w}_{k,i} \tag{8.106}$$

$$\widetilde{\boldsymbol{\psi}}_{k,i} \triangleq w^\star - \boldsymbol{\psi}_{k,i} \tag{8.107}$$

$$\widetilde{\boldsymbol{\phi}}_{k,i-1} \triangleq w^\star - \boldsymbol{\phi}_{k,i-1} \tag{8.108}$$

Moreover, with each agent k, we associate a gradient noise vector in addition to a mismatch (or bias) vector, namely,

$$\boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1}) \triangleq \widehat{\nabla_{w^*} J_k}(\boldsymbol{\phi}_{k,i-1}) - \nabla_{w^*} J_k(\boldsymbol{\phi}_{k,i-1}) \tag{8.109}$$

and

$$b_k \triangleq -\nabla_{w^*} J_k(w^\star) \tag{8.110}$$

In the special case when all individual costs, $J_k(w)$, have the same minimizer at $w_k^o \equiv w^o$ (which is the situation considered in Example 8.1 over MSE networks), then $w^\star = w^o$ and the vector b_k will be identically zero. In general, though, the vector b_k is nonzero. Let $\boldsymbol{\mathcal{F}}_{i-1}$ represent the collection of all random events generated by the processes $\{\boldsymbol{w}_{k,j}\}$ at all agents $k = 1, 2, \ldots, N$ up to time $i-1$:

$$\boldsymbol{\mathcal{F}}_{i-1} \triangleq \text{filtration}\{\boldsymbol{w}_{k,-1}, \boldsymbol{w}_{k,0}, \boldsymbol{w}_{k,1}, \ldots, \boldsymbol{w}_{k,i-1}, \text{ all } k\} \tag{8.111}$$

Similarly to Assumption 5.2, we assume that the gradient noise processes across the agents satisfy the following conditions.

Assumption 8.1 (Conditions on gradient noise)**.** It is assumed that the first and second-order conditional moments of the individual gradient noise processes, $\boldsymbol{s}_{k,i}(\boldsymbol{\phi})$, satisfy the following conditions for any iterates $\boldsymbol{\phi} \in \boldsymbol{\mathcal{F}}_{i-1}$ and for all $k, \ell = 1, 2, \ldots, N$:

$$\mathbb{E}\,[\,\boldsymbol{s}_{k,i}(\boldsymbol{\phi})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,] = 0 \tag{8.112}$$

$$\mathbb{E}\,\big[\boldsymbol{s}_{k,i}(\boldsymbol{\phi})\boldsymbol{s}_{\ell,i}^*(\boldsymbol{\phi})|\boldsymbol{\mathcal{F}}_{i-1}\big] = 0, \quad k \neq \ell \tag{8.113}$$

$$\mathbb{E}\,\big[\boldsymbol{s}_{k,i}(\boldsymbol{\phi})\boldsymbol{s}_{\ell,i}^{\mathsf{T}}(\boldsymbol{\phi})|\boldsymbol{\mathcal{F}}_{i-1}\big] = 0, \quad k \neq \ell \tag{8.114}$$

$$\mathbb{E}\,\big[\,\|\boldsymbol{s}_{k,i}(\boldsymbol{\phi})\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\big] \leq \big(\bar{\beta}_k/h\big)^2\,\|\boldsymbol{\phi}\|^2 + \bar{\sigma}_{s,k}^2 \tag{8.115}$$

almost surely, for some nonnegative scalars $\bar{\beta}_k^2$ and $\bar{\sigma}_{s,k}^2$ and where $h = 1$ for real data and $h = 2$ for complex data.

Using the above conditions, and in a manner similar to the derivation (3.28), it is straightforward to verify that the gradient noise processes satisfy:

$$\mathbb{E}\left[\, \boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1}) \,|\, \boldsymbol{\mathcal{F}}_{i-1} \,\right] = 0 \tag{8.116}$$

$$\mathbb{E}\left[\|\boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1})\|^2 \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \leq (\beta_k^2/h^2)\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^2 + \sigma_{s,k}^2 \tag{8.117}$$

$$\mathbb{E}\,\|\boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1})\|^2 \leq (\beta_k^2/h^2)\mathbb{E}\,\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^2 + \sigma_{s,k}^2 \tag{8.118}$$

in terms of the scalars

$$\beta_k^2 \triangleq 2\bar{\beta}_k^2 \tag{8.119}$$

$$\sigma_{s,k}^2 \triangleq 2(\bar{\beta}_k/h)^2\|w^\star\|^2 + \bar{\sigma}_{s,k}^2 \tag{8.120}$$

We shall use conditions (8.116)–(8.118) more frequently in lieu of (8.112)–(8.115). We could have required these conditions directly in the statement of Assumption 8.1. We instead opted to state conditions (8.112)–(8.115) in that manner, in terms of a generic $\boldsymbol{\phi} \in \boldsymbol{\mathcal{F}}_{i-1}$ rather than $\widetilde{\boldsymbol{w}}_{k,i-1}$, so that the upper bound in (8.115) is independent of the unknown $w^\star$.

Conditions (8.116)–(8.118) will be useful in establishing the mean-square stability of the second-order moment of the error vector, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, in the next chapter. Later, in Sec. 9.2, when we examine the stability of the fourth-order moment of the same error vector, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^4$, we will need to replace the bound (8.115) by a condition similar to (5.36) on the fourth-order moments of the individual gradient noise processes, namely, by the following condition:

$$\mathbb{E}\left[\,\|\boldsymbol{s}_{k,i}(\boldsymbol{\phi})\|^4 \,|\, \boldsymbol{\mathcal{F}}_{i-1}\,\right] \leq (\bar{\beta}_k/h)^4\,\|\boldsymbol{\phi}\|^4 + \bar{\sigma}_{s,k}^4 \tag{8.121}$$

almost surely, for nonnegative scalars $\{\bar{\beta}_k^4, \bar{\sigma}_{s,k}^4\}$. Using an argument similar to (3.56), we can similarly conclude from these conditions that

$$\mathbb{E}\left[\,\|\boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1})\|^4 \,|\, \boldsymbol{\mathcal{F}}_{i-1}\,\right] \leq (\beta_{4,k}^4/h^4)\,\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^4 + \sigma_{s4,k}^4 \tag{8.122}$$

for some non-negative parameters defined by:

$$\beta_{4,k}^4 \triangleq 8\bar{\beta}_k^4 \tag{8.123}$$

$$\sigma_{s4,k}^4 \triangleq 8(\bar{\beta}_k/h)^4 \|w^\star\|^4 + \bar{\sigma}_{s4,k}^4 \tag{8.124}$$

We will not need to introduce condition (8.121) in addition to the second-order moment condition (8.115). This is because, as explained earlier following (3.50), condition (8.121) implies that condition (8.115) also holds, namely, it follows from (8.121) that

$$\mathbb{E}\left[\|\boldsymbol{s}_{k,i}(\boldsymbol{\phi})\|^2 \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \leq (\bar{\beta}_k/h)^2 \|\boldsymbol{\phi}\|^2 + \bar{\sigma}_{s,k}^2 \tag{8.125}$$

Example 8.11 (Gradient noise over MSE networks)**.** Let us continue with the setting of Example 8.8, which deals with a *variation* of MSE networks where the data model at each agent is instead assumed to be given by

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w_k^o + \boldsymbol{v}_k(i) \tag{8.126}$$

with the model vectors, w_k^o, being possibly different at the various agents. In a manner similar to (8.15), we can verify that if the distributed strategy (8.5) is employed at the agents, then the resulting gradient noise process at each agent k is now given by:

$$\boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1}) = \frac{2}{h}\left(R_{u,k} - \boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i}\right)\left(w_k^o - \boldsymbol{\phi}_{k,i-1}\right) - \frac{2}{h}\boldsymbol{u}_{k,i}^* \boldsymbol{v}_k(i) \tag{8.127}$$

where $h = 2$ for complex data and $h = 1$ for real data (in the latter case, it is understood that complex conjugation should be replaced by standard transposition so that $\boldsymbol{u}_{k,i}^*$ becomes $\boldsymbol{u}_{k,i}^{\mathsf{T}}$). Observe that (8.127) is written in terms of the difference $w_k^o - \boldsymbol{\phi}_{k,i-1}$ and not in terms of the error vector $\widetilde{\boldsymbol{\phi}}_{k,i-1}$. ■

8.4 Extended Network Error Dynamics

We explained earlier after (8.45) that because the Hessian matrices, $\nabla_w^2 J_k(w)$, are not generally block diagonal, we will need to introduce extended versions of the error quantities $\{\widetilde{\boldsymbol{w}}_{k,i}, \widetilde{\boldsymbol{\psi}}_{k,i}, \widetilde{\boldsymbol{\phi}}_{k,i-1}\}$ in order to fully capture the dynamics of the network in the general case. This is in contrast to the mean-square-error case studied in Example 8.1 where these errors were sufficient to arrive at the state recursions (8.22) or (8.25) for the evolution of the network dynamics.

To motivate the need for extended error vectors, let us first introduce some notation. Thus, note that if we express any column vector $w \in \mathbb{C}^M$ in terms of its real and imaginary parts $x, y \in \mathbb{R}^M$, then

$$
\begin{aligned}
w &= x + jy \qquad \text{(a column vector)} && (8.128)\\
w^* &= x^{\mathsf{T}} - jy^{\mathsf{T}} \qquad \text{(a row vector)} && (8.129)\\
(w^*)^{\mathsf{T}} &= x - jy \qquad \text{(a column vector)} && (8.130)
\end{aligned}
$$

In other words, the quantity $(w^*)^{\mathsf{T}}$ is again a *column* vector, just like w, except that its complex representation is obtained by replacing j by $-j$. The reason why we need to introduce the quantity $(w^*)^{\mathsf{T}}$ is because, as the discussion will reveal, we will need to track the evolution of both quantities $\boldsymbol{w}_{k,i}$ and $(\boldsymbol{w}_{k,i}^*)^{\mathsf{T}}$ in the general case in order to examine how the network is performing. Thus, using equations (8.46), we can deduce similar relations for the evolution of the complex conjugate iterates, namely,

$$
\left\{
\begin{aligned}
\left(\boldsymbol{\phi}_{k,i-1}^*\right)^{\mathsf{T}} &= \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\left(\boldsymbol{w}_{\ell,i-1}^*\right)^{\mathsf{T}}\\
\left(\boldsymbol{\psi}_{k,i}^*\right)^{\mathsf{T}} &= \sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\left(\boldsymbol{\phi}_{\ell,i-1}^*\right)^{\mathsf{T}} - \mu_k\widehat{\nabla_{w^{\mathsf{T}}}J}_k\left(\boldsymbol{\phi}_{k,i-1}\right)\\
\left(\boldsymbol{w}_{k,i}^*\right)^{\mathsf{T}} &= \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\left(\boldsymbol{\psi}_{\ell,i}^*\right)^{\mathsf{T}}
\end{aligned}
\right. \tag{8.131}
$$

Observe how the gradient vector approximation that appears in the second equation now involves differentiation relative to w^{T} and not w^*. Representations (8.46) and (8.131) can be grouped together into a single set of equations by introducing extended vectors of dimensions $2M \times 1$ as follows:

$$
\left\{
\begin{aligned}
\begin{bmatrix} \boldsymbol{\phi}_{k,i-1} \\ (\boldsymbol{\phi}_{k,i-1}^*)^{\mathsf{T}} \end{bmatrix} &= \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\begin{bmatrix} \boldsymbol{w}_{\ell,i-1} \\ \left(\boldsymbol{w}_{\ell,i-1}^*\right)^{\mathsf{T}} \end{bmatrix}\\
\begin{bmatrix} \boldsymbol{\psi}_{k,i} \\ (\boldsymbol{\psi}_{k,i}^*)^{\mathsf{T}} \end{bmatrix} &= \sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\begin{bmatrix} \boldsymbol{\phi}_{\ell,i-1} \\ (\boldsymbol{\phi}_{\ell,i-1}^*)^{\mathsf{T}} \end{bmatrix} - \mu_k\begin{bmatrix} \widehat{\nabla_{w^*}J}_k(\boldsymbol{\phi}_{k,i-1}) \\ \widehat{\nabla_{w^{\mathsf{T}}}J}_k\left(\boldsymbol{\phi}_{k,i-1}\right) \end{bmatrix}\\
\begin{bmatrix} \boldsymbol{w}_{k,i} \\ \left(\boldsymbol{w}_{k,i}^*\right)^{\mathsf{T}} \end{bmatrix} &= \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\begin{bmatrix} \boldsymbol{\psi}_{\ell,i} \\ (\boldsymbol{\psi}_{\ell,i}^*)^{\mathsf{T}} \end{bmatrix}
\end{aligned}
\right. \tag{8.132}
$$

We therefore extend the error vectors into size $2M \times 1$ and introduce

$$\widetilde{\boldsymbol{w}}_{k,i}^{e} \triangleq \begin{bmatrix} \widetilde{\boldsymbol{w}}_{k,i} \\ (\widetilde{\boldsymbol{w}}_{k,i}^{*})^{\mathsf{T}} \end{bmatrix}, \quad \widetilde{\boldsymbol{\psi}}_{k,i}^{e} \triangleq \begin{bmatrix} \widetilde{\boldsymbol{\psi}}_{k,i} \\ (\widetilde{\boldsymbol{\psi}}_{k,i}^{*})^{\mathsf{T}} \end{bmatrix}, \quad \widetilde{\boldsymbol{\phi}}_{k,i-1}^{e} \triangleq \begin{bmatrix} \widetilde{\boldsymbol{\phi}}_{k,i-1} \\ (\widetilde{\boldsymbol{\phi}}_{k,i-1}^{*})^{\mathsf{T}} \end{bmatrix} \tag{8.133}$$

where we are using the superscript "e" to refer to extended quantities of size $2M \times 1$. We also introduce extended versions of the limit vector, the gradient noise vector, and the bias vector:

$$(w^{\star})^{e} \triangleq \begin{bmatrix} w^{\star} \\ ((w^{\star})^{*})^{\mathsf{T}} \end{bmatrix}, \quad \boldsymbol{s}_{k,i}^{e} \triangleq \begin{bmatrix} \boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1}) \\ \left(\boldsymbol{s}_{k,i}^{*}(\boldsymbol{\phi}_{k,i-1})\right)^{\mathsf{T}} \end{bmatrix}, \quad b_{k}^{e} \triangleq \begin{bmatrix} b_{k} \\ (b_{k}^{*})^{\mathsf{T}} \end{bmatrix} \tag{8.134}$$

where the vector $\boldsymbol{s}_{k,i}^{e}$ in (8.134) should have been written more explicitly as $\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})$; we are dropping the argument for compactness of notation. Now, subtracting $(w^{\star})^{e}$ from both sides of the equations in (8.132) and using (8.109) gives

$$\begin{cases} \widetilde{\boldsymbol{\phi}}_{k,i-1}^{e} &= \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{1,\ell k}\, \widetilde{\boldsymbol{w}}_{\ell,i-1}^{e} \\ \widetilde{\boldsymbol{\psi}}_{k,i}^{e} &= \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{o,\ell k}\, \widetilde{\boldsymbol{\phi}}_{\ell,i-1}^{e} + \mu_k \begin{bmatrix} \nabla_{w^*} J_k(\boldsymbol{\phi}_{k,i-1}) \\ \nabla_{w^{\mathsf{T}}} J_k\left(\boldsymbol{\phi}_{k,i-1}\right) \end{bmatrix} + \mu_k \boldsymbol{s}_{k,i}^{e} \\ \widetilde{\boldsymbol{w}}_{k,i}^{e} &= \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{2,\ell k}\, \widetilde{\boldsymbol{\psi}}_{\ell,i}^{e} \end{cases} \tag{8.135}$$

We observe that the gradient vectors in (8.135) are being evaluated at the intermediate variable, $\boldsymbol{\phi}_{k,i-1}$, and not at any of the *error* variables. For this reason, equation (8.135) is still not an actual recursion. To transform it into a recursion that only involves error variables, we call upon the mean-value theorem (D.20) from the appendix, which allows us to write:

$$\begin{bmatrix} \nabla_{w^*} J_k(\boldsymbol{\phi}_{k,i-1}) \\ \\ \nabla_{w^\mathsf{T}} J_k(\boldsymbol{\phi}_{k,i-1}) \end{bmatrix} = \underbrace{\begin{bmatrix} \nabla_{w^*} J_k(w^\star) \\ \\ \nabla_{w^\mathsf{T}} J_k(w^\star) \end{bmatrix}}_{\triangleq -b_k^e} - \underbrace{\left[\int_0^1 \nabla_w^2 J_k(w^\star - t\widetilde{\boldsymbol{\phi}}_{k,i-1})dt\right]}_{\triangleq \boldsymbol{H}_{k,i-1}} \widetilde{\boldsymbol{\phi}}_{k,i-1}^e \tag{8.136}$$

That is,

$$\begin{bmatrix} \nabla_{w^*} J_k(\boldsymbol{\phi}_{k,i-1}) \\ \\ \nabla_{w^\mathsf{T}} J_k(\boldsymbol{\phi}_{k,i-1}) \end{bmatrix} = -b_k^e - \boldsymbol{H}_{k,i-1}\widetilde{\boldsymbol{\phi}}_{k,i-1}^e \tag{8.137}$$

in terms of a $2M \times 2M$ stochastic matrix $\boldsymbol{H}_{k,i-1}$ defined in terms of the integral of the $2M \times 2M$ Hessian matrix of agent k:

$$\boldsymbol{H}_{k,i-1} \triangleq \int_0^1 \nabla_w^2 J_k(w^\star - t\widetilde{\boldsymbol{\phi}}_{k,i-1})dt \tag{8.138}$$

Substituting (8.137) into (8.135) leads to

$$\begin{cases} \widetilde{\boldsymbol{\phi}}_{k,i-1}^e &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\, \widetilde{\boldsymbol{w}}_{\ell,i-1}^e \\ \\ \widetilde{\boldsymbol{\psi}}_{k,i}^e &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\, \widetilde{\boldsymbol{\phi}}_{\ell,i-1}^e - \mu_k \boldsymbol{H}_{k,i-1}\widetilde{\boldsymbol{\phi}}_{k,i-1}^e - \mu_k b_k^e + \mu_k \boldsymbol{s}_{k,i}^e \\ \\ \widetilde{\boldsymbol{w}}_{k,i}^e &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\, \widetilde{\boldsymbol{\psi}}_{\ell,i}^e \end{cases} \tag{8.139}$$

These equations describe the evolution of the error quantities at the individual agents for $k = 1, 2, \ldots, N$. Observe that when the matrix $\boldsymbol{H}_{k,i-1}$ happens to be block diagonal, which occurs when the Hessian matrix function itself is block diagonal (as happened in (8.4) with the quadratic costs in Example 8.1), then the last term in (8.137) decouples into two separate terms in the variables

$$\left\{ \widetilde{\boldsymbol{\phi}}_{k,i-1},\ \left(\widetilde{\boldsymbol{\phi}}_{k,i-1}^*\right)^\mathsf{T} \right\} \tag{8.140}$$

since then

$$\boldsymbol{H}_{k,i-1}\widetilde{\boldsymbol{\phi}}_{k,i-1}^e \equiv \begin{bmatrix} \boldsymbol{H}_{k,i-1}^{11} & 0 \\ 0 & \boldsymbol{H}_{k,i-1}^{22} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{\phi}}_{k,i-1} \\ \left(\widetilde{\boldsymbol{\phi}}_{k,i-1}^*\right)^\mathsf{T} \end{bmatrix} \tag{8.141}$$

In that case, it becomes unnecessary to propagate the extended vectors $\{\widetilde{\boldsymbol{w}}_{k,i}^{e},\ \widetilde{\boldsymbol{\psi}}_{k,i}^{e},\ \widetilde{\boldsymbol{\phi}}_{k,i-1}^{e}\}$ using (8.139); the dynamics of the network can be studied by examining solely the evolution of the original error vectors $\{\widetilde{\boldsymbol{w}}_{k,i},\ \widetilde{\boldsymbol{\psi}}_{k,i},\ \widetilde{\boldsymbol{\phi}}_{k,i-1}\}$, namely,

$$\begin{cases} \widetilde{\boldsymbol{\phi}}_{k,i-1} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\,\widetilde{\boldsymbol{w}}_{\ell,i-1} \\ \widetilde{\boldsymbol{\psi}}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\,\widetilde{\boldsymbol{\phi}}_{\ell,i-1} - \mu_k \boldsymbol{H}_{k,i-1}^{11}\widetilde{\boldsymbol{\phi}}_{k,i-1} - \mu_k b_k + \mu_k \boldsymbol{s}_{k,i} \\ \widetilde{\boldsymbol{w}}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\,\widetilde{\boldsymbol{\psi}}_{\ell,i} \end{cases} \tag{8.142}$$

We continue our discussion by treating the general case (8.139). We collect the extended error vectors from all agents into the following $N\times 1$ block error vectors (whose individual entries are of size $2M\times 1$ each):

$$\widetilde{\boldsymbol{w}}_i^{e} \triangleq \begin{bmatrix} \widetilde{\boldsymbol{w}}_{1,i}^{e} \\ \widetilde{\boldsymbol{w}}_{2,i}^{e} \\ \vdots \\ \widetilde{\boldsymbol{w}}_{N,i}^{e} \end{bmatrix}, \quad \widetilde{\boldsymbol{\phi}}_{i-1}^{e} \triangleq \begin{bmatrix} \widetilde{\boldsymbol{\phi}}_{1,i-1}^{e} \\ \widetilde{\boldsymbol{\phi}}_{2,i-1}^{e} \\ \vdots \\ \widetilde{\boldsymbol{\phi}}_{N,i-1}^{e} \end{bmatrix}, \quad \widetilde{\boldsymbol{\psi}}_i^{e} \triangleq \begin{bmatrix} \widetilde{\boldsymbol{\psi}}_{1,i}^{e} \\ \widetilde{\boldsymbol{\psi}}_{2,i}^{e} \\ \vdots \\ \widetilde{\boldsymbol{\psi}}_{N,i}^{e} \end{bmatrix} \tag{8.143}$$

We also define the following block gradient noise and bias vectors:

$$\boldsymbol{s}_i^{e} \triangleq \begin{bmatrix} \boldsymbol{s}_{1,i}^{e} \\ \boldsymbol{s}_{2,i}^{e} \\ \vdots \\ \boldsymbol{s}_{N,i}^{e} \end{bmatrix}, \quad b^{e} \triangleq \begin{bmatrix} b_1^{e} \\ b_2^{e} \\ \vdots \\ b_N^{e} \end{bmatrix} \tag{8.144}$$

Now recall from the explanation after (8.134) that each entry, $\boldsymbol{s}_{k,i}^{e}$, in (8.144) is dependent on $\boldsymbol{\phi}_{k,i-1}$. Recall also from the distributed algorithm (8.46) that $\boldsymbol{\phi}_{k,i-1}$ is a combination of various $\{\boldsymbol{w}_{\ell,i-1}\}$. Therefore, the block gradient vector, $\boldsymbol{s}_i^{e}$, defined in (8.144) is dependent on

the network vector, $\boldsymbol{w}_{i-1}^e$, namely,

$$\boldsymbol{w}_{i-1}^e \triangleq \begin{bmatrix} \boldsymbol{w}_{1,i-1}^e \\ \boldsymbol{w}_{2,i-1}^e \\ \vdots \\ \boldsymbol{w}_{N,i-1}^e \end{bmatrix}, \quad \boldsymbol{w}_{k,i-1}^e \triangleq \begin{bmatrix} \boldsymbol{w}_{k,i-1} \\ \left(\boldsymbol{w}_{k,i-1}^*\right)^{\mathsf{T}} \end{bmatrix} \tag{8.145}$$

For this reason, we shall also write $\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)$ rather than simply $\boldsymbol{s}_i^e$ when it is desired to highlight the dependency of $\boldsymbol{s}_i^e$ on $\boldsymbol{w}_{i-1}^e$.

We further introduce the Kronecker products

$$\begin{cases} \mathcal{A}_o \triangleq A_o \otimes I_{2M} \\ \mathcal{A}_1 \triangleq A_1 \otimes I_{2M} \\ \mathcal{A}_2 \triangleq A_2 \otimes I_{2M} \end{cases} \tag{8.146}$$

The matrix $\mathcal{A}_o$ is an $N \times N$ *block* matrix whose $(\ell, k)-$th block is equal to $a_{o,\ell k} I_{2M}$. Similarly, for $\mathcal{A}_1$ and $\mathcal{A}_2$. Likewise, we introduce the following $N \times N$ *block* diagonal matrices, whose individual entries are of size $2M \times 2M$ each:

$$\mathcal{M} \triangleq \mathrm{diag}\{\, \mu_1 I_{2M},\ \mu_2 I_{2M},\ \ldots,\ \mu_N I_{2M} \,\} \tag{8.147}$$

$$\boldsymbol{\mathcal{H}}_{i-1} \triangleq \mathrm{diag}\,\{\, \boldsymbol{H}_{1,i-1},\ \boldsymbol{H}_{2,i-1},\ \ldots,\ \boldsymbol{H}_{N,i-1} \,\} \tag{8.148}$$

We then conclude from (8.139) that the following relations hold for the *network* variables:

$$\begin{cases} \widetilde{\boldsymbol{\phi}}_{i-1}^e &= \mathcal{A}_1^{\mathsf{T}} \widetilde{\boldsymbol{w}}_{i-1}^e \\ \widetilde{\boldsymbol{\psi}}_i^e &= \left[\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\right] \widetilde{\boldsymbol{\phi}}_{i-1}^e + \mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) - \mathcal{M}b^e \\ \widetilde{\boldsymbol{w}}_i^e &= \mathcal{A}_2^{\mathsf{T}} \widetilde{\boldsymbol{\psi}}_i^e \end{cases} \tag{8.149}$$

so that the network weight error vector, $\widetilde{\boldsymbol{w}}_i^e$, ends up evolving according to the following *stochastic* recursion over $i \geq 0$:

$$\widetilde{\boldsymbol{w}}_i^e = \mathcal{A}_2^{\mathsf{T}} \left(\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\right) \mathcal{A}_1^{\mathsf{T}} \widetilde{\boldsymbol{w}}_{i-1}^e + \mathcal{A}_2^{\mathsf{T}} \mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) - \mathcal{A}_2^{\mathsf{T}} \mathcal{M} b^e \tag{8.150}$$

For comparison purposes, if each agent operates individually and uses the non-cooperative strategy, then the weight error vectors across all

N agents would instead evolve according to the following stochastic recursion:

$$\widetilde{\boldsymbol{w}}_i^e \;=\; (I_{2MN} - \mathcal{M}\boldsymbol{\mathcal{H}}_{i-1})\,\widetilde{\boldsymbol{w}}_{i-1}^e \;+\; \mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \;-\; \mathcal{M}b^e \tag{8.151}$$

where the matrices $\{\mathcal{A}_o, \mathcal{A}_1, \mathcal{A}_2\}$ do not appear since, in this case, $A_o = A_1 = A_2 = I_N$. We summarize the discussion so far in the following statement for complex data (we show how these results simplify for real data in the example after the lemma).

Lemma 8.1 (Network error dynamics). Consider a network of N interacting agents running the distributed strategy (8.46). The evolution of the error dynamics across the network relative to the reference vector $w^\star$ defined by (8.55) is described by the following recursion:

$$\widetilde{\boldsymbol{w}}_i^e \;=\; \boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^e \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e, \;\; i \geq 0 \tag{8.152}$$

where

$$\boldsymbol{\mathcal{B}}_{i-1} \;\triangleq\; \mathcal{A}_2^{\mathsf{T}}\left(\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\right)\mathcal{A}_1^{\mathsf{T}} \tag{8.153}$$

$$\mathcal{A}_o \;\triangleq\; A_o \otimes I_{2M}, \quad \mathcal{A}_1 \;\triangleq\; A_1 \otimes I_{2M}, \quad \mathcal{A}_2 \;\triangleq\; A_2 \otimes I_{2M} \tag{8.154}$$

$$\mathcal{M} \;\triangleq\; \mathrm{diag}\{\,\mu_1 I_{2M},\; \mu_2 I_{2M},\; \ldots,\; \mu_N I_{2M}\,\} \tag{8.155}$$

$$\boldsymbol{\mathcal{H}}_{i-1} \;\triangleq\; \mathrm{diag}\,\{\,\boldsymbol{H}_{1,i-1},\; \boldsymbol{H}_{2,i-1},\; \ldots,\; \boldsymbol{H}_{N,i-1}\,\} \tag{8.156}$$

$$\boldsymbol{H}_{k,i-1} \;\triangleq\; \int_0^1 \nabla_w^2 J_k(w^\star - t\widetilde{\boldsymbol{\phi}}_{k,i-1})dt \tag{8.157}$$

where $\nabla_w^2 J_k(w)$ denotes the $2M \times 2M$ Hessian matrix of $J_k(w)$ relative to w. Moreover, the extended vectors $\{\widetilde{\boldsymbol{w}}_i^e, \boldsymbol{s}_i^e, b^e\}$ are defined by (8.143) and (8.144).

Example 8.12 (Mean-square-error costs). Let us re-consider the scenario studied in Example 8.1 and verify that result (8.152) collapses to (8.25). Indeed, in this case, we have $w^\star = w^o$ and the bias vector, b_k^e, will be zero for all agents $k = 1, 2, \ldots, N$. Moreover since the Hessian matrix is now block diagonal, we can easily verify from the definition (8.137) that

$$\boldsymbol{H}_{k,i-1} \;=\; \begin{bmatrix} R_{u,k} & 0 \\ 0 & R_{u,k}^{\mathsf{T}} \end{bmatrix} \tag{8.158}$$

Substituting these facts into the expressions in Lemma 8.1 we recover (8.25). ■

Example 8.13 (Simplifications in the real case)**.** The network error model of Lemma 8.1 can be simplified in the case of real data. This is because when $w \in \mathbb{R}^M$ is real-valued, we do not need to introduce the extended vectors (8.133) and (8.134) any longer. The simplifications that occur are described below.

To begin with, the distributed strategy (8.46) will be given by

$$\left\{\begin{array}{rcl} \boldsymbol{\phi}_{k,i-1} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{\psi}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\,\boldsymbol{\phi}_{\ell,i-1} \;-\; \mu_k\widehat{\nabla_{w^{\mathsf{T}}} J_k}\left(\boldsymbol{\phi}_{k,i-1}\right) \\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\boldsymbol{\psi}_{\ell,i} \end{array}\right. \tag{8.159}$$

where the gradient vector approximation in the second equation is now relative to w^{T} and not w^*. Subtracting the limit vector $w^\star$ directly from both sides of the above equations gives

$$\left\{\begin{array}{rcl} \widetilde{\boldsymbol{\phi}}_{k,i-1} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\widetilde{\boldsymbol{w}}_{\ell,i-1} \\ \widetilde{\boldsymbol{\psi}}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\widetilde{\boldsymbol{\phi}}_{\ell,i-1} \;+\; \mu_k\nabla_{w^{\mathsf{T}}} J_k(\boldsymbol{\phi}_{k,i-1}) \;+\; \mu_k\boldsymbol{s}_{k,i} \\ \widetilde{\boldsymbol{w}}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\widetilde{\boldsymbol{\psi}}_{\ell,i} \end{array}\right. \tag{8.160}$$

where now

$$\boldsymbol{s}_{k,i} \;\triangleq\; \widehat{\nabla_{w^{\mathsf{T}}} J_k}(\boldsymbol{\phi}_{k,i-1}) \;-\; \nabla_{w^{\mathsf{T}}} J_k(\boldsymbol{\phi}_{k,i-1}) \tag{8.161}$$

and the error vectors are measured relative to the same limit vector $w^\star$:

$$\widetilde{\boldsymbol{w}}_{k,i} = w^\star - \boldsymbol{w}_{k,i}, \quad \widetilde{\boldsymbol{\psi}}_{k,i} = w^\star - \boldsymbol{\psi}_{k,i}, \quad \widetilde{\boldsymbol{\phi}}_{k,i-1} = w^\star - \boldsymbol{\phi}_{k,i-1} \tag{8.162}$$

We then call upon the real-version of the mean-value theorem, namely, expression (D.9) in the appendix, to write

$$\begin{aligned} \nabla_{w^{\mathsf{T}}} J_k(\boldsymbol{\phi}_{k,i-1}) &= \underbrace{\nabla_{w^{\mathsf{T}}} J_k(w^\star)}_{\triangleq\, -b_k} \;-\; \underbrace{\left[\int_0^1 \nabla_w^2 J_k(w^\star - t\widetilde{\boldsymbol{\phi}}_{k,i-1})dt\right]}_{\triangleq\, \boldsymbol{H}_{k,i-1}}\widetilde{\boldsymbol{\phi}}_{k,i-1} \\ &= -b_k \;-\; \boldsymbol{H}_{k,i-1}\widetilde{\boldsymbol{\phi}}_{k,i-1} \end{aligned} \tag{8.163}$$

where we introduced the $M\times 1$ constant vector b_k and the (now) $M\times M$

stochastic matrix $\boldsymbol{H}_{k,i}$. Substituting (8.163) into (8.160) leads to

$$\left\{\begin{array}{rcl}\widetilde{\boldsymbol{\phi}}_{k,i-1} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\widetilde{\boldsymbol{w}}_{\ell,i-1}\\ \widetilde{\boldsymbol{\psi}}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\widetilde{\boldsymbol{\phi}}_{\ell,i-1} - \mu_k\boldsymbol{H}_{k,i-1}\widetilde{\boldsymbol{\phi}}_{k,i-1} - \mu_k b_k + \mu_k\boldsymbol{s}_{k,i}\\ \widetilde{\boldsymbol{w}}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\widetilde{\boldsymbol{\psi}}_{\ell,i}\end{array}\right. \tag{8.164}$$

so that the network error vector

$$\widetilde{\boldsymbol{w}}_i \;=\; \mathrm{col}\{\widetilde{\boldsymbol{w}}_{1,i},\widetilde{\boldsymbol{w}}_{2,i},\ldots,\widetilde{\boldsymbol{w}}_{N,i}\} \tag{8.165}$$

evolves according to the recursion

$$\widetilde{\boldsymbol{w}}_i \;=\; \boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1} \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i(\boldsymbol{w}_{i-1}) \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b,\;\; i\geq 0 \tag{8.166}$$

where now

$$\boldsymbol{\mathcal{B}}_{i-1} \;\triangleq\; \mathcal{A}_2^{\mathsf{T}}\left(\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\right)\mathcal{A}_1^{\mathsf{T}} \tag{8.167}$$

$$\mathcal{A}_o \;\triangleq\; A_o\otimes I_M,\quad \mathcal{A}_1 \;\triangleq\; A_1\otimes I_M,\quad \mathcal{A}_2 \;\triangleq\; A_2\otimes I_M \tag{8.168}$$

$$\mathcal{M} \;\triangleq\; \mathrm{diag}\{\,\mu_1 I_M,\,\mu_2 I_M,\,\ldots,\,\mu_N I_M\,\} \tag{8.169}$$

$$\boldsymbol{\mathcal{H}}_{i-1} \;\triangleq\; \mathrm{diag}\,\{\,\boldsymbol{H}_{1,i-1},\,\boldsymbol{H}_{2,i-1},\,\ldots,\,\boldsymbol{H}_{N,i-1}\,\} \tag{8.170}$$

$$\boldsymbol{H}_{k,i-1} \;\triangleq\; \int_0^1 \nabla_w^2 J_k(w^\star - t\widetilde{\boldsymbol{\phi}}_{k,i-1})dt \tag{8.171}$$

$$\boldsymbol{w}_{i-1} \;\triangleq\; \mathrm{col}\{\boldsymbol{w}_{1,i-1},\,\boldsymbol{w}_{2,i-1},\,\ldots,\,\boldsymbol{w}_{N,i-1}\} \tag{8.172}$$

and $\nabla_w^2 J_k(w)$ denotes the $M\times M$ Hessian matrix of $J_k(w)$ relative to w.

■

9

Stability of Multi-Agent Networks

Building on the results from the previous chapter, we are now ready to examine the stability of the mean-error process, $\mathbb{E}\,\widetilde{\boldsymbol{w}}_i$, the mean-square-error, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, and the fourth-order moment, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^4$, by using the network error recursion (8.152). The key results proven in the current chapter are that for sufficiently small step-sizes, and for each agent k, it will hold that

$$\limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_{k,i}\| = O(\mu_{\max}) \tag{9.1}$$

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 = O(\mu_{\max}) \tag{9.2}$$

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^4 = O(\mu_{\max}^2) \tag{9.3}$$

where $\mu_{\max}$ is an upper bound on the largest step-size parameter across the network since, from (8.52), we parameterized all step-sizes as scaled multiples of $\mu_{\max}$, namely,

$$\mu_k \triangleq \tau_k\,\mu_{\max}, \quad k = 1, 2, \ldots, N \tag{9.4}$$

where $0 < \tau_k \leq 1$. The error vectors, $\{\widetilde{\boldsymbol{w}}_{k,i}\}$, in the above expressions are measured relative to the limit vector, $w^\star$:

$$\widetilde{\boldsymbol{w}}_{k,i} = w^\star - \boldsymbol{w}_{k,i} \tag{9.5}$$

where $w^\star$ was defined by (8.55) as the unique minimum of the weighted aggregate cost function, $J^{\text{glob},\star}(w)$, from (8.53), namely,

$$J^{\text{glob},\star}(w) \;\triangleq\; \sum_{k=1}^{N} q_k J_k(w) \tag{9.6}$$

and the $\{q_k\}$ are positive scalars corresponding to the entries of the vector:

$$q \;\triangleq\; \text{diag}\{\mu_1, \mu_2, \ldots, \mu_N\} A_2 p \tag{9.7}$$

Here, the vector p refers to the Perron eigenvector of the matrix product

$$P \;\triangleq\; A_1 A_o A_2 \tag{9.8}$$

and is defined through the relations:

$$Pp = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0 \tag{9.9}$$

For ease of reference, we recall the definition of the original aggregate cost function (8.44), namely,

$$J^{\text{glob}}(w) \;\triangleq\; \sum_{k=1}^{N} J_k(w) \tag{9.10}$$

9.1 Stability of Second-Order Error Moment

The first result establishes the mean-square stability of the network error process and shows that its mean-square value tends asymptotically to a bounded region in the order of $O(\mu_{\max})$.

Theorem 9.1 (Network mean-square-error stability)**.** Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P = A_1 A_o A_2$. Assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumption 6.1. Assume further that the first and second-order moments of the gradient noise process satisfy the conditions in Assumption 8.1. Then, the network is mean-square stable for sufficiently small step-sizes, namely, it holds that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \;=\; O(\mu_{\max}), \quad k = 1, 2, \ldots, N \tag{9.11}$$

for any $\mu_{\max} < \mu_o$, for some small enough μ_o.

Proof. The derivation is demanding. We follow arguments motivated by the analysis in [70, 277] and they involve, as an initial step, transforming the error recursion (9.12) shown below into a more convenient form shown later in (9.60). We establish the result for the general case of complex data and, therefore, $h = 2$ throughout this derivation.

We start from the network error recursion (8.152):

$$\widetilde{\boldsymbol{w}}_i^e \;=\; \boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^e \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e, \;\; i \geq 0 \tag{9.12}$$

where

$$\begin{aligned}
\boldsymbol{\mathcal{B}}_{i-1} &= \mathcal{A}_2^{\mathsf{T}}\left(\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\right)\mathcal{A}_1^{\mathsf{T}} \\
&= \mathcal{A}_2^{\mathsf{T}}\mathcal{A}_o^{\mathsf{T}}\mathcal{A}_1^{\mathsf{T}} \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{A}_1^{\mathsf{T}} \\
&\triangleq \mathcal{P}^{\mathsf{T}} \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{A}_1^{\mathsf{T}}
\end{aligned} \tag{9.13}$$

in terms of the matrix

$$\begin{aligned}
\mathcal{P}^{\mathsf{T}} &\triangleq \mathcal{A}_2^{\mathsf{T}}\mathcal{A}_o^{\mathsf{T}}\mathcal{A}_1^{\mathsf{T}} \\
&= (A_2^{\mathsf{T}} \otimes I_{2M})(A_o^{\mathsf{T}} \otimes I_{2M})(A_1^{\mathsf{T}} \otimes I_{2M}) \\
&= (A_2^{\mathsf{T}}A_o^{\mathsf{T}}A_1^{\mathsf{T}} \otimes I_{2M}) \\
&= P^{\mathsf{T}} \otimes I_{2M}
\end{aligned} \tag{9.14}$$

The matrix $P = A_1A_oA_2$ is left-stochastic and assumed primitive. It follows that it has a single eigenvalue at one while all other eigenvalues are strictly inside the unit circle. We let p denote its Perron eigenvector, which is already defined by (9.9). This vector determines the entries of q defined by (9.7). Note, for later reference, that the $k-$entry of q can be extracted by computing the inner product of q with the $k-$th basis vector, e_k, which has a unit entry at the $k-$th location and zeros elsewhere, i.e.,

$$\begin{aligned}
q_k &= \mu_k(e_k^{\mathsf{T}}A_2p) \\
&\overset{(9.4)}{=} \mu_{\max}\,\tau_k(e_k^{\mathsf{T}}A_2p)
\end{aligned} \tag{9.15}$$

Obviously, it holds for the extended matrices $\{\mathcal{P}, \mathcal{A}_2\}$ that

$$\mathcal{P}(p \otimes I_{2M}) = (p \otimes I_{2M}) \tag{9.16}$$

$$\mathcal{M}\mathcal{A}_2(p \otimes I_{2M}) = (q \otimes I_{2M}) \tag{9.17}$$

$$(\mathbb{1}^{\mathsf{T}} \otimes I_{2M})(p \otimes I_{2M}) = I_{2M} \tag{9.18}$$

Moreover, since A_1 and A_2 are left-stochastic, it holds that

$$\mathcal{A}_1^{\mathsf{T}}(\mathbb{1} \otimes I_{2M}) = (\mathbb{1} \otimes I_{2M}) \tag{9.19}$$

$$\mathcal{A}_2^{\mathsf{T}}(\mathbb{1} \otimes I_{2M}) = (\mathbb{1} \otimes I_{2M}) \tag{9.20}$$

The derivation that follows exploits the eigen-structure of P. We start by noting that the $N \times N$ matrix P admits a Jordan canonical decomposition of the form [113, p.128]:

$$P \triangleq V_\epsilon J V_\epsilon^{-1} \tag{9.21}$$

$$J = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & J_\epsilon \end{array}\right] \tag{9.22}$$

$$V_\epsilon = \left[\begin{array}{c|c} p & V_R \end{array}\right] \tag{9.23}$$

$$V_\epsilon^{-1} = \left[\begin{array}{c} \mathbb{1}^{\mathsf{T}} \\ \hline \\ V_L^{\mathsf{T}} \end{array}\right] \tag{9.24}$$

where the matrix J_ϵ consists of Jordan blocks, with each one of them having the generic form (say, for a Jordan block of size 4×4):

$$\left[\begin{array}{cccc} \lambda & & & \\ \epsilon & \lambda & & \\ & \epsilon & \lambda & \\ & & \epsilon & \lambda \end{array}\right] \tag{9.25}$$

with $\epsilon > 0$ appearing on the lower[1] diagonal, and where the eigenvalue λ may be complex but has magnitude strictly less than one. The scalar ϵ is any small positive number that is independent of $\mu_{\max}$. Obviously, since $V_\epsilon^{-1} V_\epsilon = I_N$, it holds that

$$\mathbb{1}^{\mathsf{T}} V_R = 0 \tag{9.26}$$

$$V_L^{\mathsf{T}} p = 0 \tag{9.27}$$

$$V_L^{\mathsf{T}} V_R = I_{N-1} \tag{9.28}$$

The matrices $\{V_\epsilon, J, V_\epsilon^{-1}\}$ have dimensions $N \times N$ while the matrices $\{V_L, J_\epsilon, V_R\}$ have dimensions $(N-1) \times (N-1)$. The Jordan decomposition of the extended matrix $\mathcal{P} = P \otimes I_{2M}$ is given by

$$\mathcal{P} = (V_\epsilon \otimes I_{2M})(J \otimes I_{2M})(V_\epsilon^{-1} \otimes I_{2M}) \tag{9.29}$$

so that substituting into (9.13) we obtain

$$\boldsymbol{\mathcal{B}}_{i-1} = \big((V_\epsilon^{-1})^{\mathsf{T}} \otimes I_{2M}\big) \Big\{ (J^{\mathsf{T}} \otimes I_{2M}) - \boldsymbol{\mathcal{D}}_{i-1}^{\mathsf{T}} \Big\} \big(V_\epsilon^{\mathsf{T}} \otimes I_{2M}\big) \tag{9.30}$$

[1] For any $N \times N$ matrix A, the traditional Jordan decomposition $A = TJ'T^{-1}$ involves Jordan blocks in J' that have ones on the lower diagonal instead of ϵ. However, if we introduce the diagonal matrix $E = \text{diag}\{1, \epsilon, \epsilon^2, \ldots, \epsilon^{N-1}\}$, then $A = TE^{-1}EJ'E^{-1}ET^{-1}$, which we rewrite as $A = V_\epsilon J V_\epsilon^{-1}$ with $V_\epsilon = TE^{-1}$ and $J = EJ'E^{-1}$. The matrix J now has ϵ values instead of ones on the lower diagonal.

where

$$\begin{aligned}\boldsymbol{\mathcal{D}}_{i-1}^{\mathsf{T}} &\triangleq \left(V_\epsilon^{\mathsf{T}}\otimes I_{2M}\right)\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{A}_1^{\mathsf{T}}\left((V_\epsilon^{-1})^{\mathsf{T}}\otimes I_{2M}\right)\\ &\equiv \begin{bmatrix}\boldsymbol{D}_{11,i-1}^{\mathsf{T}} & \boldsymbol{D}_{21,i-1}^{\mathsf{T}}\\ \boldsymbol{D}_{12,i-1}^{\mathsf{T}} & \boldsymbol{D}_{22,i-1}^{\mathsf{T}}\end{bmatrix}\end{aligned} \tag{9.31}$$

Using the partitioning (9.23)–(9.24) and the fact that

$$\mathcal{A}_1 \;=\; A_1\otimes I_{2M}, \qquad \mathcal{A}_2 \;=\; A_2\otimes I_{2M} \tag{9.32}$$

we find that the block entries $\{\boldsymbol{D}_{mn,i-1}\}$ in (9.31) are given by

$$\boldsymbol{D}_{11,i-1} \;=\; \sum_{k=1}^{N} q_k\boldsymbol{H}_{k,i-1}^{\mathsf{T}} \tag{9.33}$$

$$\boldsymbol{D}_{12,i-1} \;=\; (\mathbb{1}^{\mathsf{T}}\otimes I_{2M})\boldsymbol{\mathcal{H}}_{i-1}^{\mathsf{T}}\mathcal{M}(A_2V_R\otimes I_{2M}) \tag{9.34}$$

$$\boldsymbol{D}_{21,i-1} \;=\; (V_L^{\mathsf{T}}A_1\otimes I_{2M})\boldsymbol{\mathcal{H}}_{i-1}^{\mathsf{T}}(q\otimes I_{2M}) \tag{9.35}$$

$$\boldsymbol{D}_{22,i-1} \;=\; (V_L^{\mathsf{T}}A_1\otimes I_{2M})\boldsymbol{\mathcal{H}}_{i-1}^{\mathsf{T}}\mathcal{M}(A_2V_R\otimes I_{2M}) \tag{9.36}$$

Let us now show that the entries in each of these matrices is in the order of $O(\mu_{\max})$, as well as verify that the matrix norm sequences of these matrices are uniformly bounded from above for all i. To begin with, recall from (8.157) that

$$\boldsymbol{H}_{k,i-1} \;\triangleq\; \int_0^1 \nabla_w^2 J_k(w^\star - t\widetilde{\boldsymbol{\phi}}_{k,i-1})dt \tag{9.37}$$

and, moreover, by assumption, all individual costs $J_k(w)$ are convex functions with at least one of them, say, the cost function of index k_o, being ν_d-strongly-convex. This fact implies that, for any w,

$$\nabla_w^2 J_{k_o}(w) \geq \frac{\nu_d}{h}I_{hM} > 0, \qquad \nabla_w^2 J_k(w)\geq 0, \qquad k\neq k_o \tag{9.38}$$

Consequently,

$$\boldsymbol{H}_{k_o,i-1} \geq \frac{\nu_d}{h}I_{hM} > 0, \qquad \boldsymbol{H}_{k,i-1}\geq 0, \qquad k\neq k_o \tag{9.39}$$

and, therefore, $\boldsymbol{D}_{11,i-1} > 0$. More specifically, the matrix sequence $\boldsymbol{D}_{11,i-1}$ is uniformly bounded from below as follows:

$$\begin{aligned}\boldsymbol{D}_{11,i-1} &\geq q_{k_o}\frac{\nu_d}{h}I_{hM}\\ &\overset{(9.15)}{=} \mu_{\max}\,\tau_{k_o}(e_{k_o}^{\mathsf{T}}A_2p)\frac{\nu_d}{h}I_{hM}\\ &= O(\mu_{\max})\end{aligned} \tag{9.40}$$

On the other hand, from the upper bound on the sum of the Hessian matrices in (6.13), and since each individual Hessian matrix is at least non-negative definite, we get

$$\boldsymbol{H}_{k,i-1} \;\leq\; \frac{\delta_d}{h} I_{hM} \tag{9.41}$$

so that the matrix sequence $\boldsymbol{D}_{11,i-1}$ is uniformly bounded from above as well:

$$\begin{aligned} \boldsymbol{D}_{11,i-1} &\leq q_{\max} N \frac{\delta_d}{h} I_{hM} \\ &\overset{(9.15)}{=} \mu_{\max}\, \tau_{k_{\max}} (e_{k_{\max}}^{\mathsf{T}} A_2 p) N \frac{\delta_d}{h} I_{hM} \\ &= O(\mu_{\max}) \end{aligned} \tag{9.42}$$

where $k_{\max}$ denotes the $k-$index of the largest q_k entry. Combining results (9.40)–(9.42) we conclude that

$$\boldsymbol{D}_{11,i-1} \;=\; O(\mu_{\max}) \tag{9.43}$$

Actually, since $\boldsymbol{D}_{11,i-1}$ is Hermitian positive-definite, we also conclude that its eigenvalues (which are positive and real) are $O(\mu_{\max})$. This is because from the relation

$$\mu_{\max}\, \tau_{k_o} (e_{k_o}^{\mathsf{T}} A_2 p) \frac{\nu_d}{h} I_{hM} \;\leq\; \boldsymbol{D}_{11,i-1} \;\leq\; \mu_{\max}\, \tau_{k_{\max}} (e_{k_{\max}}^{\mathsf{T}} A_2 p) N \frac{\delta_d}{h} I_{hM} \tag{9.44}$$

we can write, more compactly,

$$c_1 \mu_{\max} I_{hM} \;\leq\; \boldsymbol{D}_{11,i-1} \;\leq\; c_2 \mu_{\max} I_{hM} \tag{9.45}$$

for some positive constants c_1 and c_2 that are independent of $\mu_{\max}$ and i. Accordingly, for the eigenvalues of $\boldsymbol{D}_{11,i-1}$, we can write

$$c_1 \mu_{\max} \leq \lambda(\boldsymbol{D}_{11,i-1}) \leq c_2 \mu_{\max} \tag{9.46}$$

It follows that the eigenvalues of $I_{2M} - \boldsymbol{D}_{11,i-1}^{\mathsf{T}}$ are $1 - O(\mu_{\max})$ so that, in terms of the $2-$induced norm and for sufficiently small $\mu_{\max}$:

$$\begin{aligned} \|I_{2M} - \boldsymbol{D}_{11,i-1}^{\mathsf{T}}\| &= \rho(I_{2M} - \boldsymbol{D}_{11,i-1}^{\mathsf{T}}) \\ &\leq 1 - \sigma_{11} \mu_{\max} \\ &= 1 - O(\mu_{\max}) \end{aligned} \tag{9.47}$$

for some positive constant σ_{11} that is independent of $\mu_{\max}$ and i.

Similarly, from (9.39) and (9.41), and since each $\boldsymbol{H}_{k,i-1}$ is bounded from below and from above, we can conclude that

$$\boldsymbol{D}_{12,i-1} \;=\; O(\mu_{\max}),\;\; \boldsymbol{D}_{21,i-1} \;=\; O(\mu_{\max}),\;\; \boldsymbol{D}_{22,i-1} \;=\; O(\mu_{\max}) \tag{9.48}$$

and that the norms of these matrix sequences are also uniformly bounded from above. For example, using the 2−induced norm (i.e., maximum singular value):

$$\begin{aligned}
\|\boldsymbol{D}_{21,i-1}\| &\leq \|V_L^{\mathsf{T}} A_1 \otimes I_{2M}\|\,\|q\otimes I_{2M}\|\,\|\boldsymbol{\mathcal{H}}_{i-1}^{\mathsf{T}}\| \\
&\leq \|V_L^{\mathsf{T}} A_1 \otimes I_{2M}\|\,\|q\otimes I_{2M}\|\,\left(\max_{1\leq k\leq N}\|\boldsymbol{H}_{k,i-1}\|\right) \\
&\overset{(9.41)}{\leq} \|V_L^{\mathsf{T}} A_1 \otimes I_{2M}\|\,\|q\otimes I_{2M}\|\,\left(\frac{\delta_d}{h}\right) \\
&= \|V_L^{\mathsf{T}} A_1 \otimes I_{2M}\|\,\|q\|\,\left(\frac{\delta_d}{h}\right) \\
&\leq \|V_L^{\mathsf{T}} A_1 \otimes I_{2M}\|\,\sqrt{N\,q_{\max}^2}\,\left(\frac{\delta_d}{h}\right) \\
&= \|V_L^{\mathsf{T}} A_1 \otimes I_{2M}\|\,\sqrt{N}\,\mu_{\max}\,\tau_{k_{\max}}(e_{k_{\max}}^{\mathsf{T}} A_2 p)\,\left(\frac{\delta_d}{h}\right)
\end{aligned} \tag{9.49}$$

so that

$$\|\boldsymbol{D}_{21,i-1}\| \;\leq\; \sigma_{21}\mu_{\max} = O(\mu_{\max}) \tag{9.50}$$

for some positive constant σ_{21}. In the above derivation we used the fact that $\|q\otimes I_{2M}\| = \|q\|$ since, from Table F.1 in the appendix, the singular values of a Kronecker product are given by all possible products of the singular values of the individual matrices. A similar argument applies to $\boldsymbol{D}_{12,i-1}$ and $\boldsymbol{D}_{22,i-1}$ for which we can verify that

$$\|\boldsymbol{D}_{12,i-1}\| \leq \sigma_{12}\mu_{\max} = O(\mu_{\max}), \qquad \|\boldsymbol{D}_{22,i-1}\| \leq \sigma_{22}\mu_{\max} = O(\mu_{\max}) \tag{9.51}$$

for some positive constants σ_{21} and σ_{22}. Let

$$\mathcal{V}_\epsilon \;\triangleq\; V_\epsilon \otimes I_{2M}, \qquad \mathcal{J}_\epsilon \;\triangleq\; J_\epsilon \otimes I_{2M} \tag{9.52}$$

Then, using (9.30), we can write

$$\boldsymbol{\mathcal{B}}_{i-1} = \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} \begin{bmatrix} I_{2M} - \boldsymbol{D}_{11,i-1}^{\mathsf{T}} & -\boldsymbol{D}_{21,i-1}^{\mathsf{T}} \\ -\boldsymbol{D}_{12,i-1}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - \boldsymbol{D}_{22,i-1}^{\mathsf{T}} \end{bmatrix} \mathcal{V}_\epsilon^{\mathsf{T}} \tag{9.53}$$

To simplify the notation, we drop the argument $\boldsymbol{w}_{i-1}^e$ in (9.12) and write $\boldsymbol{s}_i^e$ instead of $\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)$ from this point onwards. We now multiply both sides of the error recursion (9.12) from the left by $\mathcal{V}_\epsilon^{\mathsf{T}}$:

$$\mathcal{V}_\epsilon^{\mathsf{T}}\widetilde{\boldsymbol{w}}_i^e \;=\; \mathcal{V}_\epsilon^{\mathsf{T}}\boldsymbol{\mathcal{B}}_{i-1}\left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}}\mathcal{V}_\epsilon^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1}^e \;+\; \mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e \;-\; \mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e, \;\; i\geq 0 \tag{9.54}$$

and let

$$\mathcal{V}_\epsilon^{\mathsf{T}}\widetilde{\boldsymbol{w}}_i^e \;=\; \begin{bmatrix} (p^{\mathsf{T}}\otimes I_{2M})\widetilde{\boldsymbol{w}}_i^e \\ (V_R^{\mathsf{T}}\otimes I_{2M})\widetilde{\boldsymbol{w}}_i^e \end{bmatrix} \;\triangleq\; \begin{bmatrix} \bar{\boldsymbol{w}}_i^e \\ \check{\boldsymbol{w}}_i^e \end{bmatrix} \tag{9.55}$$

$$\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e \;=\; \begin{bmatrix} (p^{\mathsf{T}}\otimes I_{2M})\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e \\ (V_R^{\mathsf{T}}\otimes I_{2M})\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e \end{bmatrix} \;\triangleq\; \begin{bmatrix} \bar{\boldsymbol{s}}_i^e \\ \check{\boldsymbol{s}}_i^e \end{bmatrix} \tag{9.56}$$

$$\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \;=\; \begin{bmatrix} (p^{\mathsf{T}}\otimes I_{2M})\mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \\ (V_R^{\mathsf{T}}\otimes I_{2M})\mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \end{bmatrix} \;\triangleq\; \begin{bmatrix} 0 \\ \check{b}^e \end{bmatrix} \tag{9.57}$$

where the zero entry in the last equality is due to the fact that

$$\begin{aligned}
(p^{\mathsf{T}}\otimes I_{2M})\mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e &= (q^{\mathsf{T}}\otimes I_{2M})b^e \\
&= \sum_{k=1}^{N} q_k b_k^e \\
&= -\sum_{k=1}^{N} q_k \begin{bmatrix} \nabla_{w^*}J_k(w^\star) \\ \nabla_{w^{\mathsf{T}}}J_k(w^\star) \end{bmatrix} \\
&= -\sum_{k=1}^{N} q_k \begin{bmatrix} [\nabla_w J_k(w^\star)]^* \\ [\nabla_w J_k(w^\star)]^{\mathsf{T}} \end{bmatrix} \\
&\overset{(8.55)}{=} 0
\end{aligned} \tag{9.58}$$

Moreover, from the expression for $\check{b}^e$ in (9.57), we note that it depends on $\mathcal{M}$ and b^e. Recall from (8.110) and (8.144) that the entries of b^e are defined in terms of the gradient vectors $\nabla_{w^*}J_k(w^\star)$. Since each $J_k(w)$ is twice-differentiable from Assumption 6.1, then each gradient vector of $J_k(w)$ is a differentiable function and therefore bounded. It follows that b^e has bounded norm and we conclude that

$$\check{b}^e \;=\; O(\mu_{\max}) \tag{9.59}$$

Using the just introduced transformed variables, we can rewrite (9.54) in the form

$$\begin{bmatrix} \bar{\boldsymbol{w}}_i^e \\ \check{\boldsymbol{w}}_i^e \end{bmatrix} = \begin{bmatrix} I_{2M} - \boldsymbol{D}_{11,i-1}^{\mathsf{T}} & -\boldsymbol{D}_{21,i-1}^{\mathsf{T}} \\ -\boldsymbol{D}_{12,i-1}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - \boldsymbol{D}_{22,i-1}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{w}}_{i-1}^e \\ \check{\boldsymbol{w}}_{i-1}^e \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{s}}_i^e \\ \check{\boldsymbol{s}}_i^e \end{bmatrix} - \begin{bmatrix} 0 \\ \check{b}^e \end{bmatrix} \tag{9.60}$$

or, in expanded form,

$$\bar{\boldsymbol{w}}_i^e \;=\; (I_{2M} - \boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e \;-\; \boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e \;+\; \bar{\boldsymbol{s}}_i^e \tag{9.61}$$

$$\check{\boldsymbol{w}}_i^e \;=\; (\mathcal{J}_\epsilon^{\mathsf{T}} - \boldsymbol{D}_{22,i-1}^{\mathsf{T}})\check{\boldsymbol{w}}_{i-1}^e \;-\; \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e \;+\; \check{\boldsymbol{s}}_i^e \;-\; \check{b}^e \tag{9.62}$$

Conditioning both sides on $\boldsymbol{\mathcal{F}}_{i-1}$, computing the conditional second-order moments, and using the conditions from Assumption 8.1 on the gradient noise process we get

$$\mathbb{E}\left[\,\|\bar{\boldsymbol{w}}_i^e\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\right] = \|(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e - \boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\|^2 + \mathbb{E}\left[\,\|\bar{\boldsymbol{s}}_i^e\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\right] \tag{9.63}$$

and

$$\begin{aligned}\mathbb{E}\left[\,\|\check{\boldsymbol{w}}_i^e\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\right] \;=\; &\|(\mathcal{J}_\epsilon^{\mathsf{T}}-\boldsymbol{D}_{22,i-1}^{\mathsf{T}})\check{\boldsymbol{w}}_{i-1}^e - \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e - \check{b}^e\|^2 \;+\\ &\mathbb{E}\left[\,\|\check{\boldsymbol{s}}_i^e\|^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\,\right]\end{aligned} \tag{9.64}$$

Computing the expectations again we conclude that

$$\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^2 = \mathbb{E}\,\|(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e - \boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\|^2 + \mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2 \tag{9.65}$$

and

$$\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2 = \mathbb{E}\,\|(\mathcal{J}_\epsilon^{\mathsf{T}}-\boldsymbol{D}_{22,i-1}^{\mathsf{T}})\check{\boldsymbol{w}}_{i-1}^e - \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e - \check{b}^e\|^2 + \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2 \tag{9.66}$$

Continuing with the first variance (9.65), we can appeal to Jensen's inequality (F.26) from the appendix and apply it to the function $f(x)=\|x\|^2$ to bound the variance as follows:

$$\begin{aligned}&\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^2\\ &= \mathbb{E}\,\left\|(1-t)\frac{1}{1-t}(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e - t\frac{1}{t}\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\right\|^2 + \mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2\\ &\le (1-t)\mathbb{E}\,\left\|\frac{1}{1-t}(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e\right\|^2 + t\,\mathbb{E}\,\left\|\frac{1}{t}\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\right\|^2 + \mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2\\ &\le \frac{1}{1-t}\mathbb{E}\,\left[\|I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}}\|^2\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2\right] + \frac{1}{t}\mathbb{E}\,\left[\|\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\|^2\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\right] + \mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2\\ &\le \frac{(1-\sigma_{11}\mu_{\max})^2}{1-t}\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \frac{\sigma_{21}^2\mu_{\max}^2}{t}\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2\end{aligned} \tag{9.67}$$

for any arbitrary positive number $t\in(0,1)$. We select

$$t = \sigma_{11}\mu_{\max} \tag{9.68}$$

Then, the last inequality can be written as

$$\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^2 \;\le\; (1-\sigma_{11}\mu_{\max})\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \left(\frac{\sigma_{21}^2\mu_{\max}}{\sigma_{11}}\right)\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2 \tag{9.69}$$

We now repeat a similar argument for the second variance relation (9.66). Thus, using Jensen's inequality again we have

$$\begin{aligned}
&\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2 = \qquad (9.70)\\
&= \mathbb{E}\,\left\|\mathcal{J}_\epsilon^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e - \left[\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e + \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e + \check{b}^e\right]\right\|^2 + \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2\\
&= \mathbb{E}\,\left\|t\frac{1}{t}\mathcal{J}_\epsilon^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e - (1-t)\frac{1}{1-t}\left[\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e + \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e + \check{b}^e\right]\right\|^2 + \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2\\
&\le \frac{1}{t}\mathbb{E}\,\left\|\mathcal{J}_\epsilon^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\right\|^2 + \frac{1}{1-t}\mathbb{E}\,\left\|\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e + \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e + \check{b}^e\right\|^2 + \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2
\end{aligned}$$

for any arbitrary positive number $t \in (0,1)$. Now note that

$$\begin{aligned}
\left\|\mathcal{J}_\epsilon^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\right\|^2 &= \left(\check{\boldsymbol{w}}_{i-1}^e\right)^*\left(\mathcal{J}_\epsilon^{\mathsf{T}}\right)^*\mathcal{J}_\epsilon^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\\
&= \left(\check{\boldsymbol{w}}_{i-1}^e\right)^*\left(\mathcal{J}_\epsilon\mathcal{J}_\epsilon^*\right)^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\\
&\le \rho\left(\mathcal{J}_\epsilon\mathcal{J}_\epsilon^*\right)\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2 \qquad (9.71)
\end{aligned}$$

where we called upon the Rayleigh-Ritz characterization of the eigenvalues of Hermitian matrices [104, 113], namely,

$$\lambda_{\min}(C)\,\|x\|^2 \;\le\; x^*Cx \;\le\; \lambda_{\max}(C)\,\|x\|^2 \qquad (9.72)$$

for any Hermitian matrix C. Applying this result to the Hermitian and non-negative definite matrix $C = (\mathcal{J}_\epsilon\mathcal{J}_\epsilon^*)^{\mathsf{T}}$, and noting that $\rho(C) = \rho(C^{\mathsf{T}})$, we obtain (9.71). From definition (9.52) for $\mathcal{J}_\epsilon$ we further get

$$\begin{aligned}
\rho\left(\mathcal{J}_\epsilon\mathcal{J}_\epsilon^*\right) &= \rho\left[(J_\epsilon\otimes I_{2M})(J_\epsilon^*\otimes I_{2M})\right]\\
&= \rho\left[(J_\epsilon J_\epsilon^*\otimes I_{2M})\right]\\
&= \rho(J_\epsilon J_\epsilon^*) \qquad (9.73)
\end{aligned}$$

The matrix J_ϵ is block diagonal and consists of Jordan blocks. Assume initially that it consists of a single Jordan block, say, of size 4×4, for illustration purposes. Then, we can write:

$$\begin{aligned}
J_\epsilon J_\epsilon^* &= \begin{bmatrix}\lambda & & & \\ \epsilon & \lambda & & \\ & \epsilon & \lambda & \\ & & \epsilon & \lambda\end{bmatrix}\begin{bmatrix}\lambda^* & \epsilon & & \\ & \lambda^* & \epsilon & \\ & & \lambda^* & \epsilon \\ & & & \lambda^*\end{bmatrix}\\
&= \begin{bmatrix}|\lambda|^2 & \epsilon\lambda & & \\ \epsilon\lambda^* & |\lambda|^2+\epsilon^2 & \epsilon\lambda & \\ & \epsilon\lambda^* & |\lambda|^2+\epsilon^2 & \epsilon\lambda \\ & & \epsilon\lambda^* & |\lambda|^2+\epsilon^2\end{bmatrix} \qquad (9.74)
\end{aligned}$$

Using the property that the spectral radius of a matrix is bounded by any of its norms, and using the $1-$norm (maximum absolute column sum), we get for the above example

$$\begin{aligned}\rho(J_\epsilon J_\epsilon^*) &\leq \|J_\epsilon J_\epsilon^*\|_1 \\ &= |\lambda|^2+\epsilon^2+\epsilon|\lambda^*|+\epsilon|\lambda| \\ &= (|\lambda|+\epsilon)^2 \end{aligned} \tag{9.75}$$

If J_ϵ consists of multiple Jordan blocks, say, L of them with eigenvalue λ_ℓ each, then

$$\rho(J_\epsilon J_\epsilon^*) \leq \max_{1\leq \ell\leq L} (|\lambda_\ell|+\epsilon)^2 = (\rho(J_\epsilon)+\epsilon)^2 \tag{9.76}$$

where $\rho(J_\epsilon)$ does not depend on ϵ and is equal to the second largest eigenvalue in magnitude in J, which we know is strictly less than one in magnitude. Substituting this conclusion into (9.70) gives

$$\begin{aligned}\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2 \leq\ & \frac{1}{t}(\rho(J_\epsilon)+\epsilon)^2\,\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2 + \\ & \frac{1}{1-t}\mathbb{E}\,\left\|\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e+\boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e+\check{b}^e\right\|^2+\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2\end{aligned} \tag{9.77}$$

Since we know that $\rho(J_\epsilon)\in(0,1)$, then we can select ϵ small enough to ensure $\rho(J_\epsilon)+\epsilon\in(0,1)$. We then select

$$t=\rho(J_\epsilon)+\epsilon \tag{9.78}$$

and rewrite (9.77) as

$$\begin{aligned}\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2 \leq\ & (\rho(J_\epsilon)+\epsilon)\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2+\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2+ \\ & \left(\frac{1}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\left\|\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e+\boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e+\check{b}^e\right\|^2\end{aligned} \tag{9.79}$$

We can bound the last term on the right-hand side of the above expression as follows:

$$\mathbb{E}\,\left\|\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e+\boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e+\check{b}^e\right\|^2 = \tag{9.80}$$

$$\begin{aligned}&= \mathbb{E}\,\left\|\frac{1}{3}3\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e+\frac{1}{3}3\boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e+\frac{1}{3}3\check{b}^e\right\|^2 \\ &\leq \frac{1}{3}\mathbb{E}\,\left\|3\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\right\|^2+\frac{1}{3}\mathbb{E}\,\left\|3\boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e\right\|^2+\frac{1}{3}\|3\check{b}^e\|^2 \\ &\leq 3\mathbb{E}\,\left\|\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\right\|^2+3\mathbb{E}\,\left\|\boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e\right\|^2+3\|\check{b}^e\|^2 \\ &\leq 3\sigma_{22}^2\mu_{\max}^2\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2+3\sigma_{12}^2\mu_{\max}^2\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^2+3\|\check{b}^e\|^2\end{aligned}$$

Substituting into (9.79) we obtain

$$\begin{aligned}\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2 \;\leq\; & \left(\rho(J_\epsilon)+\epsilon+\frac{3\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2 + \\ & \left(\frac{3\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^2 + \\ & \left(\frac{3}{1-\rho(J_\epsilon)-\epsilon}\right)\|\check{b}^e\|^2+\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2 \end{aligned} \tag{9.81}$$

We now bound the noise terms, $\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2$ in (9.69) and $\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2$ in (9.81). For that purpose, we first note that

$$\begin{aligned}\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2+\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2 &= \mathbb{E}\,\left\|\begin{bmatrix}\bar{\boldsymbol{s}}_i^e\\ \check{\boldsymbol{s}}_i^e\end{bmatrix}\right\|^2 \\ &= \mathbb{E}\,\left\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e\right\|^2 \\ &\leq \left\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\right\|^2\,\|\mathcal{M}\|^2\,\mathbb{E}\,\|\boldsymbol{s}_i^e\|^2 \\ &\leq v_1^2\mu_{\max}^2\mathbb{E}\,\|\boldsymbol{s}_i^e\|^2 \end{aligned} \tag{9.82}$$

where the positive constant v_1 is independent of $\mu_{\max}$ and is equal to the following norm

$$v_1 \;\triangleq\; \left\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\right\| \tag{9.83}$$

On the other hand, using (8.113)–(8.114), we have

$$\begin{aligned}\mathbb{E}\,\|\boldsymbol{s}_i^e\|^2 &= \sum_{k=1}^{N}\mathbb{E}\,\|\boldsymbol{s}_{k,i}^e\|^2 \\ &= 2\left(\sum_{k=1}^{N}\mathbb{E}\,\|\boldsymbol{s}_{k,i}\|^2\right) \end{aligned} \tag{9.84}$$

in terms of the variances of the individual gradient noise processes, $\mathbb{E}\,\|\boldsymbol{s}_{k,i}\|^2$, and where we used the fact that

$$\|\boldsymbol{s}_{k,i}^e\|^2 = 2\|\boldsymbol{s}_{k,i}\|^2 \tag{9.85}$$

Now, for each term $\boldsymbol{s}_{k,i}$ we have

$$\begin{aligned}
\mathbb{E}\,\|\boldsymbol{s}_{k,i}\|^2 &\overset{(8.118)}{\leq} (\beta_k^2/h^2)\mathbb{E}\,\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^2 \;+\; \sigma_{s,k}^2 \\
&= (\beta_k^2/h^2)\mathbb{E}\,\Big\|\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\,\widetilde{\boldsymbol{w}}_{\ell,i-1}\Big\|^2 \;+\; \sigma_{s,k}^2 \\
&\overset{(F.26)}{\leq} (\beta_k^2/h^2)\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^2 \;+\; \sigma_{s,k}^2 \\
&\leq (\beta_k^2/h^2)\sum_{\ell=1}^{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^2 \;+\; \sigma_{s,k}^2 \\
&= (\beta_k^2/2h^2)\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \sigma_{s,k}^2 \\
&= (\beta_k^2/2h^2)\mathbb{E}\,\Big\|\big(\mathcal{V}_\epsilon^{-1}\big)^{\mathsf{T}}\mathcal{V}_\epsilon^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1}^e\Big\|^2 \;+\; \sigma_{s,k}^2 \\
&\leq (\beta_k^2/2h^2)\,\Big\|\big(\mathcal{V}_\epsilon^{-1}\big)^{\mathsf{T}}\Big\|^2\,\mathbb{E}\,\|\mathcal{V}_\epsilon^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \sigma_{s,k}^2 \\
&\overset{(9.55)}{=} (\beta_k^2/2h^2)\,v_2^2\,\big[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\big] \;+\; \sigma_{s,k}^2
\end{aligned} \tag{9.86}$$

where $h = 2$ for complex data, while the positive constant v_2 is independent of $\mu_{\max}$ and denotes the norm

$$v_2 \;\triangleq\; \Big\|\big(\mathcal{V}_\epsilon^{-1}\big)^{\mathsf{T}}\Big\| \tag{9.87}$$

In this way, we can bound the term $\mathbb{E}\,\|\boldsymbol{s}_i^e\|^2$ as follows:

$$\begin{aligned}
\mathbb{E}\,\|\boldsymbol{s}_i^e\|^2 &= 2\left(\sum_{k=1}^{N}\mathbb{E}\,\|\boldsymbol{s}_{k,i}\|^2\right) \\
&\leq v_2^2\beta_d^2\,\big[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\big] \;+\; \sigma_s^2
\end{aligned} \tag{9.88}$$

where we introduced the scalars:

$$\beta_d^2 \;\triangleq\; \sum_{k=1}^{N}\beta_k^2/h^2 \tag{9.89}$$

$$\sigma_s^2 \;\triangleq\; \sum_{k=1}^{N} 2\,\sigma_{s,k}^2 \tag{9.90}$$

Substituting into (9.82) we get

$$\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2 + \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2 \;\leq\; v_1^2v_2^2\beta_d^2\mu_{\max}^2\,\big[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 + \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\big] + v_1^2\mu_{\max}^2\sigma_s^2 \tag{9.91}$$

Using this bound in (9.69) and (9.81) we find that

$$\begin{aligned}\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^2 \;\leq\; & \big(1-\sigma_{11}\mu_{\max}+v_1^2v_2^2\beta_d^2\mu_{\max}^2\big)\,\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \;+\\ & \left(\frac{\sigma_{21}^2\mu_{\max}}{\sigma_{11}}+v_1^2v_2^2\beta_d^2\mu_{\max}^2\right)\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; v_1^2\mu_{\max}^2\sigma_s^2\end{aligned} \tag{9.92}$$

and

$$\begin{aligned}\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2 \;\leq\; & \left(\rho(J_\epsilon)+\epsilon+\frac{3\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}+v_1^2v_2^2\beta_d^2\mu_{\max}^2\right)\mathbb{E}\,\big\|\check{\boldsymbol{w}}_{i-1}^e\big\|^2+\\ & \left(\frac{3\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}+v_1^2v_2^2\beta_d^2\mu_{\max}^2\right)\mathbb{E}\,\big\|\bar{\boldsymbol{w}}_{i-1}^e\big\|^2+\\ & \left(\frac{3}{1-\rho(J_\epsilon)-\epsilon}\right)\|\check{b}^e\|^2+v_1^2\mu_{\max}^2\sigma_s^2\end{aligned} \tag{9.93}$$

We introduce the scalar coefficients

$$a \;=\; 1-\sigma_{11}\mu_{\max}+v_1^2v_2^2\beta_d^2\mu_{\max}^2 \;=\; 1-O(\mu_{\max}) \tag{9.94}$$

$$b \;=\; \frac{\sigma_{21}^2\mu_{\max}}{\sigma_{11}}+v_1^2v_2^2\beta_d^2\mu_{\max}^2 \;=\; O(\mu_{\max}) \tag{9.95}$$

$$c \;=\; \frac{3\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}+v_1^2v_2^2\beta_d^2\mu_{\max}^2 \;=\; O(\mu_{\max}^2) \tag{9.96}$$

$$\begin{aligned}d \;&=\; \rho(J_\epsilon)+\epsilon+\frac{3\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}+v_1^2v_2^2\beta_d^2\mu_{\max}^2\\ &=\; \rho(J_\epsilon)+\epsilon+O(\mu_{\max}^2)\end{aligned} \tag{9.97}$$

$$e \;=\; v_1^2\mu_{\max}^2\sigma_s^2 \;=\; O(\mu_{\max}^2) \tag{9.98}$$

$$f \;=\; \left(\frac{3}{1-\rho(J_\epsilon)-\epsilon}\right)\|\check{b}^e\|^2 \;=\; O(\mu_{\max}^2) \tag{9.99}$$

since $\|\check{b}^e\| = O(\mu_{\max})$. Using these parameters, we can combine (9.92) and (9.93) into a single compact inequality recursion as follows:

$$\begin{bmatrix}\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^2\\ \mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2\end{bmatrix} \preceq \underbrace{\begin{bmatrix}a & b\\ c & d\end{bmatrix}}_{\Gamma}\begin{bmatrix}\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2\\ \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\end{bmatrix}+\begin{bmatrix}e\\ e+f\end{bmatrix} \tag{9.100}$$

in terms of the 2×2 coefficient matrix Γ indicated above and whose entries are of the form

$$\Gamma \;=\; \begin{bmatrix}1-O(\mu_{\max}) & O(\mu_{\max})\\ O(\mu_{\max}^2) & \rho(J_\epsilon)+\epsilon+O(\mu_{\max}^2)\end{bmatrix} \tag{9.101}$$

Now, we invoke again the property that the spectral radius of a matrix is upper bounded by any of its norms, and use the $1-$norm (maximum absolute column sum), to conclude that

$$\rho(\Gamma) \;\leq\; \max\left\{1 - O(\mu_{\max}) + O(\mu_{\max}^2),\; \rho(J_\epsilon) + \epsilon + O(\mu_{\max}) + O(\mu_{\max}^2)\right\} \tag{9.102}$$

Since $\rho(J_\epsilon) < 1$ is independent of $\mu_{\max}$, and since ϵ and $\mu_{\max}$ are small positive numbers that can be chosen arbitrarily small and independently of each other, it is clear that the right-hand side of the above expression can be made strictly smaller than one for sufficiently small ϵ and $\mu_{\max}$. In that case, $\rho(\Gamma) < 1$ so that Γ is stable. Moreover, it holds that

$$\begin{aligned}
(I_2 - \Gamma)^{-1} &= \begin{bmatrix} 1-a & -b \\ -c & 1-d \end{bmatrix}^{-1} \\
&= \frac{1}{(1-a)(1-d) - bc}\begin{bmatrix} 1-d & b \\ c & 1-a \end{bmatrix} \\
&= \begin{bmatrix} O(1/\mu_{\max}) & O(1) \\ O(\mu_{\max}) & O(1) \end{bmatrix}
\end{aligned} \tag{9.103}$$

If we now iterate (9.100), and since Γ is stable, we conclude that

$$\begin{aligned}
\limsup_{i\to\infty} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2 \end{bmatrix} &\preceq (I_2 - \Gamma)^{-1}\begin{bmatrix} e \\ e+f \end{bmatrix} \\
&= \begin{bmatrix} O(1/\mu_{\max}) & O(1) \\ O(\mu_{\max}) & O(1) \end{bmatrix}\begin{bmatrix} O(\mu_{\max}^2) \\ O(\mu_{\max}^2) \end{bmatrix} \\
&= \begin{bmatrix} O(\mu_{\max}) \\ O(\mu_{\max}^2) \end{bmatrix}
\end{aligned} \tag{9.104}$$

from which we conclude that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^2 = O(\mu_{\max}), \quad \limsup_{i\to\infty} \mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2 = O(\mu_{\max}^2) \tag{9.105}$$

and, therefore,

$$\begin{aligned}
\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^2 &= \limsup_{i\to\infty}\; \mathbb{E}\left\|\left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}}\begin{bmatrix} \bar{\boldsymbol{w}}_i^e \\ \check{\boldsymbol{w}}_i^e \end{bmatrix}\right\|^2 \\
&\leq \limsup_{i\to\infty}\; v_2^2\left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^2 \;+\; \mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^2\right] \\
&= O(\mu_{\max})
\end{aligned} \tag{9.106}$$

which leads to the desired result (9.11).

□

We remark that the type of derivation used in the above proof, which starts from a stochastic recursion of the form (9.60) and transforms it into a deterministic recursion of the form (9.100), with the sizes of the parameters specified in terms of $\mu_{\max}$ and with a Γ matrix of the form (9.101), will be a recurring technique in our presentation. For example, we will encounter a similar derivation in two more locations in the current chapter while establishing Theorems 9.2 and 9.6 further ahead — see expressions (9.153) and (9.301); these theorems deal with the stability of the fourth and first-order moments of the error vector. We will also encounter a similar derivation in the next chapter — see expressions (10.48), (10.77), and (10.89).

9.2 Stability of Fourth-Order Error Moment

In the next chapter we will derive a long-term model to approximate the behavior of the network in the long term, as $i \to \infty$, and for sufficiently small step-sizes. The long-term model will be more tractable for performance analysis in the steady-state regime. At that point, we will argue that performance results that are derived from analyzing the long-term model provide accurate expressions for the performance results of the original network model to first-order in the step-size parameters. This is a reassuring conclusion that will lead to useful closed-form performance expressions. These results will be established under the condition that the fourth-order moment of the error vector, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^4$, is asymptotically stable. We therefore establish this fact here and call upon it later in the analysis. To do so, we will rely on condition (8.121) on the fourth-order moments of the individual gradient noise processes.

Theorem 9.2 (Fourth-order moment stability)**.** Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P = A_1 A_o A_2$. Assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumption 6.1. Assume further that the first and fourth-order moments of the gradient noise process satisfy the conditions of Assumption 8.1 with the second-order moment condition (8.115) replaced by the fourth-order moment condition (8.121). Then, the fourth-order moments of the network error vectors are stable for sufficiently small step-sizes,

namely, it holds that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^4 \;=\; O(\mu_{\max}^2), \quad k=1,2,\ldots,N \tag{9.107}$$

Proof. We again establish the result for the general case of complex data and, therefore, $h=2$ throughout this derivation. We recall relations (9.61)–(9.62), namely,

$$\bar{\boldsymbol{w}}_i^e \;=\; (I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e \;-\; \boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e \;+\; \bar{\boldsymbol{s}}_i^e \tag{9.108}$$

$$\check{\boldsymbol{w}}_i^e \;=\; (\mathcal{J}_\epsilon^{\mathsf{T}}-\boldsymbol{D}_{22,i-1}^{\mathsf{T}})\check{\boldsymbol{w}}_{i-1}^e \;-\; \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e \;+\; \check{\boldsymbol{s}}_i^e \;-\; \check{b}^e \tag{9.109}$$

Now note that, for any (deterministic or random) column vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, it holds that

$$\begin{aligned}\|\boldsymbol{a}+\boldsymbol{b}\|^4 \;=\;& \|\boldsymbol{a}\|^4+\|\boldsymbol{b}\|^4+2\|\boldsymbol{a}\|^2\,\|\boldsymbol{b}\|^2 + \\ & 4\mathrm{Re}(\boldsymbol{a}^*\boldsymbol{b})\left[\|\boldsymbol{a}\|^2+\|\boldsymbol{b}\|^2+\mathrm{Re}(\boldsymbol{a}^*\boldsymbol{b})\right]\end{aligned} \tag{9.110}$$

so that using the vector inequalities

$$[\mathrm{Re}(\boldsymbol{a}^*\boldsymbol{b})]^2 \le |\boldsymbol{a}^*\boldsymbol{b}|^2 \;\le\; \|\boldsymbol{a}\|^2\,\|\boldsymbol{b}\|^2 \tag{9.111}$$

and

$$2\mathrm{Re}(\boldsymbol{a}^*\boldsymbol{b}) \le \|\boldsymbol{a}\|^2+\|\boldsymbol{b}\|^2 \tag{9.112}$$

we get

$$\|\boldsymbol{a}+\boldsymbol{b}\|^4 \;\le\; \|\boldsymbol{a}\|^4+3\|\boldsymbol{b}\|^4+8\|\boldsymbol{a}\|^2\,\|\boldsymbol{b}\|^2+4\|\boldsymbol{a}\|^2\,\mathrm{Re}(\boldsymbol{a}^*\boldsymbol{b}) \tag{9.113}$$

Applying this inequality to (9.108) with the identifications

$$\boldsymbol{a} \;\leftarrow\; (I_{2M}-\boldsymbol{\mathcal{D}}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e-\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e \tag{9.114}$$

$$\boldsymbol{b} \;\leftarrow\; \bar{\boldsymbol{s}}_i^e \tag{9.115}$$

we obtain

$$\begin{aligned}\|\bar{\boldsymbol{w}}_i^e\|^4 \;\le\;& \|(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e-\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\|^4 \;+\; 3\|\bar{\boldsymbol{s}}_i^e\|^4 \;+ \\ & 8\|(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e \;-\; \boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\|^2\,\|\bar{\boldsymbol{s}}_i^e\|^2 \;+ \\ & 4\|(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e-\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\|^2\,\mathrm{Re}(\boldsymbol{a}^*\bar{\boldsymbol{s}}_i^e)\end{aligned} \tag{9.116}$$

Conditioning on $\boldsymbol{\mathcal{F}}_{i-1}$ and computing the expectations of both sides, we will find that the expectation of the last term on the right-hand side of the above

expression is zero due to the assumed properties on the gradient noise process. Taking expectations again we then conclude that

$$\begin{aligned}
\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^4 &\leq \mathbb{E}\,\|(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e-\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\|^4 \;+\; 3\left(\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^4\right) \;+\\
&\qquad 8\left(\mathbb{E}\,\|(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e-\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\|^2\right)\left(\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2\right)\\
&= \mathbb{E}\left\|(1-t)\frac{1}{1-t}(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e-t\frac{1}{t}\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\right\|^4 \;+\\
&\qquad 3\left(\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^4\right) \;+\; 8\left(\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2\right)\times\\
&\qquad \left(\mathbb{E}\left\|(1-t)\frac{1}{1-t}(I_{2M}-\boldsymbol{D}_{11,i-1}^{\mathsf{T}})\bar{\boldsymbol{w}}_{i-1}^e-t\frac{1}{t}\boldsymbol{D}_{21,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e\right\|^2\right)\\
&\leq \frac{(1-\sigma_{11}\mu_{\max})^4}{(1-t)^3}\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4+\frac{\sigma_{21}^4\mu_{\max}^4}{t^3}\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^4 \;+\; 3\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^4 \;+\\
&\qquad 8\left(\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2\right)\left(\frac{(1-\sigma_{11}\mu_{\max})^2}{1-t}\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^2+\frac{\sigma_{21}^2\mu_{\max}^2}{t}\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2\right)
\end{aligned}\tag{9.117}$$

for any arbitrary positive number $t\in(0,1)$. Similarly, using the identifications

$$\boldsymbol{a} \;\leftarrow\; (\mathcal{J}_\epsilon^{\mathsf{T}}-\boldsymbol{D}_{22,i-1}^{\mathsf{T}})\check{\boldsymbol{w}}_{i-1}^e-\boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e-\check{b}^e \tag{9.118}$$

$$\boldsymbol{b} \;\leftarrow\; \check{\boldsymbol{s}}_i^e \tag{9.119}$$

for relation (9.109), we can establish the inequality

$$\begin{aligned}
\|\check{\boldsymbol{w}}_i^e\|^4 \;\leq\; &\left\|(\mathcal{J}_\epsilon^{\mathsf{T}}-\boldsymbol{D}_{22,i-1}^{\mathsf{T}})\check{\boldsymbol{w}}_{i-1}^e-\boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e-\check{b}^e\right\|^4 \;+\; 3\|\check{\boldsymbol{s}}_i^e\|^4 \;+\\
&8\left\|(\mathcal{J}_\epsilon^{\mathsf{T}}-\boldsymbol{D}_{22,i-1}^{\mathsf{T}})\check{\boldsymbol{w}}_{i-1}^e-\boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e-\check{b}^e\right\|^2\|\check{\boldsymbol{s}}_i^e\|^2 \;+\\
&4\|\boldsymbol{a}\|^2\mathrm{Re}(\boldsymbol{a}^*\boldsymbol{b})
\end{aligned}\tag{9.120}$$

from which we conclude that, again for any positive scalar $t\in(0,1)$:

$$
\begin{aligned}
&\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^4 \\
&\leq \mathbb{E}\,\Big\|(\mathcal{J}_\epsilon^{\mathsf{T}} - \boldsymbol{D}_{22,i-1}^{\mathsf{T}})\check{\boldsymbol{w}}_{i-1}^e \;-\; \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e \;-\; \check{b}^e\Big\|^4 + 3\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^4 \;+ \\
&\quad 8\left(\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2\right)\left(\mathbb{E}\,\Big\|(\mathcal{J}_\epsilon^{\mathsf{T}} - \boldsymbol{D}_{22,i-1}^{\mathsf{T}})\check{\boldsymbol{w}}_{i-1}^e \;-\; \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e \;-\; \check{b}^e\Big\|^2\right) \\
&\leq \mathbb{E}\,\Big\|t\frac{1}{t}\mathcal{J}_\epsilon^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e - (1-t)\frac{1}{1-t}\left[\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e + \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e + \check{b}^e\right]\Big\|^4 + \\
&\quad 3\left(\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^4\right) \;+\; 8\left(\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2\right)\times \\
&\quad \left(\mathbb{E}\,\Big\|t\frac{1}{t}\mathcal{J}_\epsilon^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e - (1-t)\frac{1}{1-t}\left[\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e \;+\; \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e \;+\; \check{b}^e\right]\Big\|^2\right) \\
&\leq \frac{1}{t^3}\,\|\mathcal{J}_\epsilon\|^4\;\mathbb{E}\,\Big\|\check{\boldsymbol{w}}_{i-1}^e\Big\|^4 + \\
&\quad \frac{1}{(1-t)^3}\mathbb{E}\,\Big\|\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e + \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e + \check{b}^e\Big\|^4 + \\
&\quad 3\left(\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^4\right) \;+\; 8\left(\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2\right)\times \\
&\quad \left(\frac{1}{t}\|\mathcal{J}_\epsilon\|^2\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 + \frac{1}{1-t}\mathbb{E}\,\Big\|\boldsymbol{D}_{22,i-1}^{\mathsf{T}}\check{\boldsymbol{w}}_{i-1}^e + \boldsymbol{D}_{12,i-1}^{\mathsf{T}}\bar{\boldsymbol{w}}_{i-1}^e + \check{b}^e\Big\|^2\right) \\
&\leq \frac{(\rho(J_\epsilon)+\epsilon)^4}{t^3}\,\mathbb{E}\,\Big\|\check{\boldsymbol{w}}_{i-1}^e\Big\|^4 \;+ \\
&\quad \frac{27\sigma_{22}^4\mu_{\max}^4}{(1-t)^3}\mathbb{E}\,\Big\|\check{\boldsymbol{w}}_{i-1}^e\Big\|^4 + \frac{27\sigma_{12}^4\mu_{\max}^4}{(1-t)^3}\mathbb{E}\,\Big\|\bar{\boldsymbol{w}}_{i-1}^e\Big\|^4 + 27\|\check{b}^e\|^4 + \\
&\quad 3\left(\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^4\right) \;+\; 8\frac{(\rho(J_\epsilon)+\epsilon)^2}{t}\left(\mathbb{E}\,\Big\|\check{\boldsymbol{w}}_{i-1}^e\Big\|^2\right)\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2 + \\
&\quad 8\left(\frac{3\sigma_{22}^2\mu_{\max}^2}{1-t}\mathbb{E}\,\Big\|\check{\boldsymbol{w}}_{i-1}^e\Big\|^2 \;+\; \frac{3\sigma_{12}^2\mu_{\max}^2}{1-t}\mathbb{E}\,\Big\|\bar{\boldsymbol{w}}_{i-1}^e\Big\|^2 \;+\; 3\|\check{b}^e\|^2\right)\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2
\end{aligned}
\tag{9.121}
$$

where in the last inequality we used the result

$$
\begin{aligned}
\|a+b+c\|^4 &= \left\|\frac{1}{3}3a + \frac{1}{3}3b + \frac{1}{3}3c\right\|^4 \\
&\leq \frac{1}{3}\|3a\|^4 + \frac{1}{3}\|3b\|^4 + \frac{1}{3}\|3c\|^4 \\
&= 27\|a\|^4 + 27\|b\|^4 + 27\|c\|^4
\end{aligned}
\tag{9.122}
$$

We now bound the fourth-order noise terms that appear in expressions (9.121) and (9.121) for $\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^4$ and $\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^4$, namely, $\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^4$ and $\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^4$. Thus, note

that

$$\begin{aligned}
\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^4 \;+\; \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^4 &\leq \mathbb{E}\,(\|\bar{\boldsymbol{s}}_i^e\|^2 \;+\; \|\check{\boldsymbol{s}}_i^e\|^2)^2 \\
&= \mathbb{E}\left(\left\|\begin{bmatrix}\bar{\boldsymbol{s}}_i^e \\ \check{\boldsymbol{s}}_i^e\end{bmatrix}\right\|^2\right)^2 \\
&= \mathbb{E}\,\left\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e\right\|^4 \\
&\leq \left\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\right\|^4\,\|\mathcal{M}\|^4\,\mathbb{E}\,\|\boldsymbol{s}_i^e\|^4 \\
&\leq v_1^4\mu_{\max}^4\,\mathbb{E}\,\|\boldsymbol{s}_i^e\|^4 \qquad (9.123)
\end{aligned}$$

On the other hand, using Jensen's inequality (F.26) and applying it to the convex function $f(x) = x^2$,

$$\begin{aligned}
\mathbb{E}\,\|\boldsymbol{s}_i^e\|^4 &= \mathbb{E}\,(\|\boldsymbol{s}_i^e\|^2)^2 \\
&= 4\,\mathbb{E}\left(\sum_{k=1}^{N}\|\boldsymbol{s}_{k,i}\|^2\right)^2 \\
&= 4\,\mathbb{E}\left(\frac{1}{N}N\|\boldsymbol{s}_{1,i}\|^2 + \frac{1}{N}N\|\boldsymbol{s}_{2,i}\|^2 + \ldots + \frac{1}{N}N\|\boldsymbol{s}_{N,i}\|^2\right)^2 \\
&\overset{(F.26)}{\leq} \frac{4}{N}\mathbb{E}\,\left((N\|\boldsymbol{s}_{1,i}\|^2)^2 + (N\|\boldsymbol{s}_{2,i}\|^2)^2 + \ldots + (N\|\boldsymbol{s}_{N,i}\|^2)^2\right) \\
&= 4N\sum_{k=1}^{N}\mathbb{E}\,\|\boldsymbol{s}_{k,i}\|^4 \qquad (9.124)
\end{aligned}$$

in terms of the fourth-order moments of the individual gradient noise processes, $\mathbb{E}\,\|\boldsymbol{s}_{k,i}\|^4$. Likewise, we have

$$\begin{aligned}
\sum_{\ell=1}^{N}\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^4 &\leq \left(\|\widetilde{\boldsymbol{w}}_{1,i-1}\|^2 + \|\widetilde{\boldsymbol{w}}_{2,i-1}\|^2 + \ldots + \|\widetilde{\boldsymbol{w}}_{N,i-1}\|^2\right)^2 \\
&= \left(\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\right)^2 \\
&= \left(\frac{1}{2}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^2\right)^2 \\
&= \frac{1}{4}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 \qquad (9.125)
\end{aligned}$$

Therefore, for each term $\boldsymbol{s}_{k,i}$ in (9.124) we can write

$$\begin{aligned}
\mathbb{E}\,\|\boldsymbol{s}_{k,i}\|^4 &\overset{(8.122)}{\leq} (\beta_{4,k}^4/h^4)\mathbb{E}\,\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^4 \;+\; \sigma_{s4,k}^4 \\
&= (\beta_{4,k}^4/h^4)\mathbb{E}\,\Big\|\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\,\widetilde{\boldsymbol{w}}_{\ell,i-1}\Big\|^4 \;+\; \sigma_{s4,k}^4 \\
&\overset{(F.26)}{\leq} (\beta_{4,k}^4/h^4)\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^4 \;+\; \sigma_{s4,k}^4 \\
&\leq (\beta_{4,k}^4/h^4)\sum_{\ell=1}^{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^4 \;+\; \sigma_{s4,k}^4 \\
&\overset{(9.125)}{\leq} (\beta_{4,k}^4/4h^4)\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 \;+\; \sigma_{s4,k}^4 \\
&= (\beta_{4,k}^4/4h^4)\mathbb{E}\,\Big\|\big(\mathcal{V}_\epsilon^{-1}\big)^{\mathsf{T}}\mathcal{V}_\epsilon^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1}^e\Big\|^4 \;+\; \sigma_{s4,k}^4 \\
&\leq (\beta_{4,k}^4/4h^4)\,\Big\|\big(\mathcal{V}_\epsilon^{-1}\big)^{\mathsf{T}}\Big\|^4\,\mathbb{E}\,\|\mathcal{V}_\epsilon^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 \;+\; \sigma_{s4,k}^4 \\
&\overset{(a)}{=} 2(\beta_{4,k}^4/4h^4)\,v_2^4\,\big[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4 \;+\; \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4\big] \;+\; \sigma_{s4,k}^4
\end{aligned} \tag{9.126}$$

where in step (a) we used (9.55) to conclude that

$$\begin{aligned}
\|\mathcal{V}_\epsilon^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 &= \left(\left\|\begin{bmatrix}\bar{\boldsymbol{w}}_{i-1}^e\\ \check{\boldsymbol{w}}_{i-1}^e\end{bmatrix}\right\|^2\right)^2 \\
&= \Big(\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \|\check{\boldsymbol{w}}_{i-1}^e\|^2\Big)^2 \\
&\leq 2\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4 \;+\; 2\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4
\end{aligned} \tag{9.127}$$

In this way, we can bound the term $\mathbb{E}\,\|\boldsymbol{s}_i^e\|^4$ as follows:

$$\mathbb{E}\,\|\boldsymbol{s}_i^e\|^4 \;\leq\; v_2^4\beta_{d4}^4\,\big[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4 \;+\; \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4\big] \;+\; \sigma_{s4}^4 \tag{9.128}$$

where we introduced the scalars:

$$\beta_{d4}^4 \;\triangleq\; 2N\left(\sum_{k=1}^{N}\beta_{4,k}^4/h^4\right) \tag{9.129}$$

$$\sigma_{s4}^4 \;\triangleq\; 4N\left(\sum_{k=1}^{N}\sigma_{s4,k}^4\right) \tag{9.130}$$

Substituting into (9.123) we get

$$\begin{aligned}\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^4+\mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^4 \;\;&\leq\;\; v_1^4v_2^4\beta_{d4}^4\mu_{\max}^4\left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4+\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4\right]+\\ &\quad v_1^4\mu_{\max}^4\sigma_{s4}^4\end{aligned}\tag{9.131}$$

Returning to (9.117), selecting $t=\sigma_{11}\mu_{\max}$, and using the bounds (9.91) and (9.131), we then find that

$$\begin{aligned}\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^4 \;\;&\leq\;\; (1-\sigma_{11}\mu_{\max})\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4\;+\;\frac{\sigma_{21}^4\mu_{\max}}{\sigma_{11}^3}\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^4\;+\\ &\quad 3v_1^4v_2^4\beta_{d4}^4\mu_{\max}^4\left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4+\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4\right]\;+\;3v_1^4\mu_{\max}^4\sigma_{s4}^4+\\ &\quad 8v_1^2v_2^2\beta_d^2\mu_{\max}^2(1-\sigma_{11}\mu_{\max})\left(\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^2\right)^2\;+\\ &\quad 8v_1^2v_2^2\beta_d^2\mu_{\max}^2(1-\sigma_{11}\mu_{\max})\left(\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^2\right)\left(\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2\right)+\\ &\quad 8v_1^2\mu_{\max}^2\sigma_s^2(1-\sigma_{11}\mu_{\max})\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^2+\\ &\quad 8\frac{\sigma_{21}^2\mu_{\max}}{\sigma_{11}}v_1^2v_2^2\beta_d^2\mu_{\max}^2\left(\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2\right)^2+\\ &\quad 8\frac{\sigma_{21}^2\mu_{\max}}{\sigma_{11}}v_1^2v_2^2\beta_d^2\mu_{\max}^2\left(\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2\right)\left(\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^2\right)+\\ &\quad 8\frac{\sigma_{21}^2\mu_{\max}}{\sigma_{11}}v_1^2\mu_{\max}^2\sigma_s^2\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\end{aligned}\tag{9.132}$$

Now, for any real random variables $\boldsymbol{a}$ and $\boldsymbol{b}$ it holds that

$$(\mathbb{E}\,\boldsymbol{a})^2\leq\mathbb{E}\,\boldsymbol{a}^2\tag{9.133}$$

and

$$2\left(\mathbb{E}\,\boldsymbol{a}^2\right)\cdot\left(\mathbb{E}\,\boldsymbol{b}^2\right)\;\leq\;\mathbb{E}\,\boldsymbol{a}^4\;+\;\mathbb{E}\,\boldsymbol{b}^4\tag{9.134}$$

This latter property can be established as follows. Using $(\mathbb{E}\,\boldsymbol{a}^2-\mathbb{E}\,\boldsymbol{b}^2)^2\geq 0$, we get

$$\begin{aligned}2\left(\mathbb{E}\,\boldsymbol{a}^2\right)\cdot\left(\mathbb{E}\,\boldsymbol{b}^2\right)\;\;&\leq\;\;\left(\mathbb{E}\,\boldsymbol{a}^2\right)^2\;+\;\left(\mathbb{E}\,\boldsymbol{b}^2\right)^2\\ &\leq\;\;\mathbb{E}\,\boldsymbol{a}^4\;+\;\mathbb{E}\,\boldsymbol{b}^4\end{aligned}\tag{9.135}$$

These properties enable us to write

$$2\left(\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2\right)\left(\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^2\right)\;\leq\;\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^4\;+\;\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^4\tag{9.136}$$

$$\left(\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^2\right)^2\;\leq\;\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^4\tag{9.137}$$

$$\left(\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^2\right)^2\;\leq\;\mathbb{E}\,\left\|\bar{\boldsymbol{w}}_{i-1}^e\right\|^4\tag{9.138}$$

so that

$$\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^4 \;\le\; a\,\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4 + b\,\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^4 + a'\,\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 + b'\,\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 + e \tag{9.139}$$

where the constant parameters $\{a, b, a', b', e\}$ have the following form

$$\begin{aligned}
a &= 1 - \sigma_{11}\mu_{\max} + O(\mu_{\max}^2) && (9.140)\\
b &= O(\mu_{\max}) && (9.141)\\
a' &= O(\mu_{\max}^2) && (9.142)\\
b' &= O(\mu_{\max}^3) && (9.143)\\
e &= O(\mu_{\max}^4) && (9.144)
\end{aligned}$$

In a similar manner, using (9.121) and selecting

$$t = \rho(J_\epsilon) + \epsilon < 1 \tag{9.145}$$

we can verify that

$$\begin{aligned}
\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^4 \;\le\;& (\rho(J_\epsilon)+\epsilon)\,\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4 \;+\\
&\frac{27\mu_{\max}^4}{(1-\rho(J_\epsilon)-\epsilon)^3}\left[\sigma_{22}^4\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4 \;+\; \sigma_{12}^4\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4\right] \;+\\
&3v_1^4v_2^4\beta_{d4}^4\mu_{\max}^4\left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4 + \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4\right] \;+\; 3v_1^4\mu_{\max}^4\sigma_{s4}^4\\
&8\,(\rho(J_\epsilon)+\epsilon)\,v_1^2\mu_{\max}^2\sigma_s^2\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; 27\|\check{b}^e\|^4 \;+\\
&4\,(\rho(J_\epsilon)+\epsilon)\,v_1^2v_2^2\beta_d^2\mu_{\max}^2\left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4 + 3\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4\right] \;+\\
&\frac{24\mu_{\max}^4v_1^2\sigma_s^2}{1-\rho(J_\epsilon)-\epsilon}\left[\sigma_{22}^2\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 + \sigma_{12}^2\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2\right] \;+\\
&\frac{12\mu_{\max}^4v_1^2v_2^2\beta_d^2}{1-\rho(J_\epsilon)-\epsilon}\;\times\\
&\left[(\sigma_{22}^2+3\sigma_{12}^2)\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 + (\sigma_{12}^2+3\sigma_{22}^2)\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\right] +\\
&24\|\check{b}^e\|^2v_1^2v_2^2\beta_d^2\mu_{\max}^2\left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 + \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\right] \;+\\
&24\|\check{b}^e\|^2v_1^2\mu_{\max}^2\sigma_s^2
\end{aligned} \tag{9.146}$$

so that

$$\mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^4 \;\le\; c\,\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4 + d\,\mathbb{E}\,\left\|\check{\boldsymbol{w}}_{i-1}^e\right\|^4 + c'\,\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 + d'\,\mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 + f \tag{9.147}$$

where the constant parameters $\{c, d, c', d', f\}$ have the following form

$$
\begin{aligned}
c &= O(\mu_{\max}^2) && (9.148)\\
d &= \rho(J_\epsilon) + \epsilon + O(\mu_{\max}^2) && (9.149)\\
c' &= O(\mu_{\max}^4) && (9.150)\\
d' &= O(\mu_{\max}^2) && (9.151)\\
f &= O(\mu_{\max}^4) && (9.152)
\end{aligned}
$$

In other words, we can write

$$
\begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^4 \\ \mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^4 \end{bmatrix} \preceq \underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{\triangleq\, \Gamma} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4 \\ \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4 \end{bmatrix} + \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{9.153}
$$

in terms of the 2×2 coefficient matrix Γ indicated above and whose entries are of the form

$$
\Gamma = \begin{bmatrix} 1 - O(\mu_{\max}) & O(\mu_{\max}) \\ O(\mu_{\max}^2) & \rho(J_\epsilon) + \epsilon + O(\mu_{\max}^2) \end{bmatrix} \tag{9.154}
$$

We again find that Γ is a stable matrix for sufficiently small $\mu_{\max}$ and ϵ. Moreover, using (9.105) we have

$$
\limsup_{i\to\infty} \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 \end{bmatrix} = \begin{bmatrix} O(\mu_{\max}^3) \\ O(\mu_{\max}^4) \end{bmatrix} \tag{9.155}
$$

In this case, we can iterate (9.153) and use (9.103) to conclude that

$$
\limsup_{i\to\infty} \mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^4 = O(\mu_{\max}^2), \quad \limsup_{i\to\infty} \mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^4 = O(\mu_{\max}^4) \tag{9.156}
$$

and, therefore,

$$
\begin{aligned}
\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^4 &= \limsup_{i\to\infty}\ \mathbb{E}\left(\left\|\left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} \begin{bmatrix} \bar{\boldsymbol{w}}_i^e \\ \check{\boldsymbol{w}}_i^e \end{bmatrix}\right\|^2\right)^2 \\
&\le \left\|\left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}}\right\|^4 \left(\limsup_{i\to\infty}\ \mathbb{E}\,\left(\|\bar{\boldsymbol{w}}_i^e\|^2 + \|\check{\boldsymbol{w}}_i^e\|^2\right)^2\right) \\
&\le \limsup_{i\to\infty}\ 2v_2^4\left(\mathbb{E}\,\|\bar{\boldsymbol{w}}_i^e\|^4 \ +\ \mathbb{E}\,\|\check{\boldsymbol{w}}_i^e\|^4\right) \\
&= O(\mu_{\max}^2) && (9.157)
\end{aligned}
$$

which leads to the desired result (9.107).

□

9.3 Stability of First-Order Error Moment

Using the fact that $(\mathbb{E}\,\boldsymbol{a})^2 \leq \mathbb{E}\,\boldsymbol{a}^2$ for any real-valued random variable $\boldsymbol{a}$, we can readily conclude from (9.11), by using $\boldsymbol{a} = \|\widetilde{\boldsymbol{w}}_{k,i}\|$, that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\| \;=\; O(\mu_{\max}^{1/2}), \quad k = 1, 2, \ldots, N \tag{9.158}$$

so that the first-order moment of the error vector tends to a bounded region in the order of $O(\mu_{\max}^{1/2})$. However, a smaller upper bound on $\|\mathbb{E}\,\widetilde{\boldsymbol{w}}_{k,i}\|$ can be derived with $O(\mu_{\max}^{1/2})$ replaced by $O(\mu_{\max})$, as shown in (9.1) and as we proceed to verify in this section. To do so, we examine the evolution of the mean-error vector more closely.

We reconsider the network error recursion (9.12), namely,

$$\widetilde{\boldsymbol{w}}_i^e \;=\; \boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^e \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e, \;\; i \geq 0 \tag{9.159}$$

where, from the expressions in Lemma 8.1:

$$\boldsymbol{\mathcal{B}}_{i-1} \;=\; \mathcal{P}^{\mathsf{T}} \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{A}_1^{\mathsf{T}} \tag{9.160}$$

$$\mathcal{P}^{\mathsf{T}} \;=\; \mathcal{A}_2^{\mathsf{T}}\mathcal{A}_o^{\mathsf{T}}\mathcal{A}_1^{\mathsf{T}} \tag{9.161}$$

$$\boldsymbol{\mathcal{H}}_{i-1} \;\triangleq\; \mathrm{diag}\{\,\boldsymbol{H}_{1,i-1},\; \boldsymbol{H}_{2,i-1},\; \ldots,\; \boldsymbol{H}_{N,i-1}\,\} \tag{9.162}$$

$$\boldsymbol{H}_{k,i-1} \;\triangleq\; \int_0^1 \nabla_w^2 J_k(w^\star - t\widetilde{\boldsymbol{\phi}}_{k,i-1})dt \tag{9.163}$$

Conditioning both sides of (9.159) on $\boldsymbol{\mathcal{F}}_{i-1}$, invoking the conditions on the gradient noise process from Assumption 8.1, and computing the conditional expectations we obtain:

$$\mathbb{E}\,[\,\widetilde{\boldsymbol{w}}_i^e\,|\,\boldsymbol{\mathcal{F}}_{i-1}] \;=\; \boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^e \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \tag{9.164}$$

where the term involving $\boldsymbol{s}_i^e$ is eliminated since $\mathbb{E}\,[\boldsymbol{s}_i^e|\boldsymbol{\mathcal{F}}_{i-1}] = 0$. Taking expectations again we arrive at

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i^e \;=\; \mathbb{E}\,[\,\boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^e] \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \tag{9.165}$$

Let

$$\widetilde{\boldsymbol{\mathcal{H}}}_{i-1} \;\triangleq\; \mathcal{H} \;-\; \boldsymbol{\mathcal{H}}_{i-1} \tag{9.166}$$

where, in a manner similar to (9.162), we define the constant matrix

$$\mathcal{H} \;\triangleq\; \mathrm{diag}\{\,H_1,\; H_2,\; \ldots,\; H_N\,\} \tag{9.167}$$

with each $H_{k,i-1}$ given by the value of the Hessian matrix at the limit point defined by (8.55), namely,

$$H_k \triangleq \nabla_w^2 J_k(w^\star) \tag{9.168}$$

Then, using (9.166) in the expression for $\boldsymbol{\mathcal{B}}_{i-1}$, we can write

$$\begin{aligned} \boldsymbol{\mathcal{B}}_{i-1} &= \mathcal{P}^{\mathsf{T}} - \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\mathcal{H}\mathcal{A}_1^{\mathsf{T}} + \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\widetilde{\boldsymbol{\mathcal{H}}}_{i-1}\mathcal{A}_1^{\mathsf{T}} \\ &\triangleq \mathcal{B} + \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\widetilde{\boldsymbol{\mathcal{H}}}_{i-1}\mathcal{A}_1^{\mathsf{T}} \end{aligned} \tag{9.169}$$

in terms of the constant coefficient matrix

$$\mathcal{B} \triangleq \mathcal{P}^{\mathsf{T}} - \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\mathcal{H}\mathcal{A}_1^{\mathsf{T}} \tag{9.170}$$

In this way, the mean-error relation (9.165) becomes

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i^e = \mathcal{B}\,(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}^e) - \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e + \mathcal{A}_2^{\mathsf{T}}\mathcal{M}c_{i-1} \tag{9.171}$$

in terms of a deterministic perturbation sequence defined by

$$c_{i-1} \triangleq \mathbb{E}\left(\widetilde{\boldsymbol{\mathcal{H}}}_{i-1}\mathcal{A}_1^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1}^e\right) \tag{9.172}$$

The constant matrix $\mathcal{B}$ defined by (9.170), and which drives the mean-error recursion (9.171), will play a critical role in characterizing the performance of multi-agent networks in future chapters. It also plays an important role in characterizing the mean-error stability of the network in this section. We therefore establish several important properties for $\mathcal{B}$ and subsequently use these properties to establish result (9.1) later in Theorem 9.6.

Stability of the Coefficient Matrix $\mathcal{B}$

The first key result pertains to the stability of the matrix $\mathcal{B}$ for sufficiently small step-sizes.

Theorem 9.3 (Stability of $\mathcal{B}$)**.** Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P = A_1 A_o A_2$. Assume the aggregate cost (9.10) satisfies condition (6.13) in Assumption 6.1. Then, the constant matrix $\mathcal{B}$ defined by (9.170) is stable for sufficiently small step-sizes and its spectral radius is given by

$$\rho(\mathcal{B}) = 1 - \lambda_{\min}\left(\sum_{k=1}^{N} q_k H_k\right) + O\left(\mu_{\max}^{(N+1)/N}\right) \tag{9.173}$$

where $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue of its Hermitian matrix argument.

Proof. We first establish the result for diffusion and consensus networks and then extend the conclusion to the general distributed structure (8.46) with three combination matrices $\{A_1, A_o, A_2\}$. The arguments used in steps (a) and (b) below are justified when all step-sizes in $\mathcal{M}$ are strictly positive, which is the situation under study. The more general argument under step (c) below is applicable even to situations where some of the step-sizes are zero (a scenario we shall encounter later in Chapter 13).

(a) Diffusion strategies. For the case of diffusion strategies, the stability argument follows directly by examining the expression for the matrix $\mathcal{B}$. Recall that different choices for $\{A_o, A_1, A_2\}$ correspond to different strategies, as already shown by (8.7)–(8.10). In particular, for ATC and CTA diffusion, we set $A_1 = A$ or $A_2 = A$, for some left-stochastic matrix A, and the matrix A_o disappears from $\mathcal{B}$ since $A_o = I_N$ for these strategies. Specifically, the expression for $\mathcal{B}$ becomes

$$\mathcal{B}_{\rm atc} = \mathcal{A}^{\mathsf{T}}\,(I_{2MN} - \mathcal{M}\mathcal{H}) \tag{9.174}$$

$$\mathcal{B}_{\rm cta} = (I_{2MN} - \mathcal{M}\mathcal{H})\,\mathcal{A}^{\mathsf{T}} \tag{9.175}$$

where $\mathcal{A} = A \otimes I_{2M}$ is left-stochastic and

$$\mathcal{M} \triangleq \mathrm{diag}\{\,\mu_1 I_{2M},\ \mu_2 I_{2M},\ \ldots,\ \mu_N I_{2M}\,\} \tag{9.176}$$

$$\mathcal{H} \triangleq \mathrm{diag}\,\{\,H_1,\ H_2,\ \ldots,\ H_N\,\} \tag{9.177}$$

The important fact to note from (9.174) and (9.175) is that the combination matrix $\mathcal{A}^{\mathsf{T}}$ appears *multiplying* (from left or right) the block diagonal matrix $I_{2MN} - \mathcal{M}\mathcal{H}$. We can then immediately call upon result (F.24) from the appendix, and employ the block maximum norm with blocks of size $2M \times 2M$ each, to conclude that

$$\rho\,(\mathcal{B}_{\rm atc}) \leq \rho\,(I_{2MN} - \mathcal{M}\mathcal{H}) \tag{9.178}$$

$$\rho\,(\mathcal{B}_{\rm cta}) \leq \rho\,(I_{2MN} - \mathcal{M}\mathcal{H}) \tag{9.179}$$

Therefore, for both cases of ATC and CTA diffusion, the respective coefficient matrices $\mathcal{B}$ become stable whenever the block-diagonal matrix $I_{2MN} - \mathcal{M}\mathcal{H}$ is stable. It is easily seen that this latter condition is guaranteed for step-sizes μ_k satisfying

$$\mu_k < \frac{2}{\rho(H_k)}, \quad k = 1, 2, \ldots, N \tag{9.180}$$

from which we conclude that sufficiently small step-sizes stabilize $\mathcal{B}_{\rm atc}$ or $\mathcal{B}_{\rm cta}$.

(b) Consensus strategy. For the consensus strategy, we set $A_1 = A_2 = I_N$ and $A_o = A$. In this case, the expression for $\mathcal{B}$ becomes

$$\mathcal{B}_{\rm cons} \;=\; \mathcal{A}^{\mathsf{T}} - \mathcal{M}\mathcal{H} \tag{9.181}$$

where $\mathcal{A}$ now appears as an additive term. A condition on the step-sizes to ensure the stability of $\mathcal{B}_{\rm cons}$ can be deduced from Weyl's Theorem (F.33) in the appendix if we additionally assume that the left-stochastic matrix A is *symmetric* [248], in which case it will also be doubly stochastic. Since A is then both symmetric and left-stochastic, its eigenvalues will be real and lie inside the interval $[-1, 1]$. Hence, $(I_{2MN} - \mathcal{A}^{\mathsf{T}}) \geq 0$. Moreover, since the matrices $\mathcal{M}$ and $\mathcal{H}$ are block-diagonal Hermitian and commute with each other, i.e., $\mathcal{H}\mathcal{M} = \mathcal{M}\mathcal{H}$, it follows that $\mathcal{B}_{\rm cons}$ in (9.181) is Hermitian, as well as the matrix $\mathcal{B}_{\rm ncop} = I_{2MN} - \mathcal{M}\mathcal{H}$. Now note that we can write the following two trivial equalities (by adding and subtracting equal terms):

$$\mathcal{B}_{\rm ncop} \;=\; \mathcal{B}_{\rm cons} + (I_{2MN} - \mathcal{A}^{\mathsf{T}}) \tag{9.182}$$

$$\mathcal{B}_{\rm cons} \;=\; (\lambda_{\min}(A) \cdot I_{2MN} - \mathcal{M}\mathcal{H}) + \left(\mathcal{A}^{\mathsf{T}} - \lambda_{\min}(A) \cdot I_{2MN}\right) \tag{9.183}$$

so that by applying Weyl's Theorem (F.33) to both representations, we obtain the following eigenvalue relations:

$$\lambda_\ell(\mathcal{B}_{\rm cons}) \;\leq\; \lambda_\ell(\mathcal{B}_{\rm ncop}) \tag{9.184}$$

$$\lambda_\ell(\mathcal{B}_{\rm cons}) \;\geq\; \lambda_\ell\left\{\lambda_{\min}(A) \cdot I_{2MN} - \mathcal{M}\mathcal{H}\right\} \tag{9.185}$$

for $\ell = 1, 2, \ldots, 2MN$ and where we are assuming ordered eigenvalues, namely, $\lambda_1 \geq \lambda_2 \geq \ldots$, for any of the matrix arguments. It follows that the matrix $\mathcal{B}_{\rm cons}$ will be stable, namely, $-1 < \lambda_\ell(\mathcal{B}_{\rm cons}) < 1$ for all ℓ if

$$\lambda_\ell(\mathcal{B}_{\rm ncop}) \;<\; 1 \tag{9.186}$$

$$\lambda_\ell\left\{\lambda_{\min}(A) \cdot I_{2MN} - \mathcal{M}\mathcal{H}\right\} \;>\; -1 \tag{9.187}$$

The first condition is automatically satisfied due to the form of the matrix $\mathcal{B}_{\rm ncop}$ and since $\mathcal{M}\mathcal{H} > 0$. For the second condition, it will be satisfied by step-sizes $\{\mu_k\}$ such that

$$\mu_k \;<\; \frac{1 + \lambda_{\min}(A)}{\rho(H_k)}, \quad k = 1, 2, \ldots, N \tag{9.188}$$

Since we are dealing with strongly-connected networks, the matrix A is primitive and, therefore, it has a single eigenvalue matching its spectral radius, which is equal to one. That eigenvalue occurs at $+1$ so that

$\lambda_{\min}(A) > -1$ and the upper bound in (9.188) is positive. We therefore conclude that sufficiently small step-sizes stabilize $\mathcal{B}$ for consensus strategies with a *symmetric* combination policy A. If A is not symmetric, then the next argument would apply to this case.

(c) General case (eigenvalue perturbation analysis). For the general case, when the matrix A_o is not necessarily the identity matrix or symmetric, and when all three matrices $\{A_o, A_1, A_2\}$ or subsets thereof may be present, the argument is more demanding. The argument that follows is based on an eigenvalue perturbation analysis in the small step-size regime similar to [277]. We establish the result for the general case of complex data and, therefore, $h = 2$ throughout this derivation.

We introduce the same Jordan canonical decomposition (9.24) for the matrix P, namely,

$$P \triangleq V_\epsilon J V_\epsilon^{-1} \tag{9.189}$$

$$J = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & J_\epsilon \end{array}\right] \tag{9.190}$$

where the matrix J_ϵ consists of Jordan blocks of forms similar to (9.25) with $\epsilon > 0$ appearing on the lower diagonal. The value of ϵ can be chosen to be arbitrarily small and is independent of $\mu_{\max}$. The Jordan decomposition of the extended matrix $\mathcal{P} = P \otimes I_{2M}$ is given by

$$\mathcal{P} = (V_\epsilon \otimes I_{2M})(J \otimes I_{2M})(V_\epsilon^{-1} \otimes I_{2M}) \tag{9.191}$$

so that substituting into (9.170) we obtain

$$\mathcal{B} = \left((V_\epsilon^{-1})^{\mathsf{T}} \otimes I_{2M}\right)\left\{(J^{\mathsf{T}} \otimes I_{2M}) - \mathcal{D}^{\mathsf{T}}\right\}\left(V_\epsilon^{\mathsf{T}} \otimes I_{2M}\right) \tag{9.192}$$

where

$$\begin{aligned} \mathcal{D}^{\mathsf{T}} &\triangleq \left(V_\epsilon^{\mathsf{T}} \otimes I_{2M}\right)\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\mathcal{H}\mathcal{A}_1^{\mathsf{T}}\left((V_\epsilon^{-1})^{\mathsf{T}} \otimes I_{2M}\right) \\ &\equiv \left[\begin{array}{cc} D_{11}^{\mathsf{T}} & D_{21}^{\mathsf{T}} \\ D_{12}^{\mathsf{T}} & D_{22}^{\mathsf{T}} \end{array}\right] \end{aligned} \tag{9.193}$$

Using the partitioning (9.23)–(9.24) and the fact that

$$\mathcal{A}_1 = A_1 \otimes I_{2M}, \quad \mathcal{A}_2 = A_2 \otimes I_{2M} \tag{9.194}$$

we find that the block entries $\{D_{mn}\}$ in (9.193) are given by

$$D_{11} = \sum_{k=1}^{N} q_k H_k^{\mathsf{T}} \tag{9.195}$$

$$D_{12} = (\mathbb{1}^{\mathsf{T}} \otimes I_{2M})\mathcal{H}^{\mathsf{T}}\mathcal{M}(A_2 V_R \otimes I_{2M}) \tag{9.196}$$

$$D_{21} = (V_L^{\mathsf{T}} A_1 \otimes I_{2M})\mathcal{H}^{\mathsf{T}}(q \otimes I_{2M}) \tag{9.197}$$

$$D_{22} = (V_L^{\mathsf{T}} A_1 \otimes I_{2M})\mathcal{H}^{\mathsf{T}}\mathcal{M}(A_2 V_R \otimes I_{2M}) \tag{9.198}$$

In a manner similar to the arguments used in the proof of Theorem 9.1, we can verify that

$$D_{11} = O(\mu_{\max}) \tag{9.199}$$

$$D_{12} = O(\mu_{\max}) \tag{9.200}$$

$$D_{21} = O(\mu_{\max}) \tag{9.201}$$

$$D_{22} = O(\mu_{\max}) \tag{9.202}$$

$$\rho(I_{2M} - D_{11}^{\mathsf{T}}) = 1 - \sigma_{11}\mu_{\max} = 1 - O(\mu_{\max}) \tag{9.203}$$

where σ_{11} is a positive scalar independent of $\mu_{\max}$.

Let

$$\mathcal{V}_\epsilon \stackrel{\Delta}{=} V_\epsilon \otimes I_{2M}, \qquad \mathcal{J}_\epsilon \stackrel{\Delta}{=} J_\epsilon \otimes I_{2M} \tag{9.204}$$

Then, using (9.192), we can write

$$\mathcal{B} = \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} \begin{bmatrix} I_{2M} - D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - D_{22}^{\mathsf{T}} \end{bmatrix} \mathcal{V}_\epsilon^{\mathsf{T}} \tag{9.205}$$

so that

$$\mathcal{V}_\epsilon^{\mathsf{T}} \mathcal{B} \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} = \begin{bmatrix} I_{2M} - D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - D_{22}^{\mathsf{T}} \end{bmatrix} \tag{9.206}$$

which shows that the matrix $\mathcal{B}$ is similar to, and therefore has the same eigenvalues as, the block matrix on the right-hand side, written as

$$\mathcal{B} \sim \begin{bmatrix} I_{2M} - O(\mu_{\max}) & O(\mu_{\max}) \\ O(\mu_{\max}) & \mathcal{J}_\epsilon^{\mathsf{T}} + O(\mu_{\max}) \end{bmatrix} \tag{9.207}$$

Now recall that J_ϵ is $(N-1) \times (N-1)$ and has a Jordan structure. For ease of presentation, and without any loss of generality, let us assume that J_ϵ consists of two Jordan blocks, say, as

$$J_\epsilon = \left[\begin{array}{ccc|ccc} \lambda_a & & & & & \\ \epsilon & \lambda_a & & & & \\ & & & \lambda_b & & \\ \hline & & & \epsilon & \lambda_b & \\ & & & & \epsilon & \lambda_b \end{array}\right] \tag{9.208}$$

Then, the matrix $\mathcal{J}_\epsilon = J_\epsilon \otimes I_{2M}$ has dimensions $2M(N-1) \times 2M(N-1)$ and is given by

$$\mathcal{J}_\epsilon = J_\epsilon \otimes I_{2M} \left[\begin{array}{cc|ccc} \lambda_a I_{2M} & & & & \\ \epsilon I_{2M} & \lambda_a I_{2M} & & & \\ \hline & & \lambda_b I_{2M} & & \\ & & \epsilon I_{2M} & \lambda_b I_{2M} & \\ & & & \epsilon I_{2M} & \lambda_b I_{2M} \end{array}\right] \tag{9.209}$$

More generically, for multiple Jordan blocks, it is clear that we can express $\mathcal{J}_\epsilon$ in the following lower-triangular form:

$$\mathcal{J}_\epsilon \;=\; \left[\begin{array}{ccccc} \lambda_{a,2} I_{2M} & & & & \\ & \lambda_{a,3} I_{2M} & & & \\ & & \ddots & & \\ \mathcal{K} & & & \ddots & \\ & & & & \lambda_{a,L} I_{2M} \end{array}\right] \tag{9.210}$$

with scalars $\{\lambda_{a,\ell}\}$ on the diagonal, all of which have norms strictly less than one, and where the entries of the strictly lower-triangular matrix $\mathcal{K}$ are either ϵ or zero. In the above representation, we are assuming that J_ϵ consists of several Jordan blocks. It follows that

$$\mathcal{J}_\epsilon^{\mathsf{T}} + O(\mu_{\max}) \;=\; \left[\begin{array}{ccc} \lambda_{a,2} I_{2M} + O(\mu_{\max}) & & \mathcal{K}^{\mathsf{T}} + O(\mu_{\max}) \\ & \ddots & \\ O(\mu_{\max}) & & \lambda_{a,L} I_{2M} + O(\mu_{\max}) \end{array}\right] \tag{9.211}$$

We introduce the eigen-decomposition of the Hermitian positive-definite matrix D_{11}^{T} and denote it by:

$$D_{11}^{\mathsf{T}} \;\triangleq\; U \Lambda U^* \tag{9.212}$$

where U is unitary and Λ has positive-diagonal entries $\{\lambda_k\}$; the matrices U and Λ are $2M \times 2M$. Using U, we further introduce the following block-diagonal similarity transformation:

$$\mathcal{T} \;\triangleq\; \mathrm{diag}\left\{\mu_{\max}^{1/N} U,\; \mu_{\max}^{2/N} I_{2M},\; \ldots,\; \mu_{\max}^{(N-1)/N} I_{2M},\; \mu_{\max} I_{2M}\right\} \tag{9.213}$$

where all block entries are defined in terms of I_{2M}, except for the first entry defined in terms of U. We now use (9.205) to get

$$\mathcal{T}^{-1}\left(\mathcal{V}_\epsilon^{\mathsf{T}} \mathcal{B} \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}}\right)\mathcal{T} \;= \tag{9.214}$$

$$\left[\begin{array}{c|ccc} B & O\left(\mu_{\max}^{(N+1)/N}\right) & & \\ \hline & \lambda_{a,2}I_{2M}+O(\mu_{\max}) & & O\left(\mu_{\max}^{1/N}\right) \\ O(\mu_{\max}^{1/N}) & & \ddots & \\ & O\left(\mu_{\max}^{1/N}\right) & & \lambda_{a,L}I_{2M}+O(\mu_{\max}) \end{array}\right]$$

where we introduced the $2M \times 2M$ diagonal matrix

$$B \triangleq I_{2M} - \Lambda \tag{9.215}$$

It follows from (9.214) that all off-diagonal entries of the above transformed matrix are at least $O(\mu_{\max}^{1/N})$. Although the factor $\mu_{\max}^{1/N}$ decays slower than $\mu_{\max}$, it nevertheless becomes small for sufficiently small $\mu_{\max}$. Then, calling upon Gershgorin's Theorem (F.37) from the appendix, we conclude from (9.214) that the eigenvalues of $\mathcal{B}$ are are either located in the Gershgorin circles that are centered at the eigenvalues of B with radii $O(\mu_{\max}^{(N+1)/N})$ or in the Gershgorin circles that are centered at the $\{\lambda_{a,\ell}\}$ with radii $O(\mu_{\max}^{1/N})$, namely,

$$|\lambda(\mathcal{B}) - \lambda(B)| \leq O\left(\mu_{\max}^{(N+1)/N}\right) \quad \text{or} \quad |\lambda(\mathcal{B}) - \lambda_{a,\ell}| \leq O\left(\mu_{\max}^{1/N}\right) \tag{9.216}$$

where $\lambda(\mathcal{B})$ and $\lambda(B)$ denote any of the eigenvalues of $\mathcal{B}$ and B, and $\ell = 2, \ldots, L$. It follows that

$$\rho(\mathcal{B}) \leq \rho(B) + O\left(\mu_{\max}^{(N+1)/N}\right) \quad \text{or} \quad \rho(\mathcal{B}) \leq \rho(J_\epsilon) + O(\mu_{\max}^{1/N}) \tag{9.217}$$

Now since J_ϵ is a stable matrix, we know that $\rho(J_\epsilon) < 1$. We express this spectral radius as

$$\rho(J_\epsilon) \;=\; 1 - \delta_J \tag{9.218}$$

where δ_J is positive and independent of $\mu_{\max}$. We also know from (9.203) that

$$\rho(B) = 1 - \sigma_{11}\mu_{\max} \;< 1 \tag{9.219}$$

since $B = U^*(I_{2M} - D_{11}^{\mathsf{T}})U$. We conclude from (9.217) that

$$\rho(\mathcal{B}) \leq 1 - \sigma_{11}\mu_{\max} + O\left(\mu_{\max}^{(N+1)/N}\right) \;\text{ or }\; \rho(\mathcal{B}) \leq 1 - \delta_J + O(\mu_{\max}^{1/N}) \tag{9.220}$$

If we now select $\mu_{\max} \ll 1$ small enough such that

$$O\left(\mu_{\max}^{(N+1)/N}\right) < \sigma_{11}\mu_{\max} \quad \text{and} \quad O\left(\mu_{\max}^{1/N}\right) + O(\mu_{\max}) < \delta_J \tag{9.221}$$

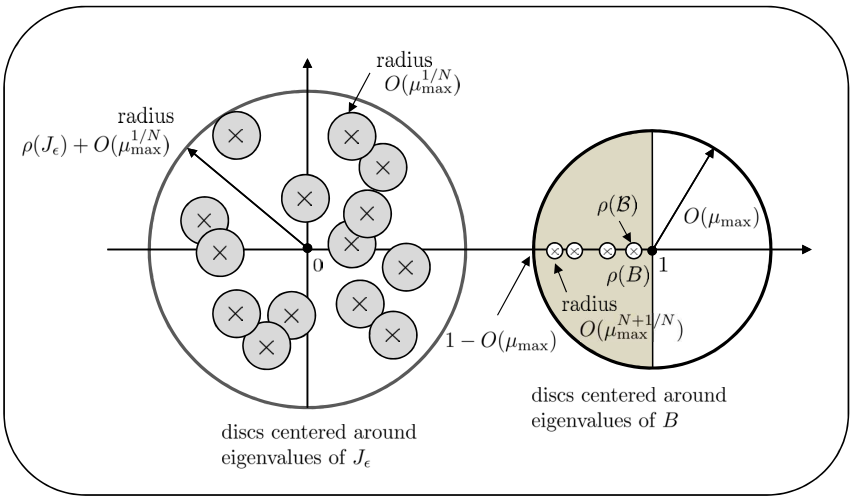

Figure 9.1: The larger circle on the left has radius $\rho(J_\epsilon) + O(\mu_{\max}^{1/N})$ and is disjoint from the smaller circle on the right whose radius is $O(\mu_{\max})$. The tiny discs inside the smaller circle on the right are disjoint and have radii $O(\mu_{\max}^{N+1/N})$ each. The eigenvalue corresponding to the spectral radius of $\mathcal{B}$ lies inside the rightmost smaller disc centered around $\rho(B)$.

then we would be able to conclude that $\rho(\mathcal{B}) < 1$ so that $\mathcal{B}$ is stable for sufficiently small step-sizes. Both conditions in (9.221) can be satisfied simultaneously and they will ensure

$$\rho(\mathcal{B}) = 1 - O(\mu_{\max}) \tag{9.222}$$

With regards to expression (9.173) for the spectral radius of $\mathcal{B}$, we call upon the stronger statement of Gershgorin's theorem mentioned after (F.37) in the appendix and which relates to how the eigenvalues of a matrix are distributed over disjoint Gershgorin sets. To begin with, note from (9.203) that for $\mu_{\max} \ll 1$, all eigenvalues of $B = I_{2M} - \Lambda$ are real-valued and positive. We then conclude from (9.222) that all eigenvalues of B lie inside the open interval

$$\lambda(B) \in (1 - O(\mu_{\max}), 1) \tag{9.223}$$

It further follows from this result that the eigenvalues of B are at most $O(\mu_{\max})$ apart from each other.

Now, referring to (9.216), the condition on the left describes a region in space that consists of the union of $2M$ Gershgorin discs: each disc is centered at one of the eigenvalues of B with radius $O(\mu_{\max}^{(N+1)/N})$. We can then choose

$\mu_{\max}$ small enough such that the discs that are centered at distinct eigenvalues of B remain disjoint from each other. The union of these discs will be contained within the circle that is centered at one and with radius $O(\mu_{\max})$ — see the region described by the smaller circle on the right in Figure 9.1.

Let us now examine the rightmost condition in (9.216). This condition describes a region in space that consists of the union of $2M(N-1)$ Gershgorin discs: each disc is now centered at an eigenvalue of $\mathcal{J}_\epsilon$ with radius $O(\mu_{\max}^{1/N})$. Therefore, again for $\mu_{\max} \ll 1$, the union of these discs is contained within a circle centered at the origin and with radius $\rho(J_\epsilon) + O(\mu_{\max}^{1/N})$; this radius is smaller than $1 - O(\mu_{\max})$ by virtue of the second condition in (9.221) — see the region described by the larger circle on the left in Figure 9.1. It follows that the two circular regions that we identified are disjoint from each other: one region is determined by the circle on the left that is centered at the origin with radius smaller than $1 - O(\mu_{\max})$, while the other region is determined by the circle on the right that is centered at one and has radius $O(\mu_{\max})$. The $2M$ discs that appear within this smaller circle are disjoint from the discs that appear inside the larger circle on the left. We conclude that $2M$ of the eigenvalues of $\mathcal{B}$ are located inside the discs in the rightmost circle. The eigenvalue that attains the spectral radius of $\mathcal{B}$ occurs inside this region so that

$$\rho(\mathcal{B}) \;=\; \rho(B) + O\left(\mu_{\max}^{(N+1)/N}\right) \tag{9.224}$$

Since it is assumed that $\mu_{\max} \ll 1$, and by referring back to expression (9.195) for D_{11}, we have

$$\rho(B) \;=\; \rho(I_{2M} - D_{11}^{\mathsf{T}}) \;=\; 1 - \lambda_{\min}\left(\sum_{k=1}^{N} q_k H_k\right) \tag{9.225}$$

Combining this relation with (9.224), we arrive at (9.173).

□

Size of Entries of $\mathcal{B}$

We can further exploit the structure revealed by expression (9.205) for $\mathcal{B}$ to examine the size of the entries of $(I-\mathcal{B})^{-1}$. In our derivations, the matrix $\mathcal{B}$ also appears transformed under the similarity transformation:

$$\bar{\mathcal{B}} \;\triangleq\; \mathcal{V}_\epsilon^{\mathsf{T}} \mathcal{B} \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} \overset{(9.206)}{=} \begin{bmatrix} I_{2M} - D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - D_{22}^{\mathsf{T}} \end{bmatrix} \tag{9.226}$$

where, according to (9.204),

$$\mathcal{V}_\epsilon \;\triangleq\; V_\epsilon \otimes I_{hM} \tag{9.227}$$

We therefore examine both matrices. The following result clarifies the size of the entries of $(I-\mathcal{B})^{-1}$ and $(I-\bar{\mathcal{B}})^{-1}$.

Lemma 9.4 (Similarity transformation)**.** Assume the matrix P is primitive. It holds that for sufficiently small step-sizes:

$$(I-\mathcal{B})^{-1} = O(1/\mu_{\max}) \tag{9.228}$$

$$(I-\bar{\mathcal{B}})^{-1} = \left[\begin{array}{c|c} O(1/\mu_{\max}) & O(1) \\ \hline O(1) & O(1) \end{array}\right] \tag{9.229}$$

where the leading $(1,1)$ block in $(I-\bar{\mathcal{B}})^{-1}$ has dimensions $hM \times hM$.

Proof. We carry out the derivation for the complex case $h=2$ without loss of generality following arguments similar to [69, 278]. We first remark that, by similarity, the matrix $\bar{\mathcal{B}}$ is stable by Theorem 9.3. Let

$$\begin{aligned} \mathcal{X} = I-\bar{\mathcal{B}} &= \left[\begin{array}{cc} D_{11}^{\mathsf{T}} & D_{21}^{\mathsf{T}} \\ D_{12}^{\mathsf{T}} & I-\mathcal{J}_{\epsilon}^{\mathsf{T}}+D_{22}^{\mathsf{T}} \end{array}\right] \\ &\triangleq \left[\begin{array}{cc} \mathcal{X}_{11} & \mathcal{X}_{12} \\ \mathcal{X}_{21} & \mathcal{X}_{22} \end{array}\right] \end{aligned} \tag{9.230}$$

where, from (9.199)–(9.202),

$$\mathcal{X}_{11} = O(\mu_{\max}) \tag{9.231}$$

$$\mathcal{X}_{12} = O(\mu_{\max}) \tag{9.232}$$

$$\mathcal{X}_{21} = O(\mu_{\max}) \tag{9.233}$$

$$\mathcal{X}_{22} = O(1) \tag{9.234}$$

The matrix $\mathcal{X}$ is invertible since $I-\bar{\mathcal{B}}$ is invertible. Moreover, $\mathcal{X}_{11}$ is invertible since $D_{11}>0$. We now appeal to the useful block matrix inversion formula [113, 206]:

$$\left[\begin{array}{cc} A & B \\ C & D \end{array}\right]^{-1} = \left[\begin{array}{cc} A^{-1} & 0 \\ 0 & 0 \end{array}\right] + \left[\begin{array}{cc} A^{-1}B\Delta^{-1}CA^{-1} & -A^{-1}B\Delta^{-1} \\ -\Delta^{-1}CA^{-1} & \Delta^{-1} \end{array}\right] \tag{9.235}$$

for matrices $\{A,B,C,D\}$ of compatible dimensions with invertible A and invertible Schur complement Δ defined by

$$\Delta = D - CA^{-1}B \tag{9.236}$$

Using this formula we can write

$$\mathcal{X}^{-1} = \left[\begin{array}{cc} \mathcal{X}_{11}^{-1}+\mathcal{X}_{11}^{-1}\mathcal{X}_{12}\Delta^{-1}\mathcal{X}_{21}\mathcal{X}_{11}^{-1} & -\mathcal{X}_{11}^{-1}\mathcal{X}_{12}\Delta^{-1} \\ -\Delta^{-1}\mathcal{X}_{21}\mathcal{X}_{11}^{-1} & \Delta^{-1} \end{array}\right] \tag{9.237}$$

where Δ denotes the Schur complement of $\mathcal{X}$ relative to $\mathcal{X}_{11}$:

$$\Delta \triangleq \mathcal{X}_{22} - \mathcal{X}_{21}\mathcal{X}_{11}^{-1}\mathcal{X}_{12} = O(1) \tag{9.238}$$

We then use (9.231)–(9.234) and (9.238) to deduce that

$$\mathcal{X}^{-1} = \begin{bmatrix} O(1/\mu_{\max}) & O(1) \\ O(1) & O(1) \end{bmatrix} \tag{9.239}$$

as claimed.

□

Low-Rank Approximation

We can establish similar results for the matrix

$$\mathcal{F} \triangleq \mathcal{B}^{\mathsf{T}} \otimes_b \mathcal{B}^{*} \tag{9.240}$$

which is defined in terms of the block Kronecker product operation using blocks of size $hM \times hM$, where $h = 1$ for real data and $h = 2$ for complex data. The matrix $\mathcal{F}$ will play a critical role in characterizing the performance and convergence rate of distributed algorithms, as will be revealed by future Theorem 11.2. In our derivations, the matrix $\mathcal{F}$ will also sometimes appear transformed under the similarity transformation:

$$\bar{\mathcal{F}} \triangleq (\mathcal{V}_\epsilon \otimes_b \mathcal{V}_\epsilon)^{-1}\, \mathcal{F}\, (\mathcal{V}_\epsilon \otimes_b \mathcal{V}_\epsilon) \tag{9.241}$$

Lemma 9.5 (Low-rank approximation). Assume the matrix P is primitive. For sufficiently small step-sizes, it holds that

$$(I - \mathcal{F})^{-1} = O(1/\mu_{\max}) \tag{9.242}$$

$$(I - \bar{\mathcal{F}})^{-1} = \left[\begin{array}{c|c} O(1/\mu_{\max}) & O(1) \\ \hline O(1) & O(1) \end{array}\right] \tag{9.243}$$

where the leading $(hM)^2 \times (hM)^2$ block in $(I - \bar{\mathcal{F}})^{-1}$ is $O(1/\mu_{\max})$. Moreover, we can also write

$$(I - \mathcal{F})^{-1} = \left[(p \otimes p)(\mathbb{1} \otimes \mathbb{1})^{\mathsf{T}}\right] \otimes Z^{-1} + O(1) \tag{9.244}$$

in terms of the regular Kronecker product operation, where the matrix Z has dimensions $(hM)^2 \times (hM)^2$ and consists of blocks of size $hM \times hM$ each:

$$Z \triangleq \sum_{k=1}^{N} q_k \left[(I_{hM} \otimes H_k) + (H_k^{\mathsf{T}} \otimes I_{hM}) \right] \tag{9.245}$$

where the vectors $\{p, q\}$ were defined earlier by (9.7)–(9.9). In addition, $Z = O(\mu_{\max})$.

Proof. We again carry out the derivation for the complex case $h = 2$ without loss of generality by extending an argument from [278] to the current context. We recall from (9.170) the expression for $\mathcal{B}$:

$$\mathcal{B} \;=\; \mathcal{P}^{\mathsf{T}} - \mathcal{A}_2^{\mathsf{T}} \mathcal{M} \mathcal{R} \mathcal{A}_1^{\mathsf{T}} \;=\; \mathcal{A}_2^{\mathsf{T}} \left(\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\mathcal{H} \right) \mathcal{A}_1^{\mathsf{T}} \tag{9.246}$$

where $\mathcal{P} = P \otimes I_{2M}$ and $P = A_1 A_o A_2$. Since the matrices $\{\mathcal{A}_o, \mathcal{A}_1, \mathcal{A}_2, \mathcal{M}\}$ are real-valued, and $\mathcal{H}$ is Hermitian, we have

$$\mathcal{B}^{\mathsf{T}} \;=\; \mathcal{A}_1(\mathcal{A}_o - \mathcal{H}^{\mathsf{T}}\mathcal{M})\mathcal{A}_2 \tag{9.247}$$

$$\mathcal{B}^{*} \;=\; \mathcal{A}_1(\mathcal{A}_o - \mathcal{H}\mathcal{M})\mathcal{A}_2 \tag{9.248}$$

We introduce the same Jordan canonical decomposition (9.21)–(9.24) and verify, in a manner similar to (9.53), that

$$\mathcal{B}^{*} \;=\; (V_\epsilon \otimes I_{2M}) \begin{bmatrix} I_{2M} - E_{11} & -E_{12} \\ -E_{21} & (J_\epsilon \otimes I_{2M}) - E_{22} \end{bmatrix} (V_\epsilon^{-1} \otimes I_{2M}) \tag{9.249}$$

where the block matrices $\{E_{mn}\}$ are given by

$$E_{11} \;=\; \sum_{k=1}^{N} q_k H_k \;=\; O(\mu_{\max}) \tag{9.250}$$

$$E_{12} \;=\; (\mathbb{1}^{\mathsf{T}} \otimes I_{2M})\mathcal{H}\mathcal{M}(A_2 V_R \otimes I_{2M}) \;=\; O(\mu_{\max}) \tag{9.251}$$

$$E_{21} \;=\; (V_L^{\mathsf{T}} A_1 \otimes I_{2M})\mathcal{H}(q \otimes I_{2M}) \;=\; O(\mu_{\max}) \tag{9.252}$$

$$E_{22} \;=\; (V_L^{\mathsf{T}} A_1 \otimes I_{2M})\mathcal{H}\mathcal{M}(A_2 V_R \otimes I_{2M}) \;=\; O(\mu_{\max}) \tag{9.253}$$

and their entries are in the order of $\mu_{\max}$; this fact can be verified in the same manner that we assessed the size of the block matrices $\{\boldsymbol{D}_{11,i-1}, \boldsymbol{D}_{12,i-1}, \boldsymbol{D}_{21,i-1}, \boldsymbol{D}_{22,i-1}\}$ in the proof of the earlier Theorem 9.1. Moreover, the dimensions of E_{11} are $2M \times 2M$.

In a similar manner, we find that

$$\mathcal{B}^{\mathsf{T}} \;=\; (V_\epsilon \otimes I_{2M}) \begin{bmatrix} I_{2M} - D_{11} & -D_{12} \\ -D_{21} & (J_\epsilon \otimes I_{2M}) - D_{22} \end{bmatrix} (V_\epsilon^{-1} \otimes I_{2M}) \tag{9.254}$$

where the block matrices $\{D_{mn}\}$ are given by

$$D_{11} = \sum_{k=1}^{N} q_k H_k^{\mathsf{T}} = O(\mu_{\max}) \tag{9.255}$$

$$D_{12} = (\mathbb{1}^{\mathsf{T}} \otimes I_{2M})\mathcal{H}^{\mathsf{T}}\mathcal{M}(A_2 V_R \otimes I_{2M}) = O(\mu_{\max}) \tag{9.256}$$

$$D_{21} = (V_L^{\mathsf{T}} A_1 \otimes I_{2M})\mathcal{H}^{\mathsf{T}}(q \otimes I_{2M}) = O(\mu_{\max}) \tag{9.257}$$

$$D_{22} = (V_L^{\mathsf{T}} A_1 \otimes I_{2M})\mathcal{H}^{\mathsf{T}}\mathcal{M}(A_2 V_R \otimes I_{2M}) = O(\mu_{\max}) \tag{9.258}$$

and D_{11} has dimensions $2M \times 2M$. Substituting expressions (9.249) and (9.254) into (9.240), and using the second property for block Kronecker products from Table F.2 in the appendix, we obtain

$$\mathcal{F} = (\mathcal{V}_\epsilon \otimes_b \mathcal{V}_\epsilon)\,\mathcal{X}\,(\mathcal{V}_\epsilon \otimes_b \mathcal{V}_\epsilon)^{-1} \tag{9.259}$$

where the block Kronecker product operation is relative to blocks of size $2M \times 2M$, and where we introduced

$$\mathcal{X} \triangleq \begin{bmatrix} I_{2M} - D_{11} & -D_{12} \\ -D_{21} & (J_\epsilon \otimes I_{2M}) - D_{22} \end{bmatrix} \otimes_b \begin{bmatrix} I_{2M} - E_{11} & -E_{12} \\ -E_{21} & (J_\epsilon \otimes I_{2M}) - E_{22} \end{bmatrix} \tag{9.260}$$

We conclude that

$$(I - \mathcal{F})^{-1} = (\mathcal{V}_\epsilon \otimes_b \mathcal{V}_\epsilon)\,(I - \mathcal{X})^{-1}\,(\mathcal{V}_\epsilon \otimes_b \mathcal{V}_\epsilon)^{-1} \tag{9.261}$$

We partition $\mathcal{X}$ into the following block structure:

$$\mathcal{X} = \begin{bmatrix} \mathcal{X}_{11} & \mathcal{X}_{12} \\ \mathcal{X}_{21} & \mathcal{X}_{22} \end{bmatrix} \tag{9.262}$$

where, for example, $\mathcal{X}_{11}$ is $(2M)^2 \times (2M)^2$ and is given by

$$\mathcal{X}_{11} = (I_{2M} - D_{11}) \otimes (I_{2M} - E_{11}) \tag{9.263}$$

It follows that

$$I - \mathcal{X} = \begin{bmatrix} I_{(2M)^2} - \mathcal{X}_{11} & -\mathcal{X}_{12} \\ -\mathcal{X}_{21} & I - \mathcal{X}_{22} \end{bmatrix} \tag{9.264}$$

and, in a manner similar to the way we assessed the size of the block matrices $\{\boldsymbol{D}_{11,i-1}, \boldsymbol{D}_{12,i-1}, \boldsymbol{D}_{21,i-1}, \boldsymbol{D}_{22,i-1}\}$ in the proof of Theorem 9.1, we can likewise verify that

$$I_{(2M)^2} - \mathcal{X}_{11} = O(\mu_{\max}) \tag{9.265}$$

$$\mathcal{X}_{12} = O(\mu_{\max}) \tag{9.266}$$

$$\mathcal{X}_{21} = O(\mu_{\max}) \tag{9.267}$$

$$I - \mathcal{X}_{22} = O(1) \tag{9.268}$$

In particular, note that

$$\begin{aligned} I_{(2M)^2} - \mathcal{X}_{11} &= I_{(2M)^2} - (I_{2M} - D_{11}) \otimes (I_{2M} - E_{11}) \\ &= (I_{2M} \otimes E_{11}) + (D_{11} \otimes I_{2M}) - (D_{11} \otimes E_{11}) \\ &= O(\mu_{\max}) \end{aligned} \tag{9.269}$$

and

$$\begin{aligned} I - \mathcal{X}_{22} &= I - ((J_\epsilon \otimes I_{2M}) - D_{22}) \otimes_b ((J_\epsilon \otimes I_{2M}) - E_{22}) \\ &= I - (J_\epsilon \otimes I_{2M}) \otimes_b (J_\epsilon \otimes I_{2M}) + O(\mu_{\max}) \\ &= O(1) \end{aligned} \tag{9.270}$$

To proceed, we call again upon the useful block matrix inversion formula (9.235). The matrix $I - \mathcal{X}$ is invertible since $I - \mathcal{F}$ is invertible; this is because $\rho(\mathcal{F}) = [\rho(\mathcal{B})]^2 < 1$. Therefore, applying (9.235) to $I - \mathcal{X}$ we get

$$\begin{aligned} (I - \mathcal{X})^{-1} &= \begin{bmatrix} (I_{(2M)^2} - \mathcal{X}_{11})^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \\ &\begin{bmatrix} (I - \mathcal{X}_{11})^{-1}\mathcal{X}_{12}\Delta^{-1}\mathcal{X}_{21}(I - \mathcal{X}_{11})^{-1} & (I - \mathcal{X}_{11})^{-1}\mathcal{X}_{12}\Delta^{-1} \\ \Delta^{-1}\mathcal{X}_{21}(I - \mathcal{X}_{11})^{-1} & \Delta^{-1} \end{bmatrix} \end{aligned} \tag{9.271}$$

It is seen from (9.269) that the entries of $(I - \mathcal{X}_{11})^{-1}$ are $O(1/\mu_{\max})$, while the entries in the second matrix on the right-hand side of equality (9.271) are $O(1)$ when the step-sizes are small. That is, we can write

$$(I - \mathcal{X})^{-1} = \left[\begin{array}{c|c} O(1/\mu_{\max}) & O(1) \\ \hline O(1) & O(1) \end{array}\right] \tag{9.272}$$

where the leading $(2M)^2 \times (2M)^2$ block is $O(1/\mu_{\max})$. Moreover, since $O(1/\mu_{\max})$ dominates $O(1)$ for sufficiently small $\mu_{\max}$, we can also write

$$\begin{aligned} (I - \mathcal{X})^{-1} &= \begin{bmatrix} (I_{(2M)^2} - \mathcal{X}_{11})^{-1} & 0 \\ 0 & 0 \end{bmatrix} + O(1) \\ &= \begin{bmatrix} \{(I_{2M} \otimes E_{11}) + (D_{11} \otimes I_{2M})\}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + O(1) \\ &= \begin{bmatrix} I_{(2M)^2} \\ 0 \end{bmatrix} Z^{-1} \begin{bmatrix} I_{(2M)^2} & 0 \end{bmatrix} + O(1) \end{aligned} \tag{9.273}$$

where we used the fact from (9.245) that, for $h = 2$,

$$Z = (I_{2M} \otimes E_{11}) + (D_{11} \otimes I_{2M}) \tag{9.274}$$

Substituting (9.273) into (9.261) and using expressions (9.250) and (9.255) for D_{11} and E_{11} we arrive at the following low-rank approximation:

$$\begin{aligned}
&(I-\mathcal{F})^{-1} \\
&= (p\otimes I_{2M})\otimes_b (p\otimes I_{2M})\, Z^{-1}\left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right)\otimes_b\left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right)+O(1) \\
&\overset{(a)}{=} [(p\otimes p)\otimes (I_{2M}\otimes I_{2M})]\,(1\otimes Z^{-1})\left[(\mathbb{1}\otimes\mathbb{1})^{\mathsf{T}}\otimes (I_{2M}\otimes I_{2M})\right]+O(1) \\
&= [(p\otimes p)\otimes I_{4M^2}]\,(1\otimes Z^{-1})\left[(\mathbb{1}\otimes\mathbb{1})^{\mathsf{T}}\otimes I_{4M^2}\right] \;+\; O(1) \\
&= \left[(p\otimes p)\otimes Z^{-1}\right]\left[(\mathbb{1}\otimes\mathbb{1})^{\mathsf{T}}\otimes I_{4M^2}\right] \;+\; O(1) \\
&= \left[(p\otimes p)(\mathbb{1}\otimes\mathbb{1})^{\mathsf{T}}\right]\otimes Z^{-1}+O(1)
\end{aligned} \tag{9.275}$$

where step (a) uses the third property from Table F.2 in the appendix. Observe that the matrix $(p\otimes p)(\mathbb{1}\otimes\mathbb{1})^{\mathsf{T}}$ has rank one and, therefore, the above representation for $(I-\mathcal{F})^{-1}$) amounts to a low-rank approximation. Moreover, since $Z=O(\mu_{\max})$, we conclude from (9.275) that (9.243) holds. We also conclude that (9.242) holds since

$$(I-\bar{\mathcal{F}})^{-1} = (\mathcal{V}_\epsilon\otimes_b\mathcal{V}_\epsilon)^{-1}\,(I-\mathcal{F})^{-1}\,(\mathcal{V}_\epsilon\otimes_b\mathcal{V}_\epsilon) \;=\; (I-\mathcal{X})^{-1} \tag{9.276}$$

□

Mean-Error Stability

We now return to examine the mean-error stability of recursion (9.171). For this purpose, we need to introduce a smoothness condition on the Hessian matrices of the individual costs. This condition was not needed while establishing the stability of the second and fourth-order moments, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ and $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^4$, in the earlier sections. This same smoothness condition will be adopted in the next two chapters when we study the long-term behavior of the network and its performance.

Theorem 9.6 (Network mean-error stability)**.** Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P=A_1A_oA_2$. Assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumption 6.1. Assume additionally that each $J_k(w)$ satisfies a smoothness condition relative to the limit point $w^\star$, defined by (8.55), of the following form:

$$\left\|\nabla_w^2\, J_k(w^\star+\Delta w)-\nabla_w^2\, J_k(w^\star)\right\| \;\le\; \kappa_d\,\|\Delta w\| \tag{9.277}$$

for small perturbations $\|\Delta w\|\le\epsilon$ and for some $\kappa_d\ge 0$. Assume further that the first and second-order moments of the gradient noise process satisfy the

conditions of Assumption 8.1. Then, the first-order moment of the network errors satisfy

$$\limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_{k,i}\| \;=\; O(\mu_{\max}), \quad k=1,2,\ldots,N \tag{9.278}$$

Proof. We multiply both sides of the error recursion (9.171) from the left by $\mathcal{V}_\epsilon^{\mathsf{T}}$ and use (9.57) and (9.206) to get

$$\underbrace{\begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_i^e \\ \mathbb{E}\,\check{\boldsymbol{w}}_i^e \end{bmatrix}}_{\triangleq\, z_i} = \underbrace{\begin{bmatrix} I_{2M}-D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}}-D_{22}^{\mathsf{T}} \end{bmatrix}}_{\triangleq\,\bar{\mathcal{B}}} \underbrace{\begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_{i-1}^e \\ \mathbb{E}\,\check{\boldsymbol{w}}_{i-1}^e \end{bmatrix}}_{\triangleq\, z_{i-1}} - \begin{bmatrix} 0 \\ \check{b}^e \end{bmatrix} + \mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}c_{i-1} \tag{9.279}$$

where the matrix $\bar{\mathcal{B}}$ from (9.226) is stable. We already know from (9.59) that $\|\check{b}^e\| = O(\mu_{\max})$. We now verify that the limit superior of $\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}c_{i-1}\|$ is $O(\mu_{\max}^2)$.

Indeed, in view of result (E.61) from the appendix, we know that condition (9.277) also holds globally for any Δw with κ_d replaced by some constant κ_d'. Then, for each agent k:

$$\begin{aligned}
\|\widetilde{\boldsymbol{H}}_{k,i-1}\| &\triangleq \|H-\boldsymbol{H}_{k,i-1}\| \\
&\le \int_0^1 \Big\|\nabla_w^2 J_k(w^\star) \;-\; \nabla_w^2\, J_k(w^\star - t\widetilde{\boldsymbol{\phi}}_{k,i-1})\Big\|\,dt \\
&\overset{(9.277)}{\le} \int_0^1 \kappa_d'\|t\widetilde{\boldsymbol{\phi}}_{k,i-1}\|dt \\
&= \frac{1}{2}\kappa_d'\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\| \\
&\le \frac{1}{2}\kappa_d'\left\|\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\widetilde{\boldsymbol{w}}_{\ell,i-1}\right\| \\
&\overset{(F.26)}{\le} \frac{1}{2}\kappa_d'\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\| \\
&\le \frac{1}{2}\kappa_d'\sum_{\ell\in\mathcal{N}_k} \|\widetilde{\boldsymbol{w}}_{\ell,i-1}\| \\
&\le \frac{1}{2}\kappa_d'\sum_{\ell\in\mathcal{N}_k} \|\widetilde{\boldsymbol{w}}_{\ell,i-1}^e\| \\
&\le \frac{1}{2}\kappa_d' N\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|
\end{aligned} \tag{9.280}$$

so that

$$\|\widetilde{\boldsymbol{\mathcal{H}}}_{i-1}\| \;=\; \max_{1\le k\le N}\|\widetilde{\boldsymbol{H}}_{k,i-1}\| \;\le\; \frac{1}{2}\kappa_d' N\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\| \tag{9.281}$$

and, consequently,

$$\begin{aligned}\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}c_{i-1}\| &\overset{(9.172)}{\leq} \|\mathcal{V}_\epsilon\|\,\|\mathcal{A}_2\|\,\|\mathcal{M}\|\,\|\mathbb{E}\,\widetilde{\boldsymbol{\mathcal{H}}}_{i-1}\,\mathcal{A}_1^{\mathsf{T}}\,\widetilde{\boldsymbol{w}}_{i-1}^e\| \\ &\leq \|\mathcal{V}_\epsilon\|\,\|\mathcal{A}_2\|\,\|\mathcal{M}\|\,\|\mathcal{A}_1\|\,\mathbb{E}\,\left[\|\widetilde{\boldsymbol{\mathcal{H}}}_{i-1}\|\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|\right] \\ &\leq \frac{1}{2}\kappa_d' N\|\mathcal{V}_\epsilon\|\,\|\mathcal{A}_2\|\,\|\mathcal{M}\|\,\|\mathcal{A}_1\|\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^2 \\ &\triangleq r\mu_{\max}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^2 \end{aligned} \tag{9.282}$$

for some constant r that is independent of $\mu_{\max}$. It then follows from (9.11) that

$$\limsup_{i\to\infty}\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}c_{i-1}\| \;=\; O(\mu_{\max}^2) \tag{9.283}$$

as claimed, where one $\mu_{\max}$ arises from $\mathcal{M}$ and the other $\mu_{\max}$ arises from (9.11).

Returning to (9.279), we partition the vectors z_i and $\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}c_{i-1}$ into

$$z_i \;\triangleq\; \begin{bmatrix}\bar{z}_i\\ \check{z}_i\end{bmatrix}, \qquad \mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}c_{i-1} \;\triangleq\; \begin{bmatrix}\bar{c}_{i-1}\\ \check{c}_{i-1}\end{bmatrix} \tag{9.284}$$

with the leading vectors, $\{\bar{z}_i, \bar{c}_{i-1}\}$, having dimensions $hM\times 1$ each. It follows that

$$\begin{bmatrix}\bar{z}_i\\ \check{z}_i\end{bmatrix} = \begin{bmatrix} I_{2M}-D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}}-D_{22}^{\mathsf{T}}\end{bmatrix}\begin{bmatrix}\bar{z}_{i-1}\\ \check{z}_{i-1}\end{bmatrix} + \begin{bmatrix}\bar{c}_{i-1}\\ \check{c}_{i-1}\end{bmatrix} + \begin{bmatrix}0\\ O(\mu_{\max})\end{bmatrix} \tag{9.285}$$

This recursion has a form similar to the earlier recursion we encountered in (9.60) while studying the mean-square stability of the original error dynamics (10.2), with two differences. First, the matrices $\{D_{11}, D_{12}, D_{21}, D_{22}\}$ in (9.285) are constant matrices; nevertheless, they satisfy the same bounds as the matrices $\{\boldsymbol{D}_{11,i-1}, \boldsymbol{D}_{12,i-1}, \boldsymbol{D}_{21,i-1}, \boldsymbol{D}_{22,i-1}\}$ in (9.60). In particular, it continues to hold that

$$\|I_{2M}-D_{11}^{\mathsf{T}}\| \overset{(9.47)}{\leq} 1-\sigma_{11}\mu_{\max} \tag{9.286}$$

$$\|D_{12}\| \overset{(9.51)}{\leq} \sigma_{12}\mu_{\max} \tag{9.287}$$

$$\|D_{21}\| \overset{(9.50)}{\leq} \sigma_{21}\mu_{\max} \tag{9.288}$$

$$\|D_{22}\| \overset{(9.51)}{\leq} \sigma_{22}\mu_{\max} \tag{9.289}$$

for some positive constants $\{\sigma_{11}, \sigma_{12}, \sigma_{21}, \sigma_{22}\}$ that are independent of $\mu_{\max}$. Second, the gradient noise terms that appeared in (9.60) are now replaced by

the deterministic sequences $\{\bar{c}_{i-1}, \check{c}_{i-1}\}$. However, from (9.282) and using the fact that $(\mathbb{E}\,\boldsymbol{a})^2 \leq \mathbb{E}\,\boldsymbol{a}^2$ for any real random variable $\boldsymbol{a}$, we have that

$$\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}c_{i-1}\|^2 \leq r^2\mu_{\max}^2\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 \tag{9.290}$$

and, hence,

$$\|\bar{c}_{i-1}\|^2 \leq r^2\mu_{\max}^2\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4, \qquad \|\check{c}_{i-1}\|^2 \leq r^2\mu_{\max}^2\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 \tag{9.291}$$

Now, if we repeat the argument that led to (9.106), with proper adjustments, we can show that relations similar to (9.69) and (9.81) continue to hold for $\{\|\bar{z}_i\|^2, \|\check{z}_i\|^2\}$. The argument is as follows.

We first appeal to Jensen's inequality (F.26) from the appendix and apply it to the function $f(x) = \|x\|^2$ to obtain the bound:

$$\begin{aligned}
\|\bar{z}_i\|^2 &= \left\|(1-t)\frac{1}{1-t}(I_{2M} - D_{11}^{\mathsf{T}})\bar{z}_{i-1} + t\frac{1}{t}\left(-D_{21}^{\mathsf{T}}\check{z}_{i-1} + \bar{c}_{i-1}\right)\right\|^2 \\
&\leq \frac{1}{1-t}(1-\sigma_{11}\mu_{\max})^2\|\bar{z}_{i-1}\|^2 + \frac{2}{t}\left(\sigma_{21}^2\mu_{\max}^2\|\check{z}_{i-1}\|^2 + \|\bar{c}_{i-1}\|^2\right) \\
&\leq (1-\sigma_{11}\mu_{\max})\|\bar{z}_{i-1}\|^2 + \frac{2}{\sigma_{11}\mu_{\max}}\left(\sigma_{21}^2\mu_{\max}^2\|\check{z}_{i-1}\|^2 + \|\bar{c}_{i-1}\|^2\right) \\
&\leq (1-\sigma_{11}\mu_{\max})\|\bar{z}_{i-1}\|^2 + \frac{2\sigma_{21}^2\mu_{\max}}{\sigma_{11}}\|\check{z}_{i-1}\|^2 + \frac{2r^2\mu_{\max}}{\sigma_{11}}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4
\end{aligned} \tag{9.292}$$

for any arbitrary positive number $t \in (0,1)$. We selected $t = \sigma_{11}\mu_{\max}$ in the above derivation. We repeat a similar argument for $\|\check{z}_i\|^2$. Thus, using Jensen's inequality again we have

$$\begin{aligned}
\|\check{z}_i\|^2 &= \left\|t\frac{1}{t}\mathcal{J}_\epsilon^{\mathsf{T}}\check{z}_{i-1} - (1-t)\frac{1}{1-t}\left[-D_{22}^{\mathsf{T}}\check{z}_{i-1} - D_{12}^{\mathsf{T}}\bar{z}_{i-1} + \check{c}_{i-1} + O(\mu_{\max})\right]\right\|^2 \\
&\overset{(9.76)}{\leq} \frac{1}{t}(\rho(J_\epsilon)+\epsilon)^2\,\|\check{z}_{i-1}\|^2 + \\
&\qquad \frac{4}{1-t}\left[\sigma_{22}^2\mu_{\max}^2\|\check{z}_{i-1}\|^2 + \sigma_{12}^2\mu_{\max}^2\|\bar{z}_{i-1}\|^2 + \|\check{c}_{i-1}\|^2 + O(\mu_{\max}^2)\right]
\end{aligned} \tag{9.293}$$

for any arbitrary positive number $t \in (0,1)$. Since we know that $\rho(J_\epsilon) \in (0,1)$, then we can select ϵ small enough to ensure $t = \rho(J_\epsilon) + \epsilon \in (0,1)$ and rewrite (9.293) as

$$\begin{aligned}\|\check{z}_i\|^2 \;\leq\; & \left(\rho(J_\epsilon)+\epsilon+\frac{4\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\|\check{z}_{i-1}\|^2 + \\ & \left(\frac{4\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\|\bar{z}_{i-1}\|^2 + \\ & \left(\frac{4r^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 + O(\mu_{\max}^2) \end{aligned} \tag{9.294}$$

If we now introduce the scalar coefficients

$$a \;=\; 1-\sigma_{11}\mu_{\max} \;=\; 1-O(\mu_{\max}) \tag{9.295}$$

$$b \;=\; \frac{2\sigma_{21}^2\mu_{\max}}{\sigma_{11}} \;=\; O(\mu_{\max}) \tag{9.296}$$

$$c \;=\; \frac{4\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon} \;=\; O(\mu_{\max}^2) \tag{9.297}$$

$$d \;=\; \rho(J_\epsilon)+\epsilon+\frac{4\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon} \;=\; \rho(J_\epsilon)+\epsilon+O(\mu_{\max}^2) \tag{9.298}$$

$$e \;=\; \frac{2r^2\mu_{\max}}{\sigma_{11}} \;=\; O(\mu_{\max}) \tag{9.299}$$

$$f \;=\; \frac{4r^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon} \;=\; O(\mu_{\max}^2) \tag{9.300}$$

we can combine (9.292) and (9.294) into a single compact inequality recursion as follows:

$$\begin{bmatrix}\|\bar{z}_i\|^2\\ \|\check{z}_i\|^2\end{bmatrix} \preceq \underbrace{\begin{bmatrix}a & b\\ c & d\end{bmatrix}}_{\Gamma}\begin{bmatrix}\|\bar{z}_{i-1}\|^2\\ \|\check{z}_{i-1}\|^2\end{bmatrix} + \begin{bmatrix}e\\ f\end{bmatrix}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 + \begin{bmatrix}0\\ O(\mu_{\max}^2)\end{bmatrix} \tag{9.301}$$

in terms of the 2×2 coefficient matrix Γ indicated above. We know from the argument (9.102) that Γ is stable for sufficiently small step-sizes. If we now recall the result

$$\limsup_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^4 \overset{(9.107)}{=} O(\mu_{\max}^2) \tag{9.302}$$

and use (9.103) we conclude that, as $i\to\infty$,

$$\limsup_{i\to\infty}\;\|\bar{z}_i\|^2 \;=\; O(\mu_{\max}^2), \qquad \limsup_{i\to\infty}\;\mathbb{E}\,\|\check{z}_i\|^2 \;=\; O(\mu_{\max}^2) \tag{9.303}$$

and, hence,

$$\limsup_{i\to\infty}\;\|z_i\|^2 \;=\; O(\mu_{\max}^2) \tag{9.304}$$

It follows that

$$\limsup_{i\to\infty} \|z_i\| \;=\; O(\mu_{\max}) \tag{9.305}$$

Consequently,

$$\limsup_{i\to\infty} \left\| \begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_i^e \\ \mathbb{E}\,\check{\boldsymbol{w}}_i^e \end{bmatrix} \right\| \;=\; O(\mu_{\max}) \tag{9.306}$$

and, hence,

$$\begin{aligned} \limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_{k,i}\| &\leq \limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i^e\| \\ &= \limsup_{i\to\infty} \left\| \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} \begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_i^e \\ \mathbb{E}\,\check{\boldsymbol{w}}_i^e \end{bmatrix} \right\| \\ &\leq \left\| \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} \right\| \left(\limsup_{i\to\infty} \left\| \begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_i^e \\ \mathbb{E}\,\check{\boldsymbol{w}}_i^e \end{bmatrix} \right\| \right) \\ &= O(\mu_{\max}) \end{aligned} \tag{9.307}$$

as claimed.

□

10

Long-Term Network Dynamics

We move on to motivate a long-term model for the evolution of the network error dynamics, $\widetilde{\boldsymbol{w}}_i^e$, after sufficient iterations have passed, i.e., for $i \gg 1$. We examine the stability property of the model, the proximity of its trajectory from that of the original network dynamics, and subsequently employ the model to assess network MSD and ER performance metrics. To do so, we will need to recall the same smoothness condition used in establishing the mean-stability result of Theorem 9.6.

Assumption 10.1. (Smoothness condition on individual cost functions). It is assumed that each $J_k(w)$ satisfies a smoothness condition close to the limit point $w^\star$, defined by (8.55), in that the corresponding Hessian matrix is Lipschitz continuous in the proximity of $w^\star$ with some parameter $\kappa_d \geq 0$, i.e.,

$$\left\|\nabla_w^2 J_k(w^\star + \Delta w) - \nabla_w^2 J_k(w^\star)\right\| \;\leq\; \kappa_d \, \|\Delta w\| \tag{10.1}$$

for small perturbations $\|\Delta w\| \leq \epsilon$.

10.1 Long-Term Error Model

We reconsider the network error recursion (9.12), namely,

$$\widetilde{\boldsymbol{w}}_i^e \;=\; \boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^e \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e, \;\; i\geq 0 \tag{10.2}$$

where, from the expressions in Lemma 8.1,

$$\boldsymbol{\mathcal{B}}_{i-1} \;=\; \mathcal{P}^{\mathsf{T}} \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{\mathcal{H}}_{i-1}\mathcal{A}_1^{\mathsf{T}} \tag{10.3}$$

$$\mathcal{P}^{\mathsf{T}} \;=\; \mathcal{A}_2^{\mathsf{T}}\mathcal{A}_o^{\mathsf{T}}\mathcal{A}_1^{\mathsf{T}} \tag{10.4}$$

$$\boldsymbol{\mathcal{H}}_{i-1} \;\triangleq\; \mathrm{diag}\{\,\boldsymbol{H}_{1,i-1},\;\boldsymbol{H}_{2,i-1},\;\ldots,\;\boldsymbol{H}_{N,i-1}\,\} \tag{10.5}$$

$$\boldsymbol{H}_{k,i-1} \;\triangleq\; \int_0^1 \nabla_w^2 J_k(w^\star - t\widetilde{\boldsymbol{\phi}}_{k,i-1})dt \tag{10.6}$$

We again introduce the error matrix:

$$\widetilde{\boldsymbol{\mathcal{H}}}_{i-1} \;\triangleq\; \mathcal{H} \;-\; \boldsymbol{\mathcal{H}}_{i-1} \tag{10.7}$$

which measures the deviation of $\boldsymbol{\mathcal{H}}_{i-1}$ from the constant matrix:

$$\mathcal{H} \;\triangleq\; \mathrm{diag}\{\,H_1,\;H_2,\;\ldots,\;H_N\,\} \tag{10.8}$$

with each H_k given by the value of the Hessian matrix at the limit point, namely,

$$H_k \;\triangleq\; \nabla_w^2\, J_k(w^\star) \tag{10.9}$$

Then, using (9.166) in the expression for $\boldsymbol{\mathcal{B}}_{i-1}$, we can write

$$\boldsymbol{\mathcal{B}}_{i-1} \;=\; \mathcal{B} \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\widetilde{\boldsymbol{\mathcal{H}}}_{i-1}\mathcal{A}_1^{\mathsf{T}} \tag{10.10}$$

in terms of the constant coefficient matrix

$$\mathcal{B} \;\triangleq\; \mathcal{P}^{\mathsf{T}} - \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\mathcal{H}\mathcal{A}_1^{\mathsf{T}} \tag{10.11}$$

We established in Theorem 9.3 that, for sufficiently small step-sizes, the matrix $\mathcal{B}$ is stable and its spectral radius is given by

$$\rho(\mathcal{B}) \;=\; 1 - \lambda_{\min}\left(\sum_{k=1}^{N} q_k H_k\right) \;+\; O\left(\mu_{\max}^{(N+1)/N}\right) \tag{10.12}$$

where $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue of its Hermitian matrix argument. Now, using (10.10), we can rewrite error recursion (10.2) as

$$\widetilde{\boldsymbol{w}}_i^e \;=\; \mathcal{B}\,\widetilde{\boldsymbol{w}}_{i-1}^e \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1} \tag{10.13}$$

in terms of the random perturbation sequence:

$$\boldsymbol{c}_{i-1} \triangleq \widetilde{\boldsymbol{\mathcal{H}}}_{i-1}\mathcal{A}_1^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1}^e \tag{10.14}$$

By exploiting the smoothness condition (10.1), and following an argument similar to (9.280)–(9.283), we can verify that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\boldsymbol{c}_{i-1}\| = O(\mu_{\max}) \tag{10.15}$$

This is because

$$\begin{aligned}
\limsup_{i\to\infty} \mathbb{E}\,\|\boldsymbol{c}_{i-1}\| &\overset{(10.14)}{\leq} \|\mathcal{A}_1\| \left(\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{\mathcal{H}}}_{i-1}\|\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|\right) \\
&\overset{(9.281)}{\leq} \frac{1}{2}\kappa_d' N\|\mathcal{A}_1\| \left(\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^2\right) \\
&\overset{(9.11)}{=} O(\mu_{\max})
\end{aligned} \tag{10.16}$$

Returning to (10.15), we deduce that $\|\boldsymbol{c}_{i-1}\| = O(\mu_{\max})$ asymptotically with *high probability* using the same argument that led to (4.53) in the single-agent case. Referring to recursion (10.13), this analysis suggests that we can assess the mean-square performance of the original error recursion (10.2) by considering instead the following long-term model, which holds with high probability after sufficient iterations:

$$\widetilde{\boldsymbol{w}}_i^e = \mathcal{B}\,\widetilde{\boldsymbol{w}}_{i-1}^e + \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) - \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e, \quad i \gg 1 \tag{10.17}$$

In this model, the perturbation term $\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1}$ that appears in (10.13) is removed. We may also consider an alternative long-term model where $\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1}$ is instead replaced by a constant driving term in the order of $O(\mu_{\max}^2)$. However, the conclusions that will follow about the performance of the original recursion (10.2) will be the same whether we remove $\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1}$ altogether or replace it by $O(\mu_{\max}^2)$. We therefore continue our analysis by using model (10.17). Obviously, the iterates $\{\widetilde{\boldsymbol{w}}_i^e\}$ that are generated by (10.17) are generally different from the iterates that are generated by the original recursion (10.2). To highlight this fact more accurately, we rewrite the long-term recursion (10.17) more explicitly as follows for $i \gg 1$:

$$\widetilde{\boldsymbol{w}}_i^{e'} = \mathcal{B}\,\widetilde{\boldsymbol{w}}_{i-1}^{e'} + \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) - \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \tag{10.18}$$

with the iterates now denoted by $\widetilde{\boldsymbol{w}}_i^{e'}$ using the prime notation for the state of the long-term model. Note that the driving process $\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)$ in (10.18) is the *same* gradient noise process from the original recursion (10.2) and is therefore evaluated at $\boldsymbol{w}_{i-1}^e$. It is instructive to compare the following statement with the earlier Lemma 8.1.

Lemma 10.1 (Long-term network dynamics). Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P = A_1 A_o A_2$. Assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumptions 6.1 and 10.1. After sufficient iterations, $i \gg 1$, the error dynamics of the network relative to the limit point $w^\star$ defined by (8.55) is well-approximated by the following model (as confirmed by future result (10.29)):

$$\widetilde{\boldsymbol{w}}_i^{e'} \;=\; \mathcal{B}\,\widetilde{\boldsymbol{w}}_{i-1}^{e'} \;+\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \tag{10.19}$$

where

$$\mathcal{B} \;\triangleq\; \mathcal{A}_2^{\mathsf{T}}\left(\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\mathcal{H}\right)\mathcal{A}_1^{\mathsf{T}} \tag{10.20}$$

$$\mathcal{A}_o \;\triangleq\; A_o \otimes I_{2M}, \quad \mathcal{A}_1 \;\triangleq\; A_1 \otimes I_{2M}, \quad \mathcal{A}_2 \;\triangleq\; A_2 \otimes I_{2M} \tag{10.21}$$

$$\mathcal{M} \;\triangleq\; \mathrm{diag}\{\,\mu_1 I_{2M},\; \mu_2 I_{2M},\; \ldots,\; \mu_N I_{2M}\,\} \tag{10.22}$$

$$\mathcal{H} \;\triangleq\; \mathrm{diag}\,\{\,H_1,\; H_2,\; \ldots,\; H_N\,\} \tag{10.23}$$

$$H_k \;\triangleq\; \nabla_w^2\, J_k(w^\star) \tag{10.24}$$

where $\nabla_w^2 J_k(w)$ denotes the $2M \times 2M$ Hessian matrix of $J_k(w)$ relative to w.

In a manner similar to the partitioning of $\widetilde{\boldsymbol{w}}_i^e$ into its constituent elements in (8.143), we partition $\widetilde{\boldsymbol{w}}_i^{e'}$ into its $2M \times 1$ block entries as follows:

$$\widetilde{\boldsymbol{w}}_i^{e'} \;\triangleq\; \begin{bmatrix} \widetilde{\boldsymbol{w}}_{1,i}^{e'} \\ \widetilde{\boldsymbol{w}}_{2,i}^{e'} \\ \vdots \\ \widetilde{\boldsymbol{w}}_{N,i}^{e'} \end{bmatrix} \tag{10.25}$$

with each $\widetilde{\boldsymbol{w}}_{k,i}^{e'}$ at every agent in turn consisting of

$$\widetilde{\boldsymbol{w}}_{k,i}^{e'} \;=\; \begin{bmatrix} \widetilde{\boldsymbol{w}}_{k,i}^{\prime} \\ \left(\widetilde{\boldsymbol{w}}_{k,i}^{\prime *}\right)^{\mathsf{T}} \end{bmatrix} \tag{10.26}$$

We can view the long-term model (10.19) as a dynamic recursion that is fed by the gradient noise sequence, $\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)$. Therefore, assuming both the original system (10.2) and the long-term model (10.19) are launched from the same initial conditions, we observe by iterating (10.19) that $\widetilde{\boldsymbol{w}}_i^{e'}$ will still be determined by the past history of the original iterates $\{\boldsymbol{w}_j,\, j \leq i-1\}$ through its dependence on the gradient noise process $\{\boldsymbol{s}_j^e(\boldsymbol{w}_{j-1}^e),\, j \leq i\}$. Therefore, it continues to hold that the error vectors $\widetilde{\boldsymbol{w}}_{k,i}^{'}$ belong to the filtration $\boldsymbol{\mathcal{F}}_{i-1}$ that is determined by the history of all iterates $\{\boldsymbol{w}_{k,j},\, j \leq i-1,\, k = 1, 2, \ldots, N\}$ that are generated by the original distributed strategy (8.46).

Working with recursion (10.19) is much more tractable for performance analysis because its dynamics is driven by the constant matrix $\mathcal{B}$ as opposed to the random matrix $\boldsymbol{\mathcal{B}}_{i-1}$ in the original error recursion (10.2). We shall therefore follow the following route to evaluate the MSD of the stochastic-gradient distributed algorithm (8.46). We shall work with the long-term model (10.19) and evaluate its MSD. Subsequently, we will argue that, under a bounding condition on the fourth-order moment of the gradient noise process, namely, condition (8.121), this MSD is within $O(\mu_{\max}^{3/2})$ from the true MSD expression that would have resulted had we worked directly with the original error recursion (10.2) without the approximation of ignoring $\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1}$. This fact will then allow us to conclude that the MSD expression that is derived from the long-term model (10.19) provides an accurate representation for the MSD of the original stochastic-gradient distributed strategy (8.46) to first-order in $\mu_{\max}$.

10.2 Size of Approximation Error

We first examine how close the trajectories of the original error recursion (10.2) and the long-term model (10.19) are to each other. We reproduce both recursions below with the state variable for the long-term model denoted by $\widetilde{\boldsymbol{w}}_i^{e'}$:

$$\widetilde{\boldsymbol{w}}_i^e = \boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{w}}_{i-1}^e + \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) - \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \tag{10.27}$$

$$\widetilde{\boldsymbol{w}}_i^{e'} = \mathcal{B}\,\widetilde{\boldsymbol{w}}_{i-1}^{e'} + \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) - \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \tag{10.28}$$

Observe that both models are driven by the *same* gradient noise process; in this way, the evolution of the long-term model is coupled to the evolution of the original recursion (but not the other way around). The next result establishes that the mean-square difference between the trajectories $\{\widetilde{\boldsymbol{w}}_i^e, \widetilde{\boldsymbol{w}}_i^{e'}\}$ is asymptotically bounded by $O(\mu_{\max}^2)$.

Theorem 10.2 (Performance error is $O(\mu_{\max}^{3/2})$)**.** Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P = A_1 A_o A_2$. Assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumptions 6.1 and 10.1. Assume further that the first and fourth-order moments of the gradient noise process satisfy the conditions of Assumption 8.1 with the second-order moment condition (8.115) replaced by the fourth-order moment condition (8.121). Then, it holds that, for sufficiently small step-sizes:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e - \widetilde{\boldsymbol{w}}_i^{e'}\|^2 \;=\; O(\mu_{\max}^2) \tag{10.29}$$

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^2 \;=\; \limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}\|^2 \;+\; O(\mu_{\max}^{3/2}) \tag{10.30}$$

Proof. To simplify the notation, we introduce the difference

$$\boldsymbol{z}_i \;\triangleq\; \widetilde{\boldsymbol{w}}_i^e - \widetilde{\boldsymbol{w}}_i^{e'} \tag{10.31}$$

Using (10.10) and (10.14), and subtracting recursions (10.27) and (10.28) we then get

$$\boldsymbol{z}_i = \mathcal{B}\boldsymbol{z}_{i-1} + \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1} \tag{10.32}$$

We also know from (9.173) that the matrix $\mathcal{B}$ is stable for sufficiently small step-sizes and, moreover, for $\mu_{\max} \ll 1$, it holds from (9.222) and (9.226) that

$$\rho(\mathcal{B}) = 1 - O(\mu_{\max}) \;=\; 1 - \sigma_b\mu_{\max} \tag{10.33}$$

for some positive constant σ_b that is independent of $\mu_{\max}$.

We multiply both sides of (10.32) from the left by $\mathcal{V}_\epsilon^{\mathsf{T}}$ and use (9.57) and (9.206) to get for $i \gg 1$:

$$\begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix} = \underbrace{\begin{bmatrix} I_{2M} - D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - D_{22}^{\mathsf{T}} \end{bmatrix}}_{\triangleq\, \bar{\mathcal{B}}} \begin{bmatrix} \bar{\boldsymbol{z}}_{i-1} \\ \check{\boldsymbol{z}}_{i-1} \end{bmatrix} + \mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1} \tag{10.34}$$

where the matrix

$$\bar{\mathcal{B}} = \mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{B}\left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} \tag{10.35}$$

is similar to $\mathcal{B}$ and is therefore stable by Theorem 9.3. We partition the vectors $\mathcal{V}_\epsilon^{\mathsf{T}}\boldsymbol{z}_i$ and $\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1}$ in recursion (10.34) into

$$\mathcal{V}_\epsilon^{\mathsf{T}}\boldsymbol{z}_i \triangleq \begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix}, \quad \mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1} \triangleq \begin{bmatrix} \bar{\boldsymbol{c}}_{i-1} \\ \check{\boldsymbol{c}}_{i-1} \end{bmatrix} \tag{10.36}$$

with the leading vectors, $\{\bar{\boldsymbol{z}}_i, \bar{\boldsymbol{c}}_{i-1}\}$, having dimensions $hM \times 1$ each. It follows that

$$\begin{bmatrix} \bar{\boldsymbol{z}}_i \\ \check{\boldsymbol{z}}_i \end{bmatrix} = \begin{bmatrix} I_{2M} - D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - D_{22}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{z}}_{i-1} \\ \check{\boldsymbol{z}}_{i-1} \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{c}}_{i-1} \\ \check{\boldsymbol{c}}_{i-1} \end{bmatrix} \tag{10.37}$$

This recursion has a form that is similar to the earlier recursion (9.285) we encountered while studying the mean stability of the original error dynamics (10.2) with two minor difference. First, the variables $\{\bar{\boldsymbol{z}}_i, \check{\boldsymbol{z}}_i, \bar{\boldsymbol{c}}_{i-1}, \check{\boldsymbol{c}}_{i-1}\}$ are now stochastic in nature and, second, the rightmost $O(\mu_{\max})$ perturbation term in (9.285) is absent from (10.37). Nevertheless, from an argument similar to the one that led to (9.282), we can similarly establish that

$$\|\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{c}_{i-1}\|^2 \leq r^2\mu_{\max}^2\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 \tag{10.38}$$

and, hence,

$$\|\bar{\boldsymbol{c}}_{i-1}\|^2 \leq r^2\mu_{\max}^2\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4, \quad \|\check{\boldsymbol{c}}_{i-1}\|^2 \leq r^2\mu_{\max}^2\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 \tag{10.39}$$

Moreover, repeating the argument that led to (9.292) and (9.294) we find that these recursions, under expectation, are now replaced by the following relations:

$$\begin{aligned} \mathbb{E}\,\|\bar{\boldsymbol{z}}_i\|^2 \;\leq\; & (1-\sigma_{11}\mu_{\max})\mathbb{E}\,\|\bar{\boldsymbol{z}}_{i-1}\|^2 + \\ & \frac{2\sigma_{21}^2\mu_{\max}}{\sigma_{11}}\mathbb{E}\,\|\check{\boldsymbol{z}}_{i-1}\|^2 + \frac{2r^2\mu_{\max}}{\sigma_{11}}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 \end{aligned} \tag{10.40}$$

and

$$\begin{aligned} \mathbb{E}\,\|\check{\boldsymbol{z}}_i\|^2 \;\leq\; & \left(\rho(J_\epsilon) + \epsilon + \frac{3\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\left\|\check{\boldsymbol{z}}_{i-1}^e\right\|^2 + \\ & \left(\frac{3\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\left\|\bar{\boldsymbol{z}}_{i-1}^e\right\|^2 + \\ & \left(\frac{3r^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^e\|^4 \end{aligned} \tag{10.41}$$

If we now introduce the scalar coefficients

$$a = 1-\sigma_{11}\mu_{\max} = 1-O(\mu_{\max}) \tag{10.42}$$

$$b = \frac{2\sigma_{21}^2\mu_{\max}}{\sigma_{11}} = O(\mu_{\max}) \tag{10.43}$$

$$c = \frac{3\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon} = O(\mu_{\max}^2) \tag{10.44}$$

$$d = \rho(J_\epsilon)+\epsilon+\frac{3\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon} = \rho(J_\epsilon)+\epsilon+O(\mu_{\max}^2) \tag{10.45}$$

$$e = \frac{2r^2\mu_{\max}}{\sigma_{11}} = O(\mu_{\max}) \tag{10.46}$$

$$f = \frac{3r^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon} = O(\mu_{\max}^2) \tag{10.47}$$

we can combine (10.40) and (10.41) into a single compact inequality recursion as follows:

$$\begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{z}}_i\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{z}}_i\|^2 \end{bmatrix} \preceq \underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{\Gamma} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{z}}_{i-1}\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{z}}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^{e}\|^4 \tag{10.48}$$

in terms of the 2×2 coefficient matrix Γ indicated above. Using the fact that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e}\|^4 \overset{(9.107)}{=} O(\mu_{\max}^2) \tag{10.49}$$

and relation (9.103) we conclude that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\bar{\boldsymbol{z}}_i\|^2 = O(\mu_{\max}^2), \qquad \limsup_{i\to\infty} \mathbb{E}\,\|\check{\boldsymbol{z}}_i\|^2 = O(\mu_{\max}^4) \tag{10.50}$$

and, hence,

$$\limsup_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2 = O(\mu_{\max}^2) \tag{10.51}$$

It follows that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e}-\widetilde{\boldsymbol{w}}_i^{e'}\|^2 = O(\mu_{\max}^2) \tag{10.52}$$

which establishes (10.29). Finally, note that

$$\begin{aligned} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}\|^2 &= \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}-\widetilde{\boldsymbol{w}}_i^{e}+\widetilde{\boldsymbol{w}}_i^{e}\|^2 \\ &\le \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}-\widetilde{\boldsymbol{w}}_i^{e}\|^2+\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e}\|^2 + 2\left|\mathbb{E}\,(\widetilde{\boldsymbol{w}}_i^{e'}-\widetilde{\boldsymbol{w}}_i^{e})^*\widetilde{\boldsymbol{w}}_i^{e}\right| \\ &\le \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}-\widetilde{\boldsymbol{w}}_i^{e}\|^2+\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e}\|^2 + 2\sqrt{\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}-\widetilde{\boldsymbol{w}}_i^{e}\|^2\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e}\|^2} \end{aligned} \tag{10.53}$$

and, hence, from (9.11) and (10.29) we get

$$\limsup_{i\to\infty}\left(\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}\|^2-\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e}\|^2\right)\;\le\;O(\mu_{\max}^2)\;+\;\sqrt{O(\mu_{\max}^3)}\;=\;O(\mu_{\max}^{3/2}) \tag{10.54}$$

since $\mu_{\max}^2<\mu_{\max}^{3/2}$ for small $\mu_{\max}\ll 1$, which establishes (10.30).

□

10.3 Stability of Second-Order Error Moment

We already know from the result of Theorem 9.1 that the original error recursion (10.2) is mean-square stable in the sense that $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ tends asymptotically to a region that is bounded by $O(\mu_{\max})$. Before launching into the performance analysis of the stochastic-gradient distributed algorithm (8.46), we first remark that the long-term approximate model (10.19) is also mean-square stable.

Lemma 10.3 (Mean-square stability of long-term model)**.** Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P=A_1A_oA_2$. Assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumptions 6.1 and 10.1. Assume further that the first and second-order moments of the gradient noise process satisfy the conditions of Assumption 8.1. Consider the iterates that are generated by the long-term model (10.19). Then, for sufficiently small step-sizes, it holds that

$$\limsup_{i\to\infty}\;\mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_{k,i}\|^2\;=\;O(\mu_{\max}),\quad k=1,2,\ldots,N \tag{10.55}$$

Proof. We multiply both sides of the long-term model (10.19) from the left by $\mathcal{V}_\epsilon^{\mathsf{T}}$ to get, for $i\gg 1$:

$$\underbrace{\begin{bmatrix}\bar{\boldsymbol{w}}_i^{e'}\\ \check{\boldsymbol{w}}_i^{e'}\end{bmatrix}}_{\triangleq\,\boldsymbol{z}_i}=\underbrace{\begin{bmatrix}I_{2M}-D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}}\\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}}-D_{22}^{\mathsf{T}}\end{bmatrix}}_{\triangleq\,\bar{\mathcal{B}}}\underbrace{\begin{bmatrix}\bar{\boldsymbol{w}}_{i-1}^{e'}\\ \check{\boldsymbol{w}}_{i-1}^{e'}\end{bmatrix}}_{\triangleq\,\boldsymbol{z}_{i-1}}+$$

$$\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{s}_i^e-\begin{bmatrix}0\\ \check{b}^e\end{bmatrix} \tag{10.56}$$

where the matrix $\bar{\mathcal{B}}$ is stable by Theorem 9.3, and where we are denoting the transformed error vector by $\boldsymbol{z}_i$ for ease of reference:

$$\boldsymbol{z}_i \triangleq \mathcal{V}_\epsilon^{\mathsf{T}} \widetilde{\boldsymbol{w}}_i^{e'} = \begin{bmatrix} \bar{\boldsymbol{w}}_i^{e'} \\ \check{\boldsymbol{w}}_i^{e'} \end{bmatrix} \tag{10.57}$$

We are also dropping the argument $\boldsymbol{w}_{i-1}^e$ from $\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)$ and writing simply $\boldsymbol{s}_i^e$. The long-term model (10.56) represents a dynamic system that is driven by two components: a deterministic (constant) driving term represented by $\check{b}^e$, and a random term represented by $\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)$. To facilitate the mean-square stability analysis, we may examine the contribution of these driving terms separately. For this purpose, we introduce the following two auxiliary recursions, one driven by the deterministic term and the other driven by the stochastic term and running over $i > i_o$ for some large enough $i_o \gg 1$:

$$a_i = \bar{\mathcal{B}}\, a_{i-1} + \begin{bmatrix} 0 \\ \check{b}^e \end{bmatrix} \tag{10.58}$$

$$\boldsymbol{b}_i = \bar{\mathcal{B}}\, \boldsymbol{b}_{i-1} + \mathcal{V}_\epsilon^{\mathsf{T}} \mathcal{A}_2^{\mathsf{T}} \mathcal{M} \boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \tag{10.59}$$

with initial conditions $a_{i_o} = 0$ and $\boldsymbol{b}_{i_o} = \boldsymbol{z}_{i_o}$ so that at any time instant $i > i_o$,

$$\boldsymbol{z}_i = \boldsymbol{b}_i - a_i \tag{10.60}$$

Consider first recursion (10.58) for a_i. Since $\bar{\mathcal{B}}$ is stable, the sequence a_i converges to

$$\begin{aligned} \lim_{i\to\infty} a_i &= (I - \bar{\mathcal{B}})^{-1} \begin{bmatrix} 0 \\ \check{b}^e \end{bmatrix} \\ &\overset{(9.229)}{=} O(\mu_{\max}) \end{aligned} \tag{10.61}$$

since $\check{b}^e = O(\mu_{\max})$. It follows that

$$\limsup_{i\to\infty} \|a_i\| = O(\mu_{\max}) \tag{10.62}$$

Consider next recursion (10.59) for $\boldsymbol{b}_i$. As was done earlier in (9.56) we partition the entries of $\mathcal{V}_\epsilon^{\mathsf{T}} \mathcal{A}_2^{\mathsf{T}} \mathcal{M} \boldsymbol{s}_i^e$ into:

$$\mathcal{V}_\epsilon^{\mathsf{T}} \mathcal{A}_2^{\mathsf{T}} \mathcal{M} \boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \triangleq \begin{bmatrix} \bar{\boldsymbol{s}}_i^e(\boldsymbol{w}_{i-1}^e) \\ \check{\boldsymbol{s}}_i^e(\boldsymbol{w}_{i-1}^e) \end{bmatrix} \tag{10.63}$$

We also partition the entries of $\boldsymbol{b}_i$ in the following manner in conformity with the dimensions of $\{\bar{\boldsymbol{s}}_i^e, \check{\boldsymbol{s}}_i^e\}$:

$$\underbrace{\begin{bmatrix} \bar{\boldsymbol{b}}_i \\ \check{\boldsymbol{b}}_i \end{bmatrix}}_{=\boldsymbol{b}_i} = \underbrace{\begin{bmatrix} I_{2M} - D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - D_{22}^{\mathsf{T}} \end{bmatrix}}_{\triangleq \bar{\mathcal{B}}} \underbrace{\begin{bmatrix} \bar{\boldsymbol{b}}_{i-1} \\ \check{\boldsymbol{b}}_{i-1} \end{bmatrix}}_{=\boldsymbol{b}_{i-1}} + \begin{bmatrix} \bar{\boldsymbol{s}}_i^e(\boldsymbol{w}_{i-1}^e) \\ \check{\boldsymbol{s}}_i^e(\boldsymbol{w}_{i-1}^e) \end{bmatrix} \tag{10.64}$$

This recursion has a form similar to the earlier recursion we encountered in (9.60) while studying the mean-square stability of the original error dynamics (10.2), with three differences. First, the driving term involving $\check{b}^e$ in (9.60) is not present in (10.64). Second, the matrices $\{D_{11}, D_{12}, D_{21}, D_{22}\}$ in (10.64) are constant matrices; nevertheless, they satisfy the same bounds as the matrices $\{\boldsymbol{D}_{11,i-1}, \boldsymbol{D}_{12,i-1}, \boldsymbol{D}_{21,i-1}, \boldsymbol{D}_{22,i-1}\}$ in (9.60). And, third, the argument of the noise terms $\{\bar{\boldsymbol{s}}_i^e, \check{\boldsymbol{s}}_i^e\}$ in (10.64) is $\boldsymbol{w}_{i-1}^e$ and not $\boldsymbol{b}_i$. However, these noise terms still satisfy the same bound given by (9.91), namely,

$$\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2 + \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2 \;\leq\; v_1^2 v_2^2 \beta_d^2 \mu_{\max}^2 \left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 + \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\right] + v_1^2 \mu_{\max}^2 \sigma_s^2 \tag{10.65}$$

in terms of the transformed vectors $\{\bar{\boldsymbol{w}}_{i-1}^e, \check{\boldsymbol{w}}_{i-1}^e\}$ defined by (9.55). Therefore, repeating the same argument that led to (9.106) will show that relations (9.69) and (9.81) still hold for $\{\mathbb{E}\,\|\bar{\boldsymbol{b}}_i\|^2, \mathbb{E}\,\|\check{\boldsymbol{b}}_i\|^2\}$, namely,

$$\mathbb{E}\,\|\bar{\boldsymbol{b}}_i\|^2 \;\leq\; (1-\sigma_{11}\mu_{\max})\mathbb{E}\,\|\bar{\boldsymbol{b}}_{i-1}\|^2 \;+\; \left(\frac{\sigma_{21}^2\mu_{\max}}{\sigma_{11}}\right)\mathbb{E}\,\|\check{\boldsymbol{b}}_{i-1}\|^2 \;+\; \mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2 \tag{10.66}$$

and

$$\begin{aligned}\mathbb{E}\,\|\check{\boldsymbol{b}}_i\|^2 \;\leq\; & \left(\rho(J_\epsilon) + \epsilon + \frac{2\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\left\|\check{\boldsymbol{b}}_{i-1}\right\|^2 + \\ & \left(\frac{2\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\left\|\bar{\boldsymbol{b}}_{i-1}\right\|^2 + \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2\end{aligned} \tag{10.67}$$

Using (10.65) we find that the last two recursive inequalities can be replaced by

$$\begin{aligned}\mathbb{E}\,\|\bar{\boldsymbol{b}}_i\|^2 \;\leq\; & (1-\sigma_{11}\mu_{\max})\,\mathbb{E}\,\|\bar{\boldsymbol{b}}_{i-1}\|^2 \;+\; \left(\frac{\sigma_{21}^2\mu_{\max}}{\sigma_{11}}\right)\mathbb{E}\,\|\check{\boldsymbol{b}}_{i-1}\|^2 \;+ \\ & v_1^2\mu_{\max}^2\sigma_s^2 \;+\; v_1^2 v_2^2 \beta_d^2 \mu_{\max}^2 \left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\right]\end{aligned} \tag{10.68}$$

and

$$\begin{aligned}\mathbb{E}\,\|\check{\boldsymbol{b}}_i\|^2 \;\leq\; & \left(\rho(J_\epsilon) + \epsilon + \frac{2\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\left\|\check{\boldsymbol{b}}_{i-1}\right\|^2 + \\ & \left(\frac{2\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon}\right)\mathbb{E}\,\left\|\bar{\boldsymbol{b}}_{i-1}\right\|^2 + v_1^2\mu_{\max}^2\sigma_s^2 \;+ \\ & v_1^2 v_2^2 \beta_d^2 \mu_{\max}^2 \left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \;+\; \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\right]\end{aligned} \tag{10.69}$$

If we now introduce the scalar coefficients

$$a = 1-\sigma_{11}\mu_{\max} = 1-O(\mu_{\max}) \tag{10.70}$$

$$b = \frac{\sigma_{21}^2\mu_{\max}}{\sigma_{11}} = O(\mu_{\max}) \tag{10.71}$$

$$c = \frac{2\sigma_{12}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon} = O(\mu_{\max}^2) \tag{10.72}$$

$$d = \rho(J_\epsilon)+\epsilon+\frac{3\sigma_{22}^2\mu_{\max}^2}{1-\rho(J_\epsilon)-\epsilon} = \rho(J_\epsilon)+\epsilon+O(\mu_{\max}^2) \tag{10.73}$$

$$e = v_1^2\mu_{\max}^2\sigma_s^2 = O(\mu_{\max}^2) \tag{10.74}$$

$$f = 0 \tag{10.75}$$

$$h = v_1^2v_2^2\beta_d^2\mu_{\max}^2 = O(\mu_{\max}^2) \tag{10.76}$$

we can combine (10.68) and (10.69) into a single compact inequality recursion as follows:

$$\begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{b}}_i\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{b}}_i\|^2 \end{bmatrix} \preceq \underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{\Gamma} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{b}}_{i-1}\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{b}}_{i-1}\| \end{bmatrix} + \begin{bmatrix} h & h \\ h & h \end{bmatrix} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 \end{bmatrix} + \begin{bmatrix} e \\ e \end{bmatrix} \tag{10.77}$$

in terms of the 2×2 coefficient matrix Γ indicated above. Using result (9.105) and the derivation leading to it we can similarly conclude that

$$\limsup_{i\to\infty}\ \mathbb{E}\,\|\bar{\boldsymbol{b}}_i\|^2 = O(\mu_{\max}), \qquad \limsup_{i\to\infty}\ \mathbb{E}\,\|\check{\boldsymbol{b}}_i\|^2 = O(\mu_{\max}^2) \tag{10.78}$$

and, hence,

$$\limsup_{i\to\infty}\ \mathbb{E}\,\|\boldsymbol{b}_i\|^2 = O(\mu_{\max}) \tag{10.79}$$

From (10.60) we have that $\|\boldsymbol{z}_i\|^2 \le 2\|a_i\|^2+2\|\boldsymbol{b}_i\|^2$ so that

$$\limsup_{i\to\infty}\mathbb{E}\,\|\boldsymbol{z}_i\|^2 = O(\mu_{\max}) \tag{10.80}$$

from which we conclude that (10.55) holds.

□

10.4 Stability of Fourth-Order Error Moment

In the next chapter we will employ the long-term model (10.19) to assess the performance of the multi-agent network as $i\to\infty$ and for sufficiently small step-sizes. In preparation for that discussion, we establish here the stability of the fourth-order moment of the error in

the long-term model (10.19) in a manner similar to what we did in Theorem 9.2 for the fourth-order moment of the error in the original recursion (10.2).

Lemma 10.4 (Fourth-order moment stability of long-term model)**.** Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P = A_1 A_o A_2$. Assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumptions 6.1 and 10.1. Assume further that the first and fourth-order moments of the gradient noise process satisfy the conditions of Assumption 8.1 with the second-order moment condition (8.115) replaced by the fourth-order moment condition (8.121). Then, the fourth-order moments of the error vectors generated by the long-term model (10.19) are stable for sufficiently small step-sizes, namely, it holds that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}'_{k,i}\|^4 \;=\; O(\mu_{\max}^2), \quad k = 1, 2, \ldots, N \tag{10.81}$$

Proof. We employ the same notation from the proof of Lemma 10.3 and reconsider recursions (10.58) and (10.64) for the auxiliary variables $\{a_i, \boldsymbol{b}_i\}$:

$$a_i \;=\; \bar{\mathcal{B}}\, a_{i-1} + \begin{bmatrix} 0 \\ \check{b}^e \end{bmatrix} \tag{10.82}$$

$$\underbrace{\begin{bmatrix} \bar{\boldsymbol{b}}_i \\ \check{\boldsymbol{b}}_i \end{bmatrix}}_{=\boldsymbol{b}_i} \;=\; \underbrace{\begin{bmatrix} I_{2M} - D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - D_{22}^{\mathsf{T}} \end{bmatrix}}_{\triangleq\, \bar{\mathcal{B}}} \underbrace{\begin{bmatrix} \bar{\boldsymbol{b}}_{i-1} \\ \check{\boldsymbol{b}}_{i-1} \end{bmatrix}}_{=\boldsymbol{b}_{i-1}} + \begin{bmatrix} \bar{\boldsymbol{s}}_i^e(\boldsymbol{w}_{i-1}^e) \\ \check{\boldsymbol{s}}_i^e(\boldsymbol{w}_{i-1}^e) \end{bmatrix} \tag{10.83}$$

Using (10.62), we readily conclude from (10.62) that

$$\limsup_{i\to\infty} \|a_i\|^4 \;=\; O(\mu_{\max}^4) \tag{10.84}$$

With regards to the recursion involving $\{\bar{\boldsymbol{b}}_i^e, \check{\boldsymbol{b}}_i^e\}$, we can unfold it and write

$$\bar{\boldsymbol{b}}_i^e \;=\; (I_{2M} - D_{11}^{\mathsf{T}})\bar{\boldsymbol{b}}_{i-1}^e \;-\; D_{21}^{\mathsf{T}}\check{\boldsymbol{b}}_{i-1}^e \;+\; \bar{\boldsymbol{s}}_i^e(\boldsymbol{w}_{i-1}^e) \tag{10.85}$$

$$\check{\boldsymbol{b}}_i^e \;=\; (\mathcal{J}_\epsilon^{\mathsf{T}} - D_{22}^{\mathsf{T}})\check{\boldsymbol{b}}_{i-1}^e \;-\; D_{12}^{\mathsf{T}}\bar{\boldsymbol{b}}_{i-1}^e \;+\; \check{\boldsymbol{s}}_i^e(\boldsymbol{w}_{i-1}^e) \tag{10.86}$$

These relations have similar forms to the earlier relations (9.108)–(9.109) we encountered while studying the stability of the fourth-order moment of the original error recursion (10.2), with three differences. First, the driving term involving $\check{b}^e$ in (9.109) is not present in (10.86). Second, the matrices $\{D_{11}, D_{12}, D_{21}, D_{22}\}$ in (10.85)–(10.86) are constant

matrices; nevertheless, they satisfy the same bounds as the matrices $\{\boldsymbol{D}_{11,i-1}, \boldsymbol{D}_{12,i-1}, \boldsymbol{D}_{21,i-1}, \boldsymbol{D}_{22,i-1}\}$ in (9.108)–(9.109). And, third, the argument of the noise terms $\{\bar{\boldsymbol{s}}_i^e, \check{\boldsymbol{s}}_i^e\}$ in (10.85)–(10.86) is $\boldsymbol{w}_{i-1}^e$ and not $\boldsymbol{b}_i$. However, these noise terms still satisfy the same bounds given by (9.91) and (9.131), namely,

$$\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^2 + \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^2 \leq v_1^2 v_2^2 \beta_d^2 \mu_{\max}^2 \left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 + \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2\right] \;+\; v_1^2 \mu_{\max}^2 \sigma_s^2 \tag{10.87}$$

and

$$\mathbb{E}\,\|\bar{\boldsymbol{s}}_i^e\|^4 + \mathbb{E}\,\|\check{\boldsymbol{s}}_i^e\|^4 \leq v_1^4 v_2^4 \beta_{d4}^4 \mu_{\max}^4 \left[\mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^4 + \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^4\right] + v_1^4 \mu_{\max}^4 \sigma_{s4}^4 \tag{10.88}$$

Therefore, repeating the same argument that led to (9.153) we can similarly show that

$$\begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{b}}_i\|^4 \\ \mathbb{E}\,\|\check{\boldsymbol{b}}_i\|^4 \end{bmatrix} \preceq \underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{\triangleq\, \Gamma'} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{b}}_{i-1}\|^4 \\ \mathbb{E}\,\|\check{\boldsymbol{b}}_{i-1}\|^4 \end{bmatrix} + \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{b}}_{i-1}\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{b}}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} a'' & b'' \\ c'' & d'' \end{bmatrix} \begin{bmatrix} \mathbb{E}\,\|\bar{\boldsymbol{w}}_{i-1}^e\|^2 \\ \mathbb{E}\,\|\check{\boldsymbol{w}}_{i-1}^e\|^2 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{10.89}$$

where

$$\begin{aligned}
a &= 1 - \sigma_{11}\mu_{\max} + O(\mu_{\max}^2) && (10.90)\\
b &= O(\mu_{\max}) && (10.91)\\
c &= O(\mu_{\max}^4) && (10.92)\\
d &= \rho(J_\epsilon) + \epsilon + O(\mu_{\max}^2) && (10.93)\\
a' &= O(\mu_{\max}^2) && (10.94)\\
b' &= O(\mu_{\max}^3) && (10.95)\\
c' &= O(\mu_{\max}^4) && (10.96)\\
d' &= O(\mu_{\max}^2) && (10.97)\\
a'' &= O(\mu_{\max}^2) && (10.98)\\
b'' &= O(\mu_{\max}^2) && (10.99)\\
c'' &= O(\mu_{\max}^2) && (10.100)\\
d'' &= O(\mu_{\max}^2) && (10.101)
\end{aligned}$$

and

$$\Gamma' = \begin{bmatrix} 1 - O(\mu_{\max}) & O(\mu_{\max}) \\ O(\mu_{\max}^4) & \rho(J_\epsilon) + \epsilon + O(\mu_{\max}^2) \end{bmatrix} \tag{10.102}$$

We again find that Γ' is a stable matrix for sufficiently small $\mu_{\max}$ and ϵ. Using results (9.105) and (10.78), and repeating the argument that led to (9.156)

we conclude that

$$\limsup_{i\to\infty}\mathbb{E}\,\|\bar{\boldsymbol{b}}_i\|^4 = O(\mu_{\max}^2), \quad \limsup_{i\to\infty}\mathbb{E}\,\|\check{\boldsymbol{b}}_i\|^4 = O(\mu_{\max}^4) \tag{10.103}$$

so that

$$\begin{aligned}
\limsup_{i\to\infty}\mathbb{E}\,\|\boldsymbol{b}_i\|^4 &= \limsup_{i\to\infty}\mathbb{E}\left(\left\|\begin{bmatrix}\bar{\boldsymbol{b}}_i\\ \check{\boldsymbol{b}}_i\end{bmatrix}\right\|^2\right)^2 \\
&= \limsup_{i\to\infty}\mathbb{E}\left(\|\bar{\boldsymbol{b}}_i\|^2+\|\check{\boldsymbol{b}}_i\|^2\right)^2 \\
&\le 2\left(\limsup_{i\to\infty}\mathbb{E}\left(\|\bar{\boldsymbol{b}}_i\|^4+\|\check{\boldsymbol{b}}_i\|^4\right)\right) \\
&= O(\mu_{\max}^2)
\end{aligned} \tag{10.104}$$

Now, from $\boldsymbol{z}_i = \boldsymbol{b}_i - a_i$ we have

$$\|\boldsymbol{z}_i\|^4 \;\le\; 8\|\boldsymbol{b}_i\|^4 \;+\; 8\|a_i\|^4 \tag{10.105}$$

and, therefore,

$$\limsup_{i\to\infty}\mathbb{E}\,\|\boldsymbol{z}_i\|^4 \;=\; O(\mu_{\max}^2) \tag{10.106}$$

Consequently,

$$\begin{aligned}
\limsup_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^4 &= \limsup_{i\to\infty}\;\mathbb{E}\left(\left\|\left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}}\begin{bmatrix}\bar{\boldsymbol{w}}_i^e\\ \check{\boldsymbol{w}}_i^e\end{bmatrix}\right\|^2\right)^2 \\
&= \limsup_{i\to\infty}\;\mathbb{E}\left(\left\|\left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}}\boldsymbol{z}_i\right\|^2\right)^2 \\
&\le \left\|\left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}}\right\|^4\left(\limsup_{i\to\infty}\;\mathbb{E}\,\|\boldsymbol{z}_i\|^4\right) \\
&= O(\mu_{\max}^2)
\end{aligned} \tag{10.107}$$

which leads to the desired result (10.81).

□

10.5 Stability of First-Order Error Moment

We can also establish the stability of the mean error for the long-term model (10.19).

Lemma 10.5 (Mean stability of long-term model)**.** Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P = A_1 A_o A_2$. Assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumptions 6.1 and 10.1. Assume further that the first and second-order moments of the gradient noise process satisfy the conditions of Assumption 8.1. Consider the iterates that are generated by the long-term model (10.19). Then, for sufficiently small step-sizes, it holds that

$$\limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}'_{k,i}\| \;=\; O(\mu_{\max}), \quad k = 1, 2, \ldots, N \tag{10.108}$$

Proof. Conditioning both sides of (10.19) on $\boldsymbol{\mathcal{F}}_{i-1}$, invoking the conditions on the gradient noise process from Assumption 8.1, and computing the conditional expectations we obtain:

$$\mathbb{E}\left[\widetilde{\boldsymbol{w}}_i^{e'} \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \;=\; \mathcal{B}\widetilde{\boldsymbol{w}}_{i-1}^{e'} \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \tag{10.109}$$

where the term involving $\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)$ is eliminated because $\mathbb{E}\,[\boldsymbol{s}_i^e|\boldsymbol{\mathcal{F}}_{i-1}] = 0$. Taking expectations again we arrive at

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i^{e'} \;=\; \mathcal{B}\left(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}^{e'}\right) \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \tag{10.110}$$

We multiply both sides of this recursion from the left by $\mathcal{V}_\epsilon^{\mathsf{T}}$ to get

$$\underbrace{\begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_i^{e'} \\ \mathbb{E}\,\check{\boldsymbol{w}}_i^{e'} \end{bmatrix}}_{\triangleq\, z_i} = \underbrace{\begin{bmatrix} I_{2M} - D_{11}^{\mathsf{T}} & -D_{21}^{\mathsf{T}} \\ -D_{12}^{\mathsf{T}} & \mathcal{J}_\epsilon^{\mathsf{T}} - D_{22}^{\mathsf{T}} \end{bmatrix}}_{\triangleq\, \bar{\mathcal{B}}} \underbrace{\begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_{i-1}^{e'} \\ \mathbb{E}\,\check{\boldsymbol{w}}_{i-1}^{e'} \end{bmatrix}}_{\triangleq\, z_{i-1}} - \begin{bmatrix} 0 \\ \check{b}^e \end{bmatrix} \tag{10.111}$$

where the matrix $\bar{\mathcal{B}}$ is stable by Theorem 9.3. For simplicity, we denote the state variable in (10.111) by z_i, so that we can rewrite the recursion more compactly in the form

$$z_i \;=\; \bar{\mathcal{B}}z_{i-1} - \begin{bmatrix} 0 \\ \check{b}^e \end{bmatrix} \tag{10.112}$$

This is a first-order recursion that is driven by a constant term. Since $\bar{\mathcal{B}}$ is stable and $\check{b}^e = O(\mu_{\max})$, we conclude from (10.112) that

$$\begin{aligned} \lim_{i\to\infty} z_i \;&=\; -(I - \bar{\mathcal{B}})^{-1} \begin{bmatrix} 0 \\ \check{b}^e \end{bmatrix} \\ &\overset{(9.229)}{=}\; \begin{bmatrix} O(1/\mu_{\max}) & O(1) \\ O(1) & O(1) \end{bmatrix} \begin{bmatrix} 0 \\ O(\mu_{\max}) \end{bmatrix} \\ &=\; O(\mu_{\max}) \end{aligned} \tag{10.113}$$

It follows that

$$\limsup_{i\to\infty} \|z_i\| \;=\; O(\mu_{\max}) \tag{10.114}$$

Consequently,

$$\limsup_{i\to\infty} \left\| \begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_i^{e'} \\ \mathbb{E}\,\check{\boldsymbol{w}}_i^{e'} \end{bmatrix} \right\| \;=\; O(\mu_{\max}) \tag{10.115}$$

and, hence,

$$\begin{aligned} \limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}'_{k,i}\| &\leq \limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i^{e'}\| \\ &= \limsup_{i\to\infty} \left\| \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} \begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_i^{e'} \\ \mathbb{E}\,\check{\boldsymbol{w}}_i^{e'} \end{bmatrix} \right\| \\ &\leq \left\| \left(\mathcal{V}_\epsilon^{-1}\right)^{\mathsf{T}} \right\| \left(\limsup_{i\to\infty} \left\| \begin{bmatrix} \mathbb{E}\,\bar{\boldsymbol{w}}_i^{e'} \\ \mathbb{E}\,\check{\boldsymbol{w}}_i^{e'} \end{bmatrix} \right\| \right) \\ &= O(\mu_{\max}) \end{aligned} \tag{10.116}$$

as claimed.

□

10.6 Comparing Consensus and Diffusion Strategies

Using results from the previous sections, we are able to compare some stability properties of diffusion and consensus networks. Recall from (8.7)–(8.10) that the consensus and diffusion strategies correspond to the following choices for $\{A_o, A_1, A_2\}$ in terms of a single combination matrix A in the general description (8.46):

$$\text{consensus:} \qquad A_o = A, \;\; A_1 = I_N = A_2 \tag{10.117}$$

$$\text{CTA diffusion:} \qquad A_1 = A, \;\; A_2 = I_N = A_o \tag{10.118}$$

$$\text{ATC diffusion:} \qquad A_2 = A, \;\; A_1 = I_N = A_o \tag{10.119}$$

Example 10.1 (Stabilizing effect of diffusion networks)**.** We revisit the conclusion of Example 8.4, albeit now under more general costs. Thus, refer to the mean recursion (10.110), namely,

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i^{e'} \;=\; \mathcal{B}\left(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}^{e'}\right) \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}b^e \tag{10.120}$$

which is driven by a constant matrix $\mathcal{B}$. Using the choices (10.117)–(10.119), the $\mathcal{B}$ matrix is given by the following expressions in terms of the $\mathcal{B}$ matrix for the non-cooperative strategy:

$$\begin{aligned}
\mathcal{B}_{\rm ncop} &= I_{hMN} - \mathcal{M}\mathcal{H} && \text{(non-cooperation)} && (10.121)\\
\mathcal{B}_{\rm cons} &= \mathcal{B}_{\rm ncop} + \big(\mathcal{A}^{\mathsf{T}} - I_{hMN}\big) && \text{(consensus)} && (10.122)\\
\mathcal{B}_{\rm atc} &= \mathcal{A}^{\mathsf{T}}\,\mathcal{B}_{\rm ncop} && \text{(ATC diffusion)} && (10.123)\\
\mathcal{B}_{\rm cta} &= \mathcal{B}_{\rm ncop}\,\mathcal{A}^{\mathsf{T}} && \text{(CTA diffusion)} && (10.124)
\end{aligned}$$

where $\mathcal{A} = A \otimes I_{hM}$ and $h = 1$ for real data and $h = 2$ for complex data. We encountered a similar structure in expressions (8.30)–(8.33) for the case of MSE networks in Example 8.3, where the mean error vector evolved instead according to the recursion:

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i \;=\; \mathcal{B}\;(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}) \tag{10.125}$$

without the additional driving terms appearing in (10.120). Now, observe that the coefficient matrices $\{\mathcal{B}_{\rm atc}, \mathcal{B}_{\rm cta}\}$ shown in (10.123)–(10.124) for the diffusion strategies are expressed in terms of $\mathcal{B}_{\rm ncop}$ in a *multiplicative* manner, while $\mathcal{B}_{\rm cons}$ is related to $\mathcal{B}_{\rm ncop}$ in an *additive* manner. These structures have an important implication on mean stability in view of the following matrix result.

Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be any left-stochastic matrices with blocks of size $hM \times hM$, and let $\mathcal{D}$ be any Hermitian block-diagonal positive-definite matrix also with blocks of size $hM \times hM$. Then, it holds from property (F.24) in the appendix that $\rho(\mathcal{X}_2^{\mathsf{T}}\mathcal{D}\mathcal{X}_1^{\mathsf{T}}) \leq \rho(\mathcal{D})$. That is, multiplication of $\mathcal{D}$ by left-stochastic transformations generally reduces the spectral radius. This result can be used to establish the stability of the diffusion dynamics (i.e., of $\mathcal{B}_{\rm diff}$) whenever the non-cooperative strategy is stable (i.e., $\mathcal{B}_{\rm ncop}$) and regardless of the combination policy, A. Indeed, note that $\mathcal{B}_{\rm ncop}$ has a Hermitian block-diagonal structure similar to $\mathcal{D}$ and that it is stable for any $\mu_{\max} < 2/\rho(\mathcal{H})$:

$$\mathcal{B}_{\rm ncop} \text{ stable} \iff \mu_{\max} < \frac{2}{\rho(\mathcal{H})} \tag{10.126}$$

The matrix $\mathcal{A}$ in (10.123)–(10.124) plays the role of $\mathcal{X}_1$ or $\mathcal{X}_2$. Therefore, it follows that, whenever (10.126) holds, it will also hold that $\rho(\mathcal{B}_{\rm atc}) < 1$ and $\rho(\mathcal{B}_{\rm cta}) < 1$ for any $\mathcal{A}$. The same conclusion does not generally hold for $\mathcal{B}_{\rm cons}$ [248]. Note further that since $\rho(\mathcal{B}_{\rm atc}) \leq \rho(\mathcal{B}_{\rm ncop})$ and $\rho(\mathcal{B}_{\rm cta}) \leq \rho(\mathcal{B}_{\rm ncop})$, it follows that diffusion strategies have a stabilizing effect.

■

Example 10.2 (Two interacting agents)**.** We illustrate further the conclusion of Example 10.1 by considering the case of an MSE network (cf. Example 8.2)

consisting of two interacting agents shown in Figure 10.1 [248], with

$$R_{u,1} = \sigma_{u,1}^2 \, I_M, \quad R_{u,2} = \sigma_{u,2}^2 \, I_M \tag{10.127}$$

Without loss of generality, we assume

$$\mu_1 \sigma_{u,1}^2 \;\leq\; \mu_2 \sigma_{u,2}^2 \tag{10.128}$$

Agent 1 uses combination weights $\{1-a, a\}$, while agent 2 uses combination weights $\{1-b, b\}$ with $a, b \in (0,1)$. The combination matrix A is therefore given by

$$A = \begin{bmatrix} 1-a & b \\ a & 1-b \end{bmatrix} \tag{10.129}$$

which is left-stochastic. If desired, a symmetric A can be obtained by setting $a = b$.

The agents run either the consensus LMS strategy (7.14) or the diffusion LMS strategy (7.22) or (7.23). We already know from (8.28) in Example 8.2 that the mean error recursion for the non-cooperative, diffusion, and consensus LMS strategies running over complex data evolve according to the following dynamics:

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i \;=\; \mathcal{B}\,(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}), \qquad i \geq 0 \tag{10.130}$$

with the coefficient matrix $\mathcal{B}$ given by the following expressions for the various strategies under consideration (we are only showing the $\mathcal{B}$ matrix for the ATC strategy since the argument is similar for CTA):

$$\mathcal{B}_{\text{ncop}} \;=\; \begin{bmatrix} 1-\mu_1\sigma_{u,1}^2 & 0 \\ 0 & 1-\mu_2\sigma_{u,2}^2 \end{bmatrix} \otimes I_M \tag{10.131}$$

$$\mathcal{B}_{\text{atc}} \;=\; \begin{bmatrix} (1-a)(1-\mu_1\sigma_{u,1}^2) & a(1-\mu_2\sigma_{u,2}^2) \\ b(1-\mu_1\sigma_{u,1}^2) & (1-b)(1-\mu_2\sigma_{u,2}^2) \end{bmatrix} \otimes I_M \tag{10.132}$$

$$\mathcal{B}_{\text{cons}} \;=\; \begin{bmatrix} (1-a)-\mu_1\sigma_{u,1}^2 & a \\ b & (1-b)-\mu_2\sigma_{u,2}^2 \end{bmatrix} \otimes I_M \tag{10.133}$$

We first assume that

$$0 < \mu_1\sigma_{u,1}^2 \leq \mu_2\sigma_{u,2}^2 < 2 \tag{10.134}$$

so that each of the individual agents is stable in the mean and, hence, the matrix $\mathcal{B}_{\text{ncop}}$ given above is stable. Then, from the conclusion of Example 10.1 above we know that the diffusion network will also be stable in the mean for any choice of the parameters $\{a, b\}$. This is because the stability of $\mathcal{B}_{\text{ncop}}$ guarantees the stability of $\mathcal{B}_{\text{atc}}$.

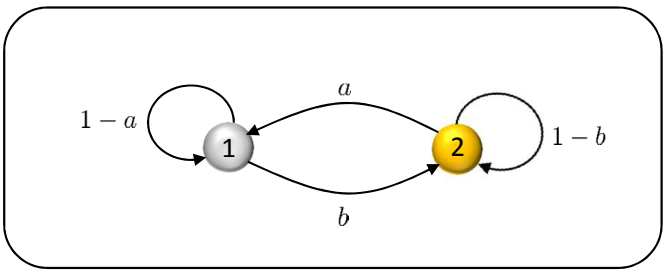

Figure 10.1: A two-agent MSE network with agent 1 using combination weights $\{a, 1-a\}$ and agent 2 using combination weights $\{b, 1-b\}$.

We now verify that there are choices for the combination parameters $\{a, b\}$ that will destabilize the consensus network (even though the individual agents are themselves stable in the mean). Specifically, we verify below that if the parameters $\{a, b\} \in (0, 1)$ are chosen to satisfy

$$a + b \geq 2 - \mu_1 \sigma_{u,1}^2 > 0 \tag{10.135}$$

then consensus will lead to unstable mean behavior, i.e., $\mathbb{E}\,\widetilde{\boldsymbol{w}}_i$ will grow unbounded. Indeed, note first that the minimum eigenvalue of $\mathcal{B}_{\text{cons}}$ can be found to be

$$\lambda_{\min}(\mathcal{B}_{\text{cons}}) = \frac{1}{2}\left((2 - a - b - \mu_1 \sigma_{u,1}^2 - \mu_2 \sigma_{u,2}^2) - \sqrt{\tau}\,\right) \tag{10.136}$$

where

$$\begin{aligned} \tau &\triangleq (b - a - \mu_1 \sigma_{u,1}^2 + \mu_2 \sigma_{u,2}^2)^2 + 4ab \\ &= (b + a + \mu_1 \sigma_{u,1}^2 - \mu_2 \sigma_{u,2}^2)^2 + 4b(\mu_2 \sigma_{u,2}^2 - \mu_1 \sigma_{u,1}^2) \end{aligned} \tag{10.137}$$

From the first equality in (10.137), we conclude that $\tau \geq 0$ and, hence, that $\lambda_{\min}(\mathcal{B}_{\text{cons}})$ is real. Moreover, using (10.134)–(10.135), we have that

$$b + a + \mu_1 \sigma_{u,1}^2 - \mu_2 \sigma_{u,2}^2 \geq 0 \tag{10.138}$$

$$4b(\mu_2 \sigma_{u,2}^2 - \mu_1 \sigma_{u,1}^2) \geq 0 \tag{10.139}$$

It follows that

$$\begin{aligned} \lambda_{\min}(\mathcal{B}_{\text{cons}}) &\leq \frac{1}{2}\left((2 - a - b - \mu_1 \sigma_{u,1}^2 - \mu_2 \sigma_{u,2}^2) - (b + a + \mu_1 \sigma_{u,1}^2 - \mu_2 \sigma_{u,2}^2) \right) \\ &= 1 - b - a - \mu_1 \sigma_{u,1}^2 \\ &\leq -1 \end{aligned} \tag{10.140}$$

where (10.140) follows from (10.135). We conclude that the consensus network is unstable since the eigenvalues of $\mathcal{B}_{\text{cons}}$ do not lie strictly inside the unit circle.

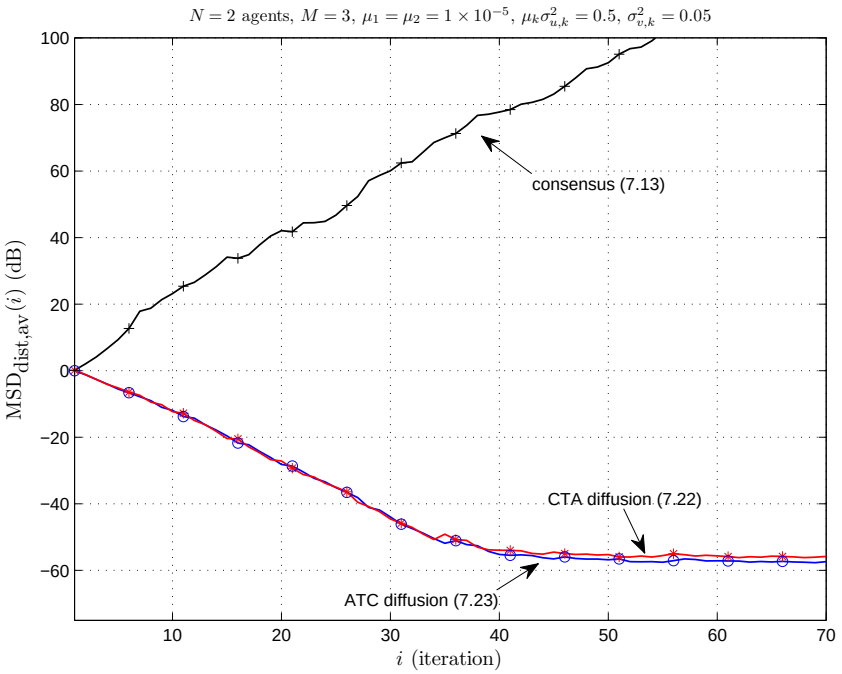

Figure 10.2: Evolution of the learning curves for the diffusion and consensus strategies for the numerical values $\mu_1 = \mu_2 = 1 \times 10^{-5}$, $\mu_1 \sigma_{u,1}^2 = \mu_2 \sigma_{u,2}^2 = 0.5$, and $(a, b) = (0.8, 0.8)$. These numerical values satisfy (10.135) for which the consensus solution becomes unstable.

Figure 10.2 illustrates these results for the two-agent MSE network of Figure 10.1 dealing with complex-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ satisfying the model $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i)$ with $M = 3$. The unknown vector w^o is generated randomly and its norm is normalized to one. The figure plots the evolution of the ensemble-average learning curves, $\frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, for consensus, ATC diffusion, and CTA diffusion using $\mu_1 = \mu_2 = 1 \times 10^{-5}$. The measure $\frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ corresponds to the average mean-square-deviation (MSD) of the agents at time i since

$$\frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; \frac{1}{2}\left(\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{1,i}\|^2 + \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{2,i}\|^2\right) \tag{10.141}$$

and $\widetilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}$. The learning curves are obtained by averaging the

trajectories $\{\frac{1}{2}\|\widetilde{\boldsymbol{w}}_i\|^2\}$ over 100 repeated experiments. The label on the vertical axis in the figure refers to the learning curves $\frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ by writing $\mathrm{MSD}_{\mathrm{dist,av}}(i)$, with an iteration index i and where the subscripts "dist" and "av" are meant to indicate that this is an average performance measure for the distributed solution. Each experiment in this simulation involves running the consensus (7.13) or diffusion (7.22)–(7.23) LMS recursions with $h = 2$ on the complex-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$. The simulations use $\sigma_{v,1}^2 = \sigma_{v,2}^2 = 0.05$, $\mu_1\sigma_{u,1}^2 = \mu_2\sigma_{u,2}^2 = 0.5$, and $(a, b) = (0.8, 0.8)$. These numerical values ensure that (10.134) and (10.135) are satisfied so that the individual agents and the diffusion strategy are both mean stable, while the consensus strategy becomes unstable in the mean. The small step-sizes ensure that the networks are mean-square stable. It is seen in the figure that the learning curve of the consensus strategy grows unbounded while the learning curve of the diffusion strategies tend towards steady-state values.

Next, we consider an example satisfying

$$0 < \mu_1\sigma_{u,1}^2 < 2 \leq \mu_2\sigma_{u,2}^2 \tag{10.142}$$

so that, for the non-cooperative mode of operation, agent 1 is still stable while agent 2 is unstable. From the first equality of (10.137), we again conclude that

$$\begin{aligned}
\lambda_{\min}(\mathcal{B}_{\mathrm{cons}}) &\leq \frac{1}{2}\left((2-a-b-\mu_1\sigma_{u,1}^2-\mu_2\sigma_{u,2}^2) - |b-a-\mu_1\sigma_{u,1}^2+\mu_2\sigma_{u,2}^2|\right) \\
&= \begin{cases} 1-a-\mu_1\sigma_{u,1}^2, & \text{if } b+\mu_2\sigma_{u,2}^2 \leq a+\mu_1\sigma_{u,1}^2 \\ 1-b-\mu_2\sigma_{u,2}^2, & \text{otherwise} \end{cases} \\
&\leq 1-b-\mu_2\sigma_{u,2}^2 \\
&\leq 1-\mu_2\sigma_{u,2}^2 \\
&\leq -1
\end{aligned} \tag{10.143}$$

That is, in this second case, no matter how we choose the parameters $\{a, b\}$, the consensus network is always unstable. In contrast, the diffusion network is able to stabilize the network, i.e., there are choices for $\{a, b\}$ that lead to stable behavior. To see this, we set $b = 1 - a$ so that the eigenvalues of $\mathcal{B}_{\mathrm{atc}}$ are

$$\lambda\left(\mathcal{B}_{\mathrm{atc}}\right) \in \{0,\ 1-\mu_1\sigma_{u,1}^2 - (\mu_2\sigma_{u,2}^2 - \mu_1\sigma_{u,1}^2)a\} \tag{10.144}$$

Some straightforward algebra shows that the magnitude of the nonzero eigenvalue will be bounded by one and, hence, the diffusion network will be stable in the mean if a satisfies:

$$0 \leq a < \frac{2-\mu_1\sigma_{u,1}^2}{\mu_2\sigma_{u,2}^2 - \mu_1\sigma_{u,1}^2} \tag{10.145}$$

■

11

Performance of Multi-Agent Networks

We established in Theorem 9.1 that a multi-agent network running the distributed strategy (8.46) is mean-square stable for sufficiently small step-size parameters. More specifically, we showed that, for each agent k, the error variance relative to the limit point, $w^\star$, defined by (8.55), enters a bounded region whose size is in the order of $O(\mu_{\max})$:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \;=\; O(\mu_{\max}), \quad k=1,2,\ldots,N \tag{11.1}$$

In this chapter, we will assess the size of these mean-square errors for both cases of real and complex data. We will measure the mean-square-deviation (MSD) at each agent k, as well as for the entire network, by using the following definitions:

$$\mathrm{MSD}_{\mathrm{dist},k} \;\triangleq\; \mu_{\max}\cdot\left(\lim_{\mu_{\max}\to 0}\limsup_{i\to\infty}\frac{1}{\mu_{\max}}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2\right) \tag{11.2}$$

$$\mathrm{MSD}_{\mathrm{dist,av}} \;\triangleq\; \frac{1}{N}\left(\sum_{k=1}^{N}\mathrm{MSD}_{\mathrm{dist},k}\right) \tag{11.3}$$

The form of expression (11.2) for the MSD was motivated earlier in (4.94) while studying single-agent adaptation, except that here we are scaling by $\mu_{\max}$ since we can now have multiple step-sizes $\{\mu_k\}$ across

the agents. The subscript "dist" in the above two expressions is used to indicate that these measures relate to the *distributed* implementation. Note that the network performance is defined in terms of the average MSD value across all agents. We will derive closed-form expressions for the MSD performance for both cases of real and complex-valued data — see, e.g., (11.118), as well as for the excess-risk (ER) metric defined later by (11.34) — see, e.g., (11.186). If we examine, for instance, expression (11.118) for the MSD, we observe that it is proportional to $\mu_{\max}$, i.e., it is small and in the order of $O(\mu_{\max})$, as expected from (11.1). In this way, we will be able to conclude that network adaptation with small constant step-sizes is able to lead to reliable performance even in the presence of gradient noise, which is a reassuring result.

11.1 Conditions on Costs and Noise

The presentation will assume the same conditions we used in the last two chapters to examine the stability of multi-agent networks. In particular, we assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumptions 6.1 and 10.1. We also assume that the first and fourth-order moments of the gradient noise process satisfy the conditions of Assumption 8.1 with the second-order moment condition (8.115) replaced by the fourth-order moment condition (8.121), in addition to a smoothness condition on the noise covariance matrices defined as follows.

We refer to the definition of the individual gradient noise processes in (8.109), namely, for any $\boldsymbol{\phi} \in \boldsymbol{\mathcal{F}}_{i-1}$:

$$\boldsymbol{s}_{k,i}(\boldsymbol{\phi}) \;\triangleq\; \widehat{\nabla_{w^*} J_k}(\boldsymbol{\phi}) \;-\; \nabla_{w^*} J_k(\boldsymbol{\phi}) \tag{11.4}$$

where $\boldsymbol{\mathcal{F}}_{i-1}$ denotes the filtration corresponding to all past iterates across all agents:

$$\boldsymbol{\mathcal{F}}_{i-1} \;=\; \text{filtration defined by } \{\boldsymbol{w}_{k,j},\; j \leq i-1,\; k = 1, 2, \ldots, N\} \tag{11.5}$$

We define the extended gradient noise vector of size $2M \times 1$:

$$\boldsymbol{s}^{e}_{k,i}(\boldsymbol{\phi}) \;\triangleq\; \begin{bmatrix} \boldsymbol{s}_{k,i}(\boldsymbol{\phi}) \\ \big(\boldsymbol{s}^{*}_{k,i}(\boldsymbol{\phi})\big)^{\mathsf{T}} \end{bmatrix} \tag{11.6}$$

We denote its conditional covariance matrix by

$$R^e_{s,k,i}(\boldsymbol{\phi}) \triangleq \mathbb{E}\left[\boldsymbol{s}^e_{k,i}(\boldsymbol{\phi})\boldsymbol{s}^{e*}_{k,i}(\boldsymbol{\phi})\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] \tag{11.7}$$

We further assume that, in the limit, the following moment matrices tend to constant values when evaluated at the limit point $w^\star$:

$$R_{s,k} \triangleq \lim_{i\to\infty}\mathbb{E}\left[\boldsymbol{s}_{k,i}(w^\star)\boldsymbol{s}^*_{k,i}(w^\star)\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] \tag{11.8}$$

$$R_{q,k} \triangleq \lim_{i\to\infty}\mathbb{E}\left[\boldsymbol{s}_{k,i}(w^\star)\boldsymbol{s}^{\mathsf{T}}_{k,i}(w^\star)\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] \tag{11.9}$$

Assumption 11.1 (Smoothness condition on noise covariance)**.** It is assumed that the conditional second-order moments of the individual noise processes satisfy smoothness conditions similar to (5.37), namely,

$$\left\|R^e_{s,k,i}(w^\star+\Delta w) - R^e_{s,k,i}(w^\star)\right\| \le \kappa_d\,\|\Delta w\|^\gamma \tag{11.10}$$

in terms of the extended covariance matrix, for small perturbations $\|\Delta w\|\le\epsilon$, and for some constants $\kappa_d\ge 0$ and exponent $0<\gamma\le 4$.

Following the argument that led to (4.24) in the single-agent case, we can similarly show that the conditional noise covariance matrix satisfies more globally a condition of the following form for all $\boldsymbol{\phi}\in\boldsymbol{\mathcal{F}}_{i-1}$:

$$\left\|R^e_{s,k,i}(\boldsymbol{\phi}) - R^e_{s,k,i}(w^\star)\right\| \le \kappa_d\,\|\widetilde{\boldsymbol{\phi}}\|^\gamma + \kappa'_d\,\|\widetilde{\boldsymbol{\phi}}\|^2 \tag{11.11}$$

where $\widetilde{\boldsymbol{\phi}} = w^\star - \boldsymbol{\phi}$ and for some constant $\kappa'_d\ge 0$.

The performance expressions that will be derived in this chapter will be expressed in terms of the following quantities, defined for both cases of real or complex data.

Definition 11.1 (Hessian and moment matrices)**.** We associate with each agent k a pair of matrices $\{H_k, G_k\}$, both of which are evaluated at the location of the limit point $w=w^\star$. The matrices are defined as follows:

$$H_k \triangleq \nabla^2_w\,J_k(w^\star),\quad G_k \triangleq \begin{cases} R_{s,k} & \text{(real case)}\\ \begin{bmatrix} R_{s,k} & R_{q,k}\\ R^*_{q,k} & R^{\mathsf{T}}_{s,k}\end{bmatrix} & \text{(complex case)}\end{cases} \tag{11.12}$$

Both matrices are dependent on the data type (whether real or complex); in particular, each H_k is $2M \times 2M$ for complex data and $M \times M$ for real data. Note that $H_k \geq 0$ and $G_k \geq 0$.

In view of the lower bound condition in (6.13), it follows that

$$\sum_{k=1}^{N} q_k H_k \;>\; 0 \tag{11.13}$$

so that the weighted sum of the $\{H_k\}$ matrices is invertible. This matrix sum will appear in the performance expressions.

In a manner similar to Lemma 4.1, one useful conclusion that follows from the smoothness condition (11.10) and from (11.11) is that, after sufficient iterations, we can express the covariance matrix of the gradient noise process, $\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi})$, in terms of the same limiting matrices $\{G_k\}$ defined by (11.12). This fact is established next and will be employed later in the proof of Theorem 11.2. For the sake of the argument used in the derivation of the lemma below, we recall from the explanation following (8.134) that each noise component, $\boldsymbol{s}_{k,i}^{e}(\cdot)$, is actually dependent on the iterate $\boldsymbol{\phi}_{k,i-1}$ and, hence, we will write this noise component more explicitly as $\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})$. We further recall from the distributed algorithm (8.46) that $\boldsymbol{\phi}_{k,i-1}$ is a convex combination of various $\{\boldsymbol{w}_{\ell,i-1}\}$ from the neighborhood of agent k. This property is exploited in the derivation.

Lemma 11.1 (Limiting second-order moment of gradient noise)**.** Under the smoothness condition (11.10), and for sufficiently small step-sizes, it holds that the covariance matrix of the extended gradient noise process, $\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})$, at each agent k satisfies for $i \gg 1$:

$$\mathbb{E}\,\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\left(\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\right)^{*} \;=\; G_k \;+\; O\left(\mu^{\min\{1,\frac{\gamma}{2}\}}\right) \tag{11.14}$$

where $0 < \gamma \leq 4$ and G_k is given by (11.12). Consequently, it holds for $i \gg 1$ that the trace of the covariance matrix satisfies:

$$\mathrm{Tr}(G_k) - b_o \;\leq\; \mathbb{E}\,\|\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\|^2 \;\leq\; \mathrm{Tr}(G_k) + b_o \tag{11.15}$$

for some nonnegative value $b_o = O\left(\mu^{\min\{1,\frac{\gamma}{2}\}}\right)$.

Proof. By adding and subtracting the same term we have [71, 278]:

$$\begin{aligned}
&\mathbb{E}\left[\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\left(\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\right)^{*}\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] \\
&\quad=\ \mathbb{E}\left[\boldsymbol{s}_{k,i}^{e}(w^{\star})\left(\boldsymbol{s}_{k,i}^{e}(w^{\star})\right)^{*}\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right]+ \\
&\qquad\mathbb{E}\left[\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\left(\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\right)^{*}\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right]- \\
&\qquad\mathbb{E}\left[\boldsymbol{s}_{k,i}^{e}(w^{\star})\left(\boldsymbol{s}_{k,i}^{e}(w^{\star})\right)^{*}\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right]
\end{aligned} \tag{11.16}$$

which, upon using definition (11.7), can be rewritten as:

$$\begin{aligned}
&\mathbb{E}\left[\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\left(\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\right)^{*}\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right] \\
&\quad=\ \mathbb{E}\left[\boldsymbol{s}_{k,i}^{e}(w^{\star})\left(\boldsymbol{s}_{k,i}^{e}(w^{\star})\right)^{*}\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right]+ \\
&\qquad R_{s,k,i}^{e}(\boldsymbol{\phi}_{k,i-1})-R_{s,k,i}^{e}(w^{\star})
\end{aligned} \tag{11.17}$$

Subtracting the covariance matrix G_k defined by (11.12) from both sides, and computing expectations, we get:

$$\begin{aligned}
&\mathbb{E}\,\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\left(\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\right)^{*}-G_k \\
&\quad=\ \mathbb{E}\left(\mathbb{E}\left[\boldsymbol{s}_{k,i}^{e}(w^{\star})\left(\boldsymbol{s}_{k,i}^{e}(w^{\star})\right)^{*}\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right]-G_k\right)+ \\
&\qquad\mathbb{E}\left(R_{s,k,i}^{e}(\boldsymbol{\phi}_{k,i-1})-R_{s,k,i}^{e}(w^{\star})\right)
\end{aligned} \tag{11.18}$$

It then follows from the triangle inequality of norms, and from Jensen's inequality (F.29) in the appendix, that

$$\begin{aligned}
&\left\|\mathbb{E}\,\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\left(\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\right)^{*}-G_k\right\| \\
&\quad\le\left\|\mathbb{E}\left(\mathbb{E}\left[\boldsymbol{s}_{k,i}^{e}(w^{\star})\left(\boldsymbol{s}_{k,i}^{e}(w^{\star})\right)^{*}\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right]-G_k\right)\right\|+ \\
&\qquad\left\|\mathbb{E}\left(R_{s,k,i}^{e}(\boldsymbol{\phi}_{k,i-1})-R_{s,k,i}^{e}(w^{\star})\right)\right\| \\
&\quad\overset{(F.29)}{\le}\ \mathbb{E}\left\|\mathbb{E}\left[\boldsymbol{s}_{k,i}^{e}(w^{\star})\left(\boldsymbol{s}_{k,i}^{e}(w^{\star})\right)^{*}\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right]-G_k\right\|+ \\
&\qquad\mathbb{E}\left\|R_{s,k,i}^{e}(\boldsymbol{\phi}_{k,i-1})-R_{s,k,i}^{e}(w^{\star})\right\|
\end{aligned} \tag{11.19}$$

Computing the limit superior of both sides, and using (11.8)–(11.9) to annihilate the limit of the first term on the right-hand side, we conclude that

$$\begin{aligned}
\limsup_{i\to\infty}\left\|\mathbb{E}\,\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\left(\boldsymbol{s}_{k,i}^{e}(\boldsymbol{\phi}_{k,i-1})\right)^{*}-G_k\right\|\ \le \\
\limsup_{i\to\infty}\mathbb{E}\left\|R_{s,k,i}^{e}(\boldsymbol{\phi}_{k,i-1})-R_{s,k,i}^{e}(w^{\star})\right\|
\end{aligned} \tag{11.20}$$

We next use the smoothness condition (11.11) to bound the right-most term as follows:

$$\|R^e_{s,k,i}(\boldsymbol{\phi}_{k,i-1}) - R^e_{s,i}(w^\star)\| \leq \kappa_d \|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^\gamma + \kappa'_d \|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^2 \tag{11.21}$$

where

$$\widetilde{\boldsymbol{\phi}}_{k,i-1} \triangleq w^\star - \boldsymbol{\phi}_{k,i-1} \tag{11.22}$$

Recall from the distributed algorithm (8.46) that

$$\widetilde{\boldsymbol{\phi}}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\, \widetilde{\boldsymbol{w}}_{\ell,i-1} \tag{11.23}$$

so that exploiting the convexity of the functions $f(x) = x^2$ and $f(x) = x^4$, and applying Jensen's inequality (F.26), we get:

$$\begin{aligned}
\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^2 &= \left\|\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\widetilde{\boldsymbol{w}}_{\ell,i-1}\right\|^2 \\
&\overset{\text{(F.26)}}{\leq} \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^2 \\
&\leq \sum_{\ell\in\mathcal{N}_k} \|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^2 \\
&\leq \sum_{\ell=1}^{N} \|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^2
\end{aligned} \tag{11.24}$$

Likewise, we have

$$\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^4 \leq \sum_{\ell=1}^{N} \|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^4 \tag{11.25}$$

and since the function $f(x) = x^{\gamma/4}$ is increasing over $x \geq 0$:

$$\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^\gamma = \left(\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^4\right)^{\gamma/4} \leq \left(\sum_{\ell=1}^{N} \|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^4\right)^{\gamma/4} \tag{11.26}$$

Substituting (11.24) and (11.26) into (11.21), we obtain

$$\|R^e_{s,k,i}(\boldsymbol{\phi}_{k,i-1}) - R^e_{s,i}(w^\star)\| \leq \kappa_d \left(\sum_{\ell=1}^{N} \|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^4\right)^{\gamma/4} + \kappa'_d \left(\sum_{\ell=1}^{N} \|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^2\right) \tag{11.27}$$

Using arguments similar to the steps that led to (4.31) in the single-agent case, we find under expectation and in the limit that:

$$\begin{aligned}
&\limsup_{i\to\infty} \mathbb{E}\,\|R^e_{s,k,i}(\boldsymbol{\phi}_{i-1}) - R^e_{s,k,i}(w^\star)\| \\
&\quad\leq \limsup_{i\to\infty} \left\{ \kappa_d\,\mathbb{E}\left(\sum_{\ell=1}^{N}\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^4\right)^{\gamma/4} + \kappa_d'\,\mathbb{E}\left(\sum_{\ell=1}^{N}\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^2\right)\right\} \\
&\quad\overset{\text{(a)}}{\leq} \limsup_{i\to\infty} \left\{ \kappa_d \left(\sum_{\ell=1}^{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^4\right)^{\gamma/4} + \kappa_d' \left(\sum_{\ell=1}^{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}\|^2\right)\right\} \\
&\quad\overset{(9.11)}{=} O(\mu_{\max}^{\gamma'/2})
\end{aligned} \tag{11.28}$$

where in step (a) we applied Jensen's inequality (F.30) to the function $f(x) = x^{\gamma/4}$; this function is concave over $x \geq 0$ for $\gamma \in (0, 4]$. Moreover, in the last step we called upon results (9.11) and (9.107), namely, that the second and fourth-order moments of $\widetilde{\boldsymbol{w}}_{\ell,i-1}$ are asymptotically bounded by $O(\mu_{\max})$ and $O(\mu_{\max}^2)$, respectively. Accordingly, the exponent γ' in the last step is given by

$$\gamma' \triangleq \min\{\gamma,\, 2\} \tag{11.29}$$

since $O(\mu_{\max}^{\gamma/2})$ dominates $O(\mu_{\max})$ for values of $\gamma \in (0, 2]$ and $O(\mu_{\max})$ dominates $O(\mu_{\max}^{\gamma/2})$ for values of $\gamma \in [2, 4]$. Substituting (11.28) into (11.20) gives

$$\limsup_{i\to\infty} \left\| \mathbb{E}\,\boldsymbol{s}^e_{k,i}(\boldsymbol{\phi}_{k,i-1})\left(\boldsymbol{s}^e_{k,i}(\boldsymbol{\phi}_{k,i-1})\right)^* - G_k \right\| = O(\mu_{\max}^{\gamma'/2}) \tag{11.30}$$

which leads to (11.14). Moreover, since for any square matrix X, it holds that $|\mathrm{Tr}(X)| \leq c\,\|X\|$, for some constant c that is independent of γ', we conclude that

$$\limsup_{i\to\infty} \left|\mathbb{E}\,\|\boldsymbol{s}^e_{k,i}(\boldsymbol{\phi}_{k,i-1})\|^2 - \mathrm{Tr}\,(G_k)\right| = O(\mu_{\max}^{\gamma'/2}) = b_1 \tag{11.31}$$

in terms of the absolute value of the difference. We are denoting the value of the limit superior by the nonnegative number b_1; we know from (11.31) that $b_1 = O(\mu^{\gamma'/2})$. The above relation then implies that, given $\epsilon > 0$, there exists an I_o large enough such that for all $i > I_o$ it holds that

$$\left|\,\mathbb{E}\,\|\boldsymbol{s}^e_{k,i}(\boldsymbol{\phi}_{k,i-1})\|^2 - \mathrm{Tr}(G_k)\,\right| \leq b_1 + \epsilon \tag{11.32}$$

If we select $\epsilon = O(\mu^{\gamma'/2})$ and introduce the sum $b_o = b_1 + \epsilon$, then we arrive at the desired result (11.15).

□

11.2 Performance Metrics

As was already explained in Sec. 4.5, besides the MSD metric (11.2)–(11.3), there is a second useful measure of performance defined in terms of the mean excess-cost; which is also called the *excess-risk* (ER). For multi-agent networks, this metric is usually of interest when the cost functions $J_k(w)$ across the agents are identical, i.e., when $J_k(w) \equiv J(w)$ and $H_k \equiv H$ for $k = 1, 2, \ldots, N$. In this case, the N agents would be cooperating to minimize the same strongly-convex cost function, $J^{\rm glob}(w) = N \cdot J(w)$, and the limit point $w^\star$ will coincide with the minimizer, w^o, of $J(w)$. We shall nevertheless define the ER metric more broadly for the general case when the individual costs may be different from each other.

For this purpose, we refer to the normalized aggregate cost, $\bar{J}^{{\rm glob},\star}(w)$, defined by (8.59) and whose global minimizer is the same $w^\star$. We already know from (11.1) that the iterates, $\boldsymbol{w}_{k,i}$, at the various agents approach $w^\star$ for sufficiently small step-sizes. We therefore define the ER measure for every agent k as the average fluctuation of $\bar{J}^{{\rm glob},\star}(w)$ around its minimum value (in a manner similar to what was defined earlier for the single-agent case in (4.95)):

$$\mathrm{ER}_{{\rm dist},k} \triangleq \tag{11.33}$$
$$\mu_{\max} \cdot \left(\lim_{\mu_{\max}\to 0} \limsup_{i\to\infty} \frac{1}{\mu_{\max}} \mathbb{E}\left\{ \bar{J}^{{\rm glob},\star}(\boldsymbol{w}_{k,i-1}) - \bar{J}^{{\rm glob},\star}(w^\star) \right\} \right)$$

The main difference in relation to (4.95) is that we are now scaling by $\mu_{\max}$ and using the normalized aggregate cost (8.59). The reason why we are using this normalized cost in (11.33), rather than the regular aggregate cost $J^{{\rm glob},\star}(w)$ from (9.6), is to ensure that the above definition of the excess-risk is compatible with the definition used earlier for non-cooperative agents in (4.95) and for centralized processing in (5.53). For example, when the individual costs happen to coincide, say, $J_k(w) \equiv J(w)$, then the expectation on the right-hand side of (11.33) reduces to $\mathbb{E}\{J(\boldsymbol{w}_{k,i-1}) - J(w^o)\}$, which is consistent with the earlier expression (4.95).

We further define the network ER measure as the average ER values across all agents:

$$\mathrm{ER}_{\mathrm{dist,av}} \triangleq \frac{1}{N}\left(\sum_{k=1}^{N}\mathrm{ER}_{\mathrm{dist},k}\right) \tag{11.34}$$

Using (9.107) and result (E.44) from the appendix, along with the same justification we employed earlier to arrive at (4.96), we can similarly express the ER measure (11.33) in terms of a weighted mean-square-error norm as follows:

$$\mathrm{ER}_{\mathrm{dist},k} = \mu_{\max}\cdot\left(\lim_{\mu_{\max}\to 0}\limsup_{i\to\infty}\frac{1}{\mu_{\max}}\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i-1}^{e}\|^2_{\frac{1}{2}\bar{H}}\right) \tag{11.35}$$

where the matrix $\bar{H}$ denotes the value of the Hessian matrix of the normalized cost, $\bar{J}^{\mathrm{glob},\star}(w)$, evaluated at $w=w^{\star}$. It follows from (8.59) that this matrix is given by

$$\bar{H} \triangleq \sum_{k=1}^{N}\bar{q}_k H_k \tag{11.36}$$

It is straightforward to verify that the MSD and ER performance measures defined so far can be equivalently expressed as follows in terms of the extended error vectors $\{\widetilde{\boldsymbol{w}}_{k,i}^{e},\widetilde{\boldsymbol{w}}_{i}^{e}\}$ defined by (8.133) and (8.143):

$$\mathrm{MSD}_{\mathrm{dist},k} \triangleq \mu_{\max}\cdot\left(\lim_{\mu_{\max}\to 0}\limsup_{i\to\infty}\frac{1}{\mu_{\max}}\frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}^{e}\|^2\right) \tag{11.37}$$

$$\mathrm{MSD}_{\mathrm{dist,av}} \triangleq \mu_{\max}\cdot\left(\lim_{\mu_{\max}\to 0}\limsup_{i\to\infty}\frac{1}{\mu_{\max}}\frac{1}{2N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i}^{e}\|^2\right) \tag{11.38}$$

$$\mathrm{ER}_{\mathrm{dist,av}} = \mu_{\max}\cdot\left(\lim_{\mu_{\max}\to 0}\limsup_{i\to\infty}\frac{1}{\mu_{\max}}\frac{1}{2N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^{e}\|^2_{(I_N\otimes\bar{H})}\right) \tag{11.39}$$

These expressions measure the mean-square-error performance of the network and its agents, as well as the mean fluctuation of the normalized aggregate cost function around its optimal value, in the steady-state regime assuming sufficiently small step-sizes. More specifically,

these expressions result in performance measures that are first-order in $\mu_{\max}$. We shall evaluate them by relying on the long-term model (10.19).

As explained earlier in Sec. 4.5, we sometimes write the expressions for the MSD and ER measures more compactly (but less rigorously) as follows for small step-sizes:

$$\mathrm{MSD}_{\mathrm{dist},k} \triangleq \lim_{i\to\infty} \frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}^{e}\|^2 \tag{11.40}$$

$$\mathrm{MSD}_{\mathrm{dist,av}} \triangleq \lim_{i\to\infty} \frac{1}{2N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i}^{e}\|^2 \tag{11.41}$$

$$\mathrm{ER}_{\mathrm{dist},k} = \lim_{i\to\infty} \frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i-1}^{e}\|_{\bar{H}}^2 \tag{11.42}$$

$$\mathrm{ER}_{\mathrm{dist,av}} = \lim_{i\to\infty} \frac{1}{2N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^{e}\|_{(I_N\otimes\bar{H})}^2 \tag{11.43}$$

with the understanding that the limits on the right-hand side are computed according to the definitions (11.35) and (11.37)–(11.39) since, strictly speaking, the limits in (11.40)–(11.43) may not exist. Yet, it is useful to note that derivations that assume the validity of these limits still lead to the same expressions for the MSD and ER to first-order in $\mu_{\max}$ as derivations that rely on the more formal expressions (11.35) and (11.37)–(11.39) — this fact can be verified by examining and repeating the proofs of Theorems 11.2 and 11.4 further ahead.

11.3 Mean-Square-Error Performance

We examine first the mean-square-error performance of the multi-agent network and derive closed-form expressions for the MSD measures of the individual agents and the entire network. The expressions given below involve the bvec and block Kronecker operations defined in Sec. F.1 in the appendix.

Theorem 11.2 (Network limiting performance). Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P = A_1 A_o A_2$. Assume the aggregate cost (9.10) and the individual costs,

$J_k(w)$, satisfy the conditions in Assumptions 6.1 and 10.1. Assume further that the first and fourth-order moments of the gradient noise process satisfy the conditions of Assumption 8.1 with the second-order moment condition (8.115) replaced by the fourth-order moment condition (8.121). Assume also (11.10). Let

$$\gamma_m \triangleq \frac{1}{2}\min\{1,\gamma\} > 0 \tag{11.44}$$

with $\gamma \in (0,4]$ from (11.10). Then, it holds that

$$\limsup_{i\to\infty} \frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}^{e}\|^2 = \frac{1}{h}\mathrm{Tr}(\mathcal{J}_k\mathcal{X}) + O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.45}$$

$$\limsup_{i\to\infty} \frac{1}{2N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i}^{e}\|^2 = \frac{1}{hN}\mathrm{Tr}(\mathcal{X}) + O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.46}$$

and, for large enough i, the convergence rate of the error variances, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, towards the steady-state region (11.45) is given by

$$\alpha = 1 - 2\lambda_{\min}\left(\sum_{k=1}^{N} q_k H_k\right) + O\left(\mu_{\max}^{(N+1)/N}\right) \tag{11.47}$$

where q_k is defined by (9.7) and $\alpha \in (0,1)$; the smaller the value of α is, the faster the convergence of $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ towards (11.45). Moreover, the matrix $\mathcal{X}$ that appears in (11.45)–(11.46) is Hermitian non-negative definite and corresponds to the unique solution of the (discrete-time) Lypaunov equation:

$$\mathcal{X} - \mathcal{B}\mathcal{X}\mathcal{B}^{*} = \mathcal{Y} \tag{11.48}$$

where the quantities $\{\mathcal{Y},\mathcal{B},\mathcal{J}_k\}$ are defined by:

$$\mathcal{A}_o = A_o \otimes I_{hM}, \quad \mathcal{A}_1 = A_1 \otimes I_{hM}, \quad \mathcal{A}_2 = A_2 \otimes I_{hM} \tag{11.49}$$

$$\mathcal{M} = \mathrm{diag}\{\,\mu_1 I_{hM}, \mu_2 I_{hM}, \ldots, \mu_N I_{hM}\} \tag{11.50}$$

$$\mathcal{H} = \mathrm{diag}\{\,H_1, H_2, \ldots, H_N\,\} \tag{11.51}$$

$$H_k = \nabla_w^2 J_k(w^\star) \tag{11.52}$$

$$\mathcal{S} = \mathrm{diag}\{\,G_1,\, G_2, \ldots, G_N\,\} \tag{11.53}$$

$$\mathcal{Y} = \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A}_2 \tag{11.54}$$

$$\mathcal{B} = \mathcal{A}_2^{\mathsf{T}}\left(\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\mathcal{H}\right)\mathcal{A}_1^{\mathsf{T}} \tag{11.55}$$

$$\mathcal{F} = \mathcal{B}^{\mathsf{T}} \otimes_b \mathcal{B}^{*} \tag{11.56}$$

$$\mathcal{J}_k = \mathrm{diag}\{\,0_{hM}, \ldots, 0_{hM}, I_{hM}, 0_{hM}, \ldots, 0_{hM}\,\} \tag{11.57}$$

with $\mathcal{J}_k$ having an identity matrix at the $k-$th diagonal block, and $h = 1$ for real data and $h = 2$ for complex data. Furthermore, the following are

equivalent characterizations for the matrix $\mathcal{X}$ or its trace:

$$\mathcal{X} = \sum_{n=0}^{\infty} \mathcal{B}^n \mathcal{Y} (\mathcal{B}^*)^n \tag{11.58}$$

$$\mathrm{bvec}(\mathcal{X}) = (I - \mathcal{F}^*)^{-1} \mathrm{bvec}(\mathcal{Y}) \tag{11.59}$$

$$\mathrm{Tr}(\mathcal{X}) = \left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}} (I - \mathcal{F})^{-1} \mathrm{bvec}\left(I_{hMN}\right) \tag{11.60}$$

$$\mathrm{Tr}(\mathcal{J}_k \mathcal{X}) = \left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}} (I - \mathcal{F})^{-1} \mathrm{bvec}\left(\mathcal{J}_k\right) \tag{11.61}$$

Proof. We start from the long-term model (10.19), namely,

$$\widetilde{\boldsymbol{w}}_i^{e'} = \mathcal{B}\,\widetilde{\boldsymbol{w}}_{i-1}^{e'} + \mathcal{A}_2^{\mathsf{T}} \mathcal{M} \boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) - \mathcal{A}_2^{\mathsf{T}} \mathcal{M} b^e \tag{11.62}$$

We drop the argument $\boldsymbol{w}_{i-1}^e$ from $\boldsymbol{s}_i^e$ for compactness of presentation. Conditioning on the past history and taking expectations gives

$$\mathbb{E}\left(\widetilde{\boldsymbol{w}}_i^{e'} \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right) = \mathcal{B}\,\widetilde{\boldsymbol{w}}_{i-1}^{e'} - \mathcal{A}_2^{\mathsf{T}} \mathcal{M} b^e \tag{11.63}$$

so that taking expectations again we obtain the mean recursion:

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i^{e'} = \mathcal{B}\left(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}^{e'}\right) - \mathcal{A}_2^{\mathsf{T}} \mathcal{M} b^e \tag{11.64}$$

Now observe that recursion (11.62) includes a constant driving term on the right-hand side represented by the factor $\mathcal{A}_2^{\mathsf{T}} \mathcal{M} b^e$. To facilitate the variance analysis, we introduce the centered variable:

$$\boldsymbol{z}_i \stackrel{\Delta}{=} \widetilde{\boldsymbol{w}}_i^{e'} - \mathbb{E}\,\widetilde{\boldsymbol{w}}_i^{e'} \tag{11.65}$$

Subtracting (11.64) from (11.62) we find that $\boldsymbol{z}_i$ satisfies the following recursion:

$$\boldsymbol{z}_i = \mathcal{B}\boldsymbol{z}_{i-1} + \mathcal{A}_2^{\mathsf{T}} \mathcal{M} \boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e) \tag{11.66}$$

where the deterministic driving terms are also removed. Although we are interested in evaluating the asymptotic size of $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}\|^2$, we can still rely on the centered variable $\boldsymbol{z}_i$ for this purpose. This is because it holds for $i \gg 1$:

$$\begin{aligned} \mathbb{E}\,\|\boldsymbol{z}_i\|^2 &= \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}\|^2 - \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i^{e'}\|^2 \\ &\overset{(10.108)}{=} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}\|^2 + O(\mu_{\max}^2) \end{aligned} \tag{11.67}$$

Moreover, we established earlier in (10.30) that under the fourth-order moment condition (8.121) on the gradient noise processes, the error variances $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e'}\|^2$ and $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^{e}\|^2$ are within $O(\mu_{\max}^{3/2})$ from each other. Therefore, we

may evaluate the network error variance (or MSD) in terms of the mean-square value of the variable $\boldsymbol{z}_i$ (similarly for any weighted square measure of $\widetilde{\boldsymbol{w}}_i^e$ such as the ER) by employing the correction:

$$\limsup_{i\to\infty} \frac{1}{2N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^2 \;=\; \limsup_{i\to\infty} \frac{1}{2N}\mathbb{E}\,\|\boldsymbol{z}_i\|^2 \;+\; O(\mu_{\max}^{3/2}) \tag{11.68}$$

We therefore continue with recursion (11.66) and proceed to examine how the mean-square value of $\boldsymbol{z}_i$ evolves over time by relying on energy conservation arguments [6, 205, 206, 269, 278].

Let Σ denote an arbitrary Hermitian positive semi-definite matrix that we are free to choose. Equating the squared weighted values of both sides of (11.66) and taking expectations conditioned on the past history gives:

$$\mathbb{E}\,\left(\|\boldsymbol{z}_i\|_{\Sigma}^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right) \;=\; \|\boldsymbol{z}_{i-1}\|_{\mathcal{B}^*\Sigma\mathcal{B}}^2 \;+\; \mathbb{E}\,\left(\|\boldsymbol{s}_i^e\|_{\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathsf{T}}\mathcal{M}}^2\,|\,\boldsymbol{\mathcal{F}}_{i-1}\right) \tag{11.69}$$

Taking expectations again removes the conditioning on $\boldsymbol{\mathcal{F}}_{i-1}$ and we get

$$\mathbb{E}\,\|\boldsymbol{z}_i\|_{\Sigma}^2 \;=\; \mathbb{E}\,\left(\|\boldsymbol{z}_{i-1}\|_{\mathcal{B}^*\Sigma\mathcal{B}}^2\right) \;+\; \mathbb{E}\,\left(\|\boldsymbol{s}_i^e\|_{\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathsf{T}}\mathcal{M}}^2\right) \tag{11.70}$$

We now evaluate the right-most term. For that purpose, we shall call upon the results of Lemma 11.1. To begin with, note that

$$\mathbb{E}\,\left(\|\boldsymbol{s}_i^e\|_{\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathsf{T}}\mathcal{M}}^2\right) \;=\; \mathrm{Tr}\left[\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\,\mathbb{E}\,\left(\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)\,\left(\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)\right)^*\right)\right] \tag{11.71}$$

where the entries of the covariance matrix $\mathbb{E}\,\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)\,\left(\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)\right)^*$ that appears in the above expression were already evaluated earlier in (11.14). Using that result, and the fact that the gradient noises across the agents are uncorrelated with each other and second-order circular, we obtain

$$\limsup_{i\to\infty}\,\left\|\mathbb{E}\,\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)\,\left(\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)\right)^* \;-\; \mathcal{S}\right\| \;=\; O(\mu_{\max}^{\gamma'/2}) \tag{11.72}$$

where γ' was defined in (11.29) as $\gamma' = \min\{\gamma, 2\}$. Using the sub-multiplicative property of norms, namely, $\|AB\| \le \|A\|\,\|B\|$, we conclude from (11.72) that

$$\begin{aligned}\limsup_{i\to\infty}\,&\left\|\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\,\left(\mathbb{E}\,\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)\,\left(\boldsymbol{s}_i^e(\boldsymbol{w}_{i-1}^e)\right)^* - \mathcal{S}\right)\right\| \\ &= \mathrm{Tr}(\Sigma)\cdot O\left(\mu_{\max}^{2+(\gamma'/2)}\right)\end{aligned} \tag{11.73}$$

where an additional factor $\mu_{\max}^2$ has been added to the big-O term; it arises from the fact that $\|\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\| = \mathrm{Tr}(\Sigma)\cdot O(\mu_{\max}^2)$. Note that we are keeping the factor $\mathrm{Tr}(\Sigma)$ explicit on the right-hand side of (11.73); this is convenient

for later use in (11.92) — the reason we have $\mathrm{Tr}(\Sigma)$ in (11.73) is because $\|\Sigma\| \leq \mathrm{Tr}(\Sigma)$ for any Hermitian positive semi-definite Σ. Using again the fact that $|\mathrm{Tr}(X)| \leq c\,\|X\|$ for any square matrix X, we conclude that

$$\limsup_{i\to\infty} \left|\mathbb{E}\,\|\boldsymbol{s}_i^e\|^2_{\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathsf{T}}\mathcal{M}} - \mathrm{Tr}\left(\Sigma\mathcal{Y}\right)\right| \;=\; \mathrm{Tr}(\Sigma)\cdot O\left(\mu_{\max}^{2+(\gamma'/2)}\right) \;=\; b_1 \tag{11.74}$$

in terms of the absolute value of the difference and where we are denoting the value of the limit superior by the nonnegative number b_1; we know from (11.74) that $b_1 = \mathrm{Tr}(\Sigma)\cdot O(\mu_{\max}^{2+(\gamma'/2)})$. The same argument that led to (11.15) then gives for $i \gg 1$:

$$\mathrm{Tr}(\Sigma\mathcal{Y}) \;-\; b_o \;\leq\; \mathbb{E}\left(\|\boldsymbol{s}_i^e\|^2_{\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathsf{T}}\mathcal{M}}\right) \;\leq\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;+\; b_o \tag{11.75}$$

for some nonnegative constant $b_o = \mathrm{Tr}(\Sigma)\cdot O(\mu_{\max}^{2+(\gamma'/2)})$. It follows from (11.75) that we can also write for $i \gg 1$:

$$\mathbb{E}\left(\|\boldsymbol{s}_i^e\|^2_{\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathsf{T}}\mathcal{M}}\right) \;=\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;+\; \mathrm{Tr}(\Sigma)\cdot O\left(\mu_{\max}^{2+(\gamma'/2)}\right) \tag{11.76}$$

Substituting (11.75) into (11.70) we obtain for $i \gg 1$:

$$\mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma} \;\leq\; \mathbb{E}\left(\|\boldsymbol{z}_{i-1}\|^2_{\mathcal{B}^*\Sigma\mathcal{B}}\right) \;+\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;+\; b_o \tag{11.77}$$

$$\mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma} \;\geq\; \mathbb{E}\left(\|\boldsymbol{z}_{i-1}\|^2_{\mathcal{B}^*\Sigma\mathcal{B}}\right) \;+\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;-\; b_o \tag{11.78}$$

Using the sub-additivity and super-additivity properties (4.117)–(4.118) of the limit superior and limit inferior operations, we conclude from the above relations that:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma} \;\leq\; \limsup_{i\to\infty} \mathbb{E}\left(\|\boldsymbol{z}_{i-1}\|^2_{\mathcal{B}^*\Sigma\mathcal{B}}\right) \;+\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;+\; b_o \tag{11.79}$$

$$\liminf_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma} \;\geq\; \liminf_{i\to\infty} \mathbb{E}\left(\|\boldsymbol{z}_{i-1}\|^2_{\mathcal{B}^*\Sigma\mathcal{B}}\right) \;+\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;-\; b_o \tag{11.80}$$

Grouping terms we get:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma-\mathcal{B}^*\Sigma\mathcal{B}} \;\leq\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;+\; b_o \tag{11.81}$$

$$\liminf_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma-\mathcal{B}^*\Sigma\mathcal{B}} \;\geq\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;-\; b_o \tag{11.82}$$

and, consequently, by using the fact that the limit inferior of a sequence is upper bounded by its limit superior, we obtain the following inequality relation:

$$\begin{aligned}\mathrm{Tr}(\Sigma\mathcal{Y}) \;-\; b_o \;&\leq\; \liminf_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma-\mathcal{B}^*\Sigma\mathcal{B}} \\ &\leq\; \limsup_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma-\mathcal{B}^*\Sigma\mathcal{B}} \;\leq\; \mathrm{Tr}(\Sigma\mathcal{Y}) + b_o \quad (11.83)\end{aligned}$$

Recalling that $b_o = \mathrm{Tr}(\Sigma)\cdot O(\mu_{\max}^{2+(\gamma'/2)})$, we conclude that the limit superior and limit inferior of the error variance satisfy:

$$\limsup_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma-\mathcal{B}^*\Sigma\mathcal{B}} \;=\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;+\; \mathrm{Tr}(\Sigma)\cdot O\left(\mu_{\max}^{2+(\gamma'/2)}\right) \quad (11.84)$$

$$\liminf_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\Sigma-\mathcal{B}^*\Sigma\mathcal{B}} \;=\; \mathrm{Tr}(\Sigma\mathcal{Y}) \;-\; \mathrm{Tr}(\Sigma)\cdot O\left(\mu_{\max}^{2+(\gamma'/2)}\right) \quad (11.85)$$

We can now use (11.84) to justify (11.46). To do so, it is useful to review first two properties of block Kronecker products, which will be used in the derivation.

Thus, consider an arbitrary square matrix C with block entries, say, of size $hM\times hM$ each. We let the notation $\mathrm{bvec}(C)$ denote the vector that is obtained by vectorizing each block entry of the matrix and then stacking the resulting columns on top of each other — see expression (F.5) in the appendix. It is then well-known that the following properties from Table F.2 in the appendix hold for arbitrary matrices $\{U, W, C\}$ with block entries of compatible dimensions and in terms of the block Kronecker product operation defined by (F.2) in the same appendix:

$$\mathrm{bvec}(UCW) \;=\; (W^{\mathsf{T}} \otimes_b U)\mathrm{bvec}(C) \quad (11.86)$$

$$\mathrm{Tr}(CW) \;=\; \left(\mathrm{bvec}(W^{\mathsf{T}})\right)^{\mathsf{T}} \mathrm{bvec}(C) \quad (11.87)$$

Returning to (11.84), we recall that we are free to choose the weighting matrix Σ. Assume we select Σ as the solution to the following (discrete-time) Lyapunov equation:

$$\Sigma - \mathcal{B}^*\Sigma\mathcal{B} \;=\; I_{hMN} \quad (11.88)$$

We know from (9.173) that the matrix $\mathcal{B}$ is stable for sufficiently small step-sizes. Accordingly, we are guaranteed from the statement of Lemma F.2 that the above Lyapunov equation has a unique solution Σ and, moreover, this solution is Hermitian and non-negative definite, as desired. The advantage of this choice for Σ is that it reduces the weighting matrix on the mean-square value of $\boldsymbol{z}_i$ in (11.84) to the identity matrix. We can then focus on evaluating the value of the right-hand side of expression (11.84).

For this purpose, we start by applying the block vectorization operation to both sides of (11.88) and use (11.86) to find that

$$\mathrm{bvec}(\Sigma) - \left(\mathcal{B}^{\mathsf{T}} \otimes_b \mathcal{B}^*\right)\mathrm{bvec}(\Sigma) \;=\; \mathrm{bvec}(I_{hMN}) \quad (11.89)$$

so that in terms of the matrix $\mathcal{F}$ defined by (11.56), which is also stable, we can write

$$\text{bvec}(\Sigma) \;=\; (I-\mathcal{F})^{-1}\text{bvec}(I_{hMN}) \tag{11.90}$$

Now, substituting this Σ into (11.84), we obtain $\mathbb{E}\,\|\boldsymbol{z}_i\|^2$ on the left-hand side while the term $\text{Tr}(\Sigma\mathcal{Y})$ on the right-hand side becomes:

$$\begin{aligned}\text{Tr}(\Sigma\mathcal{Y}) &\overset{(11.87)}{=} \left(\text{bvec}(\mathcal{Y}^{\mathsf{T}})\right)^{\mathsf{T}}\text{bvec}(\Sigma)\\ &= \left(\text{bvec}(\mathcal{Y}^{\mathsf{T}})\right)^{\mathsf{T}}(I-\mathcal{F})^{-1}\text{bvec}(I_{hMN})\end{aligned} \tag{11.91}$$

Likewise, the second term on the right-hand side of (11.84) becomes:

$$\begin{aligned}O\left(\mu_{\max}^{2+(\gamma'/2)}\right)\cdot\text{Tr}(\Sigma) &\overset{(11.87)}{=} O\left(\mu_{\max}^{2+(\gamma'/2)}\right)\cdot(\text{bvec}(I_{hMN}))^{\mathsf{T}}\,\text{bvec}(\Sigma)\\ &= O\left(\mu_{\max}^{2+(\gamma'/2)}\right)\cdot(\text{bvec}(I_{hMN}))^{\mathsf{T}}(I-\mathcal{F})^{-1}\text{bvec}(I_{hMN})\end{aligned} \tag{11.92}$$

But since $\mathcal{F}$ is a stable matrix, we can employ the expansion

$$\begin{aligned}(I-\mathcal{F})^{-1} &= I+\mathcal{F}+\mathcal{F}^2+\mathcal{F}^3+\ldots\\ &\overset{(11.56)}{=} I+\left(\mathcal{B}^{\mathsf{T}}\otimes_b\mathcal{B}^*\right)+\left(\left(\mathcal{B}^{\mathsf{T}}\right)^2\otimes_b\left(\mathcal{B}^*\right)^2\right)+\ldots\end{aligned} \tag{11.93}$$

and appeal to properties (11.86) and (11.87) again, to validate the identities:

$$\left[\text{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right]^{\mathsf{T}}(I-\mathcal{F})^{-1}\text{bvec}\left(I_{hMN}\right) \;=\; \sum_{n=0}^{\infty}\text{Tr}\left[\mathcal{B}^n\mathcal{Y}(\mathcal{B}^*)^n\right] \tag{11.94}$$

$$(\text{bvec}(I_{hMN}))^{\mathsf{T}}(I-\mathcal{F})^{-1}\text{bvec}(I_{hMN}) \;=\; \sum_{n=0}^{\infty}\text{Tr}\left[(\mathcal{B}^*)^n\mathcal{B}^n\right] \tag{11.95}$$

The two series that appear in the above expressions converge to the trace values of certain Lyapunov solutions. To see this, let

$$\mathcal{X}' \;\triangleq\; \sum_{n=0}^{\infty}(\mathcal{B}^*)^n\,\mathcal{B}^n, \qquad \mathcal{X} \;\triangleq\; \sum_{n=0}^{\infty}\mathcal{B}^n\mathcal{Y}\,(\mathcal{B}^*)^n \tag{11.96}$$

Then, these series correspond, respectively, to the unique solutions of the following Lyapunov equations (cf. Lemma F.2 from the appendix):

$$\mathcal{X}'-\mathcal{B}^*\mathcal{X}'\mathcal{B} \;=\; I_{hMN} \tag{11.97}$$

$$\mathcal{X}-\mathcal{B}\mathcal{X}\mathcal{B}^* \;=\; \mathcal{Y} \tag{11.98}$$

Moreover, the matrices $\mathcal{X}$ and $\mathcal{X}'$ so defined are Hermitian and nonnegative-definite (note for $\mathcal{X}$ that the matrix $\mathcal{Y}$ defined by (11.54) is Hermitian and non-negative definite). Therefore, we have established so far that

$$\limsup_{i\to\infty} \mathbb{E}\,\|\boldsymbol{z}_i\|^2 = \mathrm{Tr}(\mathcal{X}) \;+\; \mathrm{Tr}(\mathcal{X}')\cdot O\left(\mu_{\max}^{2+(\gamma'/2)}\right) \tag{11.99}$$

We now verify that $\mathrm{Tr}(\mathcal{X}') = O(1/\mu_{\max})$ — see (11.103); this result will permit us to assess the size of the second term on the right-hand side of (11.99) — see (11.104).

Applying the bvec operation to both sides of (11.97) and using (11.86) we find that

$$\mathrm{bvec}(\mathcal{X}') = (I-\mathcal{F})^{-1}\mathrm{bvec}(I) \tag{11.100}$$

Then,

$$\begin{aligned}
\|\mathrm{bvec}(\mathcal{X}')\| &\leq \left\|(I-\mathcal{F})^{-1}\right\|\,\|\mathrm{bvec}(I)\| \\
&\overset{\text{(a)}}{\leq} r\cdot\left\|(I-\mathcal{F})^{-1}\right\|_1\,\|\mathrm{bvec}(I)\| \\
&\overset{(9.243)}{=} O(1/\mu_{\max})
\end{aligned} \tag{11.101}$$

where in step (a) we used a positive constant r to account for the fact that matrix norms are equivalent (cf. (F.6) in the appendix). We can use this result to bound the trace of $\mathcal{X}'$ as follows.

Let $L\times L$ denote the dimensions of $\mathcal{X}'$; we know that $L = hNM$. Let further $\{x'_{nn},\, n=1,2,\ldots,L\}$ denote the diagonal entries of $\mathcal{X}'$. Since $\mathcal{X}'\geq 0$, we know that $x'_{nn}\geq 0$. We collect the diagonal entries of $\mathcal{X}'$ into the column vector $b = \mathrm{col}\{x'_{nn}\}$. Then, for any two vectors a and b of compatible dimensions, we use the Cauchy-Schwartz inequality $(a^*b)^2 \leq \|a\|^2\,\|b\|^2$ to conclude that

$$\begin{aligned}
\left(\mathrm{Tr}(\mathcal{X}')\right)^2 &\triangleq \left(\sum_{n=1}^{L} x'_{nn}\right)^2 \\
&= \left(\mathbb{1}^{\mathsf{T}}b\right)^2 \\
&\leq \|\mathbb{1}\|^2\,\|b\|^2 \\
&= L\cdot\|b\|^2 \\
&\leq L\cdot\|\mathrm{bvec}(\mathcal{X}')\|^2 \\
&\overset{(11.101)}{=} O(1/\mu_{\max}^2)
\end{aligned} \tag{11.102}$$

and, therefore,

$$\mathrm{Tr}(\mathcal{X}') \;=\; O\left(1/\mu_{\max}\right) \tag{11.103}$$

It follows that

$$O\left(\mu_{\max}^{2+(\gamma'/2)}\right)\cdot \mathrm{Tr}(\mathcal{X}') \;=\; O\left(\mu_{\max}^{1+(\gamma'/2)}\right) \tag{11.104}$$

Returning to (11.99), we conclude that

$$\limsup_{i\to\infty}\; \mathbb{E}\,\|\boldsymbol{z}_i\|^2 \;=\; \mathrm{Tr}(\mathcal{X}) \;+\; O\left(\mu_{\max}^{1+(\gamma'/2)}\right) \tag{11.105}$$

and, consequently, using (11.68), we obtain the following two equivalent characterizations for the network MSD:

$$\limsup_{i\to\infty}\frac{1}{2N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^2 \;=\; \frac{1}{2N}\mathrm{Tr}(\mathcal{X}) \;+\; O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.106}$$

$$=\; \frac{1}{2N}\sum_{n=0}^{\infty}\mathrm{Tr}\left[\mathcal{B}^n\mathcal{Y}(\mathcal{B}^*)^n\right] + O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.107}$$

with γ_m replacing $\gamma'/2$. These results, along with the arguments leading to them, justify expressions (11.46) and (11.58)–(11.60). Observe in particular from (11.54) and (9.243) that the term on the left-hand side of (11.94) is $O(\mu_{\max})$ since $\|\mathcal{Y}\| = O(\mu_{\max}^2)$ and $\|(I-\mathcal{F})^{-1}\| = O(1/\mu_{\max})$. Therefore, the value of $\mathrm{Tr}(\mathcal{X})$ in (11.60) is $O(\mu_{\max})$, which dominates the factor $O(\mu_{\max}^{1+\gamma_m})$.

Similarly, if we start from (11.85) instead, and apply the same arguments we would arrive at the following equivalent expressions:

$$\liminf_{i\to\infty}\frac{1}{2N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^2 \;=\; \frac{1}{2N}\mathrm{Tr}(\mathcal{X}) \;-\; O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.108}$$

$$=\; \frac{1}{2N}\sum_{n=0}^{\infty}\mathrm{Tr}\left[\mathcal{B}^n\mathcal{Y}(\mathcal{B}^*)^n\right] - O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.109}$$

This last result is not needed in the current derivation but is referred to later in Example 11.7.

We can also assess the mean-square performance of the *individual* agents in the network from (11.77). Let us introduce the $N\times N$ block diagonal matrix $\mathcal{J}_k$ defined by (11.57) with blocks of size $hM\times hM$, where all blocks on the diagonal are zero except for an identity matrix on the diagonal block of index k. Then, the error variance for agent k satisfies:

$$\limsup_{i\to\infty}\;\frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^2_{\mathcal{J}_k} \;=\; \limsup_{i\to\infty}\;\frac{1}{2}\mathbb{E}\,\|\boldsymbol{z}_i\|^2_{\mathcal{J}_k} \;+\; O(\mu_{\max}^{3/2}) \tag{11.110}$$

The same argument that was used to obtain expression (11.46) for the network mean-square-error can then be repeated to give (11.45) and (11.61).

With regards to the convergence rate of $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ towards the region (11.45), we substitute (11.76) into (11.70) to write for $i \gg 1$:

$$\mathbb{E}\,\|\boldsymbol{z}_i\|_{\Sigma}^2 \;=\; \mathbb{E}\,\left(\|\boldsymbol{z}_{i-1}\|_{\mathcal{B}^*\Sigma\mathcal{B}}^2\right) + \mathrm{Tr}(\Sigma\mathcal{Y}) + \mathrm{Tr}(\Sigma)\cdot O\left(\mu_{\max}^{2+(\gamma'/2)}\right) \tag{11.111}$$

Selecting the origin of time at some large time and iterating from there:

$$\mathbb{E}\,\|\boldsymbol{z}_i\|^2 \;=\; \mathbb{E}\,\|\boldsymbol{z}_{-1}\|^2_{(\mathcal{B}^*)^{i+1}\mathcal{B}^{i+1}} \;+\; \sum_{n=0}^{i}\mathrm{Tr}\left[\mathcal{B}^n\mathcal{Y}(\mathcal{B}^*)^n\right] \;+\; o(\mu^2) \tag{11.112}$$

The first-term on the right-hand side corresponds to a transient component that dies out with time. The rate of its convergence towards zero determines the rate of convergence of $\mathbb{E}\,\|\boldsymbol{z}_i\|^2$ towards its steady-state region. This rate can be characterized as follows. Note that, using properties (11.86)–(11.87) for block Kronecker products, we can express the weighted variance of $\boldsymbol{z}_{-1}$ as the following trace relation in terms of its un-weighted covariance matrix, which we denote by $R_z = \mathbb{E}\,\boldsymbol{z}_{-1}\boldsymbol{z}_{-1}^*$:

$$\begin{aligned}
\mathbb{E}\,\|\boldsymbol{z}_{-1}\|^2_{(\mathcal{B}^*)^{i+1}\mathcal{B}^{i+1}} &= \mathbb{E}\,\left(\boldsymbol{z}_{-1}^*\,(\mathcal{B}^*)^{i+1}\,\mathcal{B}^{i+1}\boldsymbol{z}_{-1}\right)\\
&= \mathrm{Tr}\left((\mathcal{B}^*)^{i+1}\,\mathcal{B}^{i+1}R_z\right)\\
&\overset{(11.87)}{=} \left[\mathrm{bvec}\left(R_z^{\mathsf{T}}\right)\right]^{\mathsf{T}}\mathrm{bvec}\left((\mathcal{B}^*)^{i+1}\,\mathcal{B}^{i+1}\right)\\
&\overset{(11.86)}{=} \left[\mathrm{bvec}(R_z^{\mathsf{T}})\right]^{\mathsf{T}}\left((\mathcal{B}^{\mathsf{T}})^{i+1}\otimes_b(\mathcal{B}^*)^{i+1}\right)\mathrm{bvec}(I)
\end{aligned} \tag{11.113}$$

It is clear now that the convergence rate of the transient component is dictated by the spectral radius of the matrix multiplying $\mathrm{bvec}(I)$, namely, by

$$\rho\left((\mathcal{B}^{\mathsf{T}})^{i+1}\otimes_b(\mathcal{B}^*)^{i+1}\right) \;=\; \left([\rho(\mathcal{B})]^2\right)^{i+1} \tag{11.114}$$

We conclude that the convergence rate of $\mathbb{E}\,\|\boldsymbol{z}_i\|^2$ towards the steady-state regime is dictated by $[\rho(\mathcal{B})]^2$ since this value characterizes the slowest rate at which the transient term dies out. Therefore, using (9.173) and the relation $(1-x)^2 = 1-2x+O(x^2)$, we can approximate the convergence rate to first-order in μ as follows:

$$\begin{aligned}
[\rho(\mathcal{B})]^2 &= \left[1-\lambda_{\min}\left(\sum_{k=1}^{N}q_kH_k\right) \;+\; O\left(\mu_{\max}^{(N+1)/N}\right)\right]^2\\
&= 1-2\lambda_{\min}\left(\sum_{k=1}^{N}q_kH_k\right) \;+\; O\left(\mu_{\max}^{(N+1)/N}\right)
\end{aligned} \tag{11.115}$$

□

Example 11.1 (Steady-state region for MSE networks)**.** Let us consider the case of MSE networks, defined earlier in Example 6.3, where the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ satisfy the linear regression model (6.14) and where the cost function associated with each agent is the mean-square-error cost, $J_k(w) = \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2$.

We showed in Example 6.1 that in this case, all individual costs are minimized at the same location w^o. It follows that the reference vectors w^o and $w^\star$ will coincide and, therefore, the bias vector b^e that appears in the error recursion (10.2) will be zero (as is evident from the definition of its entries in (8.136)). Moreover, the matrices $\boldsymbol{H}_{k,i-1}$ and H_k defined by (10.6) and (10.9), respectively, will coincide with each other since the Hessian matrix $\nabla_w^2 J_k(w)$ will be constant for all w. Thus, in this case, we get:

$$\boldsymbol{H}_{k,i-1} \;\equiv\; H_k \;=\; \nabla_w^2 J_k(w^o) \tag{11.116}$$

As a result, the perturbation term $\boldsymbol{c}_{i-1}$ in (10.13) will be identically zero and recursions (10.13) and (10.19) will therefore coincide (including having $b^e = 0$). Both models (i.e., the actual error recursion and the long-term error recursion) will then have the same MSD expressions. Therefore, we can rely on expression (11.68) without the need for the additional error factor $O(\mu_{\max}^{3/2})$. We know from the earlier result (4.16) that $\gamma = 2$ for mean-square-error costs. Using this value for γ in the derivation leading to (11.107), and ignoring the correction by $O(\mu_{\max}^{3/2})$, we arrive instead at

$$\limsup_{i\to\infty} \frac{1}{2N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i^e\|^2 \;=\; \frac{1}{2N}\sum_{n=0}^{\infty} \mathrm{Tr}\left[\mathcal{B}^n \mathcal{Y} (\mathcal{B}^*)^n\right] \;+\; O\left(\mu_{\max}^2\right) \tag{11.117}$$

with an approximation error in the order of $O(\mu_{\max}^2)$ rather than the term $O(\mu_{\max}^{3/2})$ that would result from (11.107) if we use $\gamma_m = 1/2$. We conclude that for MSE networks, the results of Theorem 11.2 are valid with the approximation error $O(\mu_{\max}^{1+\gamma_m})$ in (11.45)–(11.46) replaced by the smaller factor $O(\mu_{\max}^2)$.

■

MSD Performance

We now use the result of Theorem 11.2 to derive an expression for the MSD performance of each agent and for the entire network. We will do so by appealing to the useful low-rank approximation (9.244). Two observations are in place in relation to the forthcoming result (11.118). First, observe from (11.118) the interesting conclusion that the consensus and diffusion strategies represented by (8.46) are able to equalize the MSD performance across all agents for sufficiently small step-sizes.

This is a reassuring property since it means that all agents, regardless of the quality of their data, will end up achieving similar performance levels. At the same time, we remark that although expression (11.118) suggests that the performance of consensus and diffusion strategies match to first-order in $\mu_{\max}$, differences in performance actually occur for larger step-sizes with ATC diffusion exhibiting superior performance. These differences are illustrated and explained further ahead in Example 11.4, and also Examples 11.11–11.13.

Lemma 11.3 (Network MSD performance). Under the same conditions of Theorem 11.2, it holds that

$$\mathrm{MSD}_{\mathrm{dist},k} \;=\; \mathrm{MSD}_{\mathrm{dist,av}} = \frac{1}{2h}\mathrm{Tr}\left[\left(\sum_{k=1}^{N} q_k H_k\right)^{-1}\left(\sum_{k=1}^{N} q_k^2 G_k\right)\right] \tag{11.118}$$

where $h = 1$ for real data and $h = 2$ for complex data.

Proof. We establish the result for $h = 2$ without loss of generality by extending the argument from [71, 278] to the current context. According to definition (11.37), and expressions (11.45) and (11.61), we need to evaluate the following limit:

$$\mathrm{MSD}_{\mathrm{dist},k} = \mu_{\max}\cdot\left(\lim_{\mu_{\max}\to 0}\limsup_{i\to\infty}\frac{1}{\mu_{\max}}\frac{1}{h}\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}}(I-\mathcal{F})^{-1}\mathrm{bvec}\left(\mathcal{J}_k\right)\right) \tag{11.119}$$

We focus on the rightmost factor inside the above expression. Using (9.244), along with the first line in (9.275), we get:

$$\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}}(I-\mathcal{F})^{-1}\mathrm{bvec}(\mathcal{J}_k) \;=\; O(\mu_{\max}^2) + \tag{11.120}$$
$$\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}}(p\otimes I_{2M})\otimes_b(p\otimes I_{2M})\,Z^{-1}\left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right)\otimes_b\left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right)\mathrm{bvec}\left(\mathcal{J}_k\right)$$

Using the Kronecker product property (11.86), it is straightforward to verify that the last three terms combine into the following result, where the bvec operation is relative to blocks of size $2M\times 2M$:

$$\left[\left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right)\otimes_b\left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right)\right]\mathrm{bvec}(\mathcal{J}_k) = \mathrm{vec}(I_{2M}) \tag{11.121}$$

with the rightmost term involving the traditional (not block) vec operator. Let us therefore evaluate the matrix vector product:

$$x \;\triangleq\; Z^{-1}\mathrm{vec}(I_{2M}) \tag{11.122}$$

This vector is the unique solution to the linear system of equations

$$Zx = \mathrm{vec}(I_{2M}) \tag{11.123}$$

or, equivalently, by using definition (9.245) for Z:

$$\left(\sum_{k=1}^{N} q_k(I_{2M} \otimes H_k)\right) x \;+\; \left(\sum_{k=1}^{N} q_k(H_k^{\mathsf{T}} \otimes I_{2M})\right) x \;=\; \mathrm{vec}(I_{2M}) \tag{11.124}$$

Let $X = \mathrm{unvec}(x)$ denote the $2M \times 2M$ matrix whose vector representation is x. Applying to each of the terms appearing on the left-hand side of the above expression the Kronecker product property (11.87), albeit using vec instead of bvec operations, namely,

$$\mathrm{vec}(UCW) \;=\; (W^{\mathsf{T}} \otimes U)\mathrm{vec}(C) \tag{11.125}$$

we find that

$$\left(\sum_{k=1}^{N} q_k(I_{2M} \otimes H_k)\right) x \;=\; \mathrm{vec}\left\{\left(\sum_{k=1}^{N} q_k H_k\right) X\right\} \tag{11.126}$$

$$\left(\sum_{k=1}^{N} q_k(H_k^{\mathsf{T}} \otimes I_{2M})\right) x \;=\; \mathrm{vec}\left\{X\left(\sum_{k=1}^{N} q_k H_k\right)\right\} \tag{11.127}$$

We conclude from these equalities and from (11.124) that X is the unique solution to the (continuous-time) Lyapunov equation (cf. Lemma F.3 from the appendix):

$$\left(\sum_{k=1}^{N} q_k H_k\right) X \;+\; X\left(\sum_{k=1}^{N} q_k H_k\right) \;=\; I_{2M} \tag{11.128}$$

It is straightforward to verify that the solution X is given by

$$X = \frac{1}{2}\left(\sum_{k=1}^{N} q_k H_k\right)^{-1} \tag{11.129}$$

Therefore, substituting into (11.120) gives

$$\begin{aligned} &\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}} (I - \mathcal{F})^{-1}\mathrm{bvec}(\mathcal{J}_k) \;= \\ &\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}} \left[(p \otimes I_{2M}) \otimes_b (p \otimes I_{2M})\right] \mathrm{vec}(X) \;+\; O(\mu_{\max}^2) \end{aligned} \tag{11.130}$$

Using the Kronecker product properties (11.87) and (11.125) again, we obtain

$$
\begin{aligned}
&\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}}\left[(p\otimes I_{2M})\otimes_b(p\otimes I_{2M})\right]\mathrm{vec}(X)\\
&\quad=\quad \mathrm{Tr}\left[\mathrm{unbvec}\left\{(p\otimes I_{2M})\otimes_b(p\otimes I_{2M})\,\mathrm{vec}(X)\right\}\mathcal{Y}\right]\\
&\quad=\quad \mathrm{Tr}\left[(p\otimes I_{2M})\,X\,\left(p^{\mathsf{T}}\otimes I_{2M}\right)\mathcal{Y}\right]\\
&\quad=\quad \mathrm{Tr}\left[\left(p^{\mathsf{T}}\otimes I_{2M}\right)\mathcal{A}_2^{\mathsf{T}}\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A}_2\,(p\otimes I_{2M})\,X\right]\\
&\quad=\quad \mathrm{Tr}\left[\left(q^{\mathsf{T}}\otimes I_{2M}\right)\mathcal{S}\,(q\otimes I_{2M})\,X\right]\\
&\quad\overset{(11.129)}{=}\quad \frac{1}{2}\mathrm{Tr}\left[\left(\sum_{k=1}^{N}q_kH_k\right)^{-1}\left(\sum_{k=1}^{N}q_k^2G_k\right)\right]
\end{aligned}
\tag{11.131}
$$

Grouping terms we conclude that:

$$
\begin{aligned}
&\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}}(I-\mathcal{F})^{-1}\mathrm{bvec}(\mathcal{J}_k)\\
&\quad=\quad \frac{1}{2}\mathrm{Tr}\left[\left(\sum_{k=1}^{N}q_kH_k\right)^{-1}\left(\sum_{k=1}^{N}q_k^2G_k\right)\right]\;+\;O(\mu_{\max}^2)
\end{aligned}
\tag{11.132}
$$

We know from the definition of the scalars $\{q_k\}$ in (9.7) that each q_k is proportional to $\mu_{\max}$. Therefore, the first term on the right-hand side of the above expression is linear in $\mu_{\max}$. Now substituting (11.132) into the right-hand side of (11.119) and computing the limit as $\mu_{\max}\to 0$, we arrive at expression (11.118) for the performance of the individual agents. Since this expression is independent of the index of the agent, by averaging over all agents, we find that the network performance is given by the same expression.

□

Example 11.2 (MSD performance of consensus and diffusion networks)**.** We specialize the main result of Lemma 11.3 to the consensus and diffusion strategies, which correspond to the choices $\{A_o, A_1, A_2\}$ shown earlier in (8.7)–(8.10) in terms of a single combination matrix A, namely,

$$
\begin{aligned}
\text{consensus:}\qquad & A_o=A,\quad A_1=I_N=A_2 && (11.133)\\
\text{CTA diffusion:}\qquad & A_1=A,\quad A_2=I_N=A_o && (11.134)\\
\text{ATC diffusion:}\qquad & A_2=A,\quad A_1=I_N=A_o && (11.135)
\end{aligned}
$$

In these cases, the Perron eigenvector p defined by (9.9) will correspond to the Perron eigenvector associated with A:

$$
Ap=p,\quad \mathbb{1}^{\mathsf{T}}p=1,\quad p_k>0 \tag{11.136}
$$

Consequently, the entries q_k defined by (9.7) will reduce to

$$q_k \;=\; \mu_k p_k \tag{11.137}$$

Using these facts in (11.118) we obtain

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{1}{2h}\mathrm{Tr}\left[\left(\sum_{k=1}^{N}\mu_k p_k H_k\right)^{-1}\left(\sum_{k=1}^{N}\mu_k^2 p_k^2 G_k\right)\right] \tag{11.138}$$

where $h = 1$ for real data and $h = 2$ for complex data. Moreover, the convergence rate of the error variances, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, towards this MSD value is determined by

$$\alpha_{\mathrm{dist}} \;=\; 1 - 2\lambda_{\min}\left(\sum_{k=1}^{N}\mu_k p_k H_k\right) \;+\; O\left(\mu_{\max}^{(N+1)/N}\right) \tag{11.139}$$

where $\alpha_{\mathrm{dist}} \in (0,1)$. When A is doubly-stochastic, and the step-sizes are uniform across the agents so that $\mu_k \equiv \mu$, the above expressions reduce to

$$\mathrm{MSD}_{\mathrm{dist,av}} \;=\; \frac{\mu}{2hN}\mathrm{Tr}\left[\left(\sum_{k=1}^{N}H_k\right)^{-1}\left(\sum_{k=1}^{N}G_k\right)\right] \tag{11.140}$$

$$\alpha_{\mathrm{dist}} \;=\; 1 - \frac{2\mu}{N}\lambda_{\min}\left(\sum_{k=1}^{N}H_k\right) \;+\; o(\mu) \tag{11.141}$$

Comparing these expressions with (5.65) and (5.67) we observe that, to first-order in μ, the distributed solution is able to match the performance of the centralized solution for doubly-stochastic policies.

Observe further from (11.138) that, for sufficiently small step-sizes, the consensus and diffusion strategies are able to equalize the MSD performance across all agents. It is also instructive to compare expression (11.138) with (5.79) and (5.65) in the non-cooperative and centralized cases. Note that the effect of distributed cooperation results in the appearance of the scaling coefficients $\{p_k\}$; these factors are determined by the combination policy A. ■

Example 11.3 (MSD performance of MSE networks — Case I)**.** We revisit the setting of Example 6.3, where the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ satisfy the linear regression model (6.14) and where the cost associated with each agent is the mean-square-error cost, $J_k(w) = \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2$. As mentioned earlier, we already know from Example 6.1 that, in this case, the reference vectors w^o

and $w^\star$ coincide. We assume the agents employ uniform step-sizes and sense regression data with uniform covariance matrices, i.e., $\mu_k \equiv \mu$ and $R_{u,k} \equiv R_u$ for $k = 1, 2, \ldots, N$. We can assess the performance of the resulting consensus network (cf. Example 7.2) or diffusion network (cf. Example 7.3) as follows. In the current setting, and assuming complex data for generality, we know from (8.15) that

$$R_{s,k} \triangleq \lim_{i\to\infty} \mathbb{E}\left[\boldsymbol{s}_{k,i}(w^o)\boldsymbol{s}_{k,i}^*(w^o)\,|\boldsymbol{\mathcal{F}}_{i-1}\right] = \sigma_{v,k}^2 R_{u,k} \tag{11.142}$$

Therefore, using the definitions (11.12), we have:

$$H_k = \begin{bmatrix} R_u & 0 \\ 0 & R_u^{\mathsf{T}} \end{bmatrix} \equiv H, \qquad G_k = \sigma_{v,k}^2 \begin{bmatrix} R_u & \times \\ \times & R_u^{\mathsf{T}} \end{bmatrix} \tag{11.143}$$

where the off-diagonal block entries of G_k are not needed since H_k is block-diagonal. Substituting into (11.138), and using $h = 2$ for complex data, we conclude that the MSD performance of consensus or diffusion LMS networks is given by:

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{\mu M}{2}\left(\sum_{k=1}^{N} p_k^2 \sigma_{v,k}^2\right) \tag{11.144}$$

If the combination matrix A happens to be doubly stochastic, then $p = \mathbb{1}/N$. Substituting $p_k = 1/N$ into (11.144) gives

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{\mu M}{2}\,\frac{1}{N^2}\left(\sum_{k=1}^{N} \sigma_{v,k}^2\right) \tag{11.145}$$

which agrees with the expression that would result from (5.65) for the centralized LMS solution in the complex case, namely,

$$\mathrm{MSD}_{\mathrm{cent}} = \frac{\mu M}{2}\,\frac{1}{N}\left(\frac{1}{N}\sum_{k=1}^{N} \sigma_{v,k}^2\right) \tag{11.146}$$

Therefore, the distributed strategies are able to match the performance of the centralized solution for doubly stochastic combination policies. Observe though that, more generally, when A is not doubly-stochastic, the scaling factors $\{p_k^2\}$ appear in (11.144).

If the step-sizes were different across the agents, then we would instead obtain from (11.138) the following expression for the network performance:

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{M}{2}\left(\sum_{k=1}^{N} \mu_k p_k\right)^{-1}\left(\sum_{k=1}^{N} \mu_k^2 p_k^2 \sigma_{v,k}^2\right) \tag{11.147}$$

Another situation of interest is when the combination weights $\{a_{\ell k}\}$ are selected according to the averaging (or uniform) rule we encountered earlier in (8.89), namely,

$$a_{\ell k} = \begin{cases} 1/n_k, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{11.148}$$

where

$$n_k \triangleq |\mathcal{N}_k| \tag{11.149}$$

denotes the size of the neighborhood of agent k (or its degree). In this case, the matrix A will be left-stochastic and the entries of the corresponding Perron eigenvector are given by:

$$p_k = n_k \left(\sum_{m=1}^{N} n_m \right)^{-1} \tag{11.150}$$

Then, expression (11.144) gives

$$\text{MSD}_{\text{dist},k} = \text{MSD}_{\text{dist,av}} = \frac{\mu M}{2} \left(\sum_{k=1}^{N} n_k \right)^{-2} \left(\sum_{k=1}^{N} n_k^2 \sigma_{v,k}^2 \right) \tag{11.151}$$

which would reduce to (11.145) when the degrees of all agents are uniform, i.e., $n_k \equiv n$.

■

Example 11.4 (MSD performance of MSE networks — Case II)**.** We continue with the scenario of Example 11.3 for MSE networks except that we now assume that the regression covariance matrices are not necessarily uniform but chosen of the form $R_{u,k} = \sigma_{u,k}^2 I_M$. In this case, the expressions for $\{H_k, G_k\}$ in (11.143) become

$$H_k = \sigma_{u,k}^2 \begin{bmatrix} I_M & 0 \\ 0 & I_M \end{bmatrix}, \qquad G_k = \sigma_{v,k}^2 \, \sigma_{u,k}^2 \begin{bmatrix} I_M & \times \\ \times & I_M \end{bmatrix} \tag{11.152}$$

We can assess the performance of the resulting consensus network (cf. Example 7.2) or diffusion network (cf. Example 7.3) by substituting these values into (11.138), and using $h = 2$ for complex data, to get:

$$\text{MSD}_{\text{dist},k} = \text{MSD}_{\text{dist,av}} = \frac{M}{2} \left(\sum_{k=1}^{N} \mu_k^2 p_k^2 \sigma_{v,k}^2 \sigma_{u,k}^2 \right) \left(\sum_{k=1}^{N} \mu_k p_k \sigma_{u,k}^2 \right)^{-1} \tag{11.153}$$

If the combination matrix A happens to be doubly stochastic, then $p = \mathbb{1}/N$. Substituting $p_k = 1/N$ into (11.153) gives

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} \;=\; \frac{M}{2N}\left(\sum_{k=1}^{N}\mu_k^2\sigma_{v,k}^2\sigma_{u,k}^2\right)\left(\sum_{k=1}^{N}\mu_k\sigma_{u,k}^2\right)^{-1} \tag{11.154}$$

On the other hand, if the combination weights $\{a_{\ell k}\}$ are selected according to the averaging rule (11.148), we would then substitute (11.150) into (11.153) to give

$$\begin{aligned}\mathrm{MSD}_{\mathrm{dist},k} &= \mathrm{MSD}_{\mathrm{dist,av}}\\ &= \frac{M}{2}\left(\sum_{k=1}^{N}n_k\right)^{-1}\left(\sum_{k=1}^{N}\mu_k^2n_k^2\sigma_{v,k}^2\sigma_{u,k}^2\right)\left(\sum_{k=1}^{N}\mu_k n_k\sigma_{u,k}^2\right)^{-1}\end{aligned} \tag{11.155}$$

If the step-sizes are uniform across all agents, the above expression becomes

$$\begin{aligned}\mathrm{MSD}_{\mathrm{dist},k} &= \mathrm{MSD}_{\mathrm{dist,av}}\\ &= \frac{\mu M}{2}\left(\sum_{k=1}^{N}n_k\right)^{-1}\left(\sum_{k=1}^{N}n_k^2\sigma_{v,k}^2\sigma_{u,k}^2\right)\left(\sum_{k=1}^{N}n_k\sigma_{u,k}^2\right)^{-1}\end{aligned} \tag{11.156}$$

We illustrate these results numerically for the case of the averaging rule (11.148) with uniform step-sizes across the agents. Figure 11.1 shows the connected network topology with $N = 20$ agents used for this simulation, with the measurement noise variances, $\{\sigma_{v,k}^2\}$, and the power of the regression data, assumed of the form $R_{u,k} = \sigma_{u,k}^2 I_M$, shown in the plots of Figure 11.2, respectively. All agents are assumed to have a non-trivial self-loop so that the neighborhood of each agent includes the agent itself as well. The resulting network is therefore strongly-connected.

Figures 11.3 and 11.4 plot the evolution of the ensemble-average learning curves, $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, for consensus, ATC diffusion, and CTA diffusion for two choices of the step-size parameter: a smaller value at $\mu = 0.002$ and a second larger value at $\mu = 0.01$. The curves are obtained by averaging the trajectories $\{\frac{1}{N}\|\widetilde{\boldsymbol{w}}_i\|^2\}$ over 100 repeated experiments. The labels on the vertical axes in the figures refer to the learning curve $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ by writing $\mathrm{MSD}_{\mathrm{dist,av}}(i)$, with an iteration index i. Each experiment involves running the consensus (7.14) or diffusion (7.22)–(7.23) LMS recursions with $h = 2$ on complex-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ generated according to the model $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i)$, with $M = 10$. The unknown vector w^o is generated randomly and its norm is normalized to one.

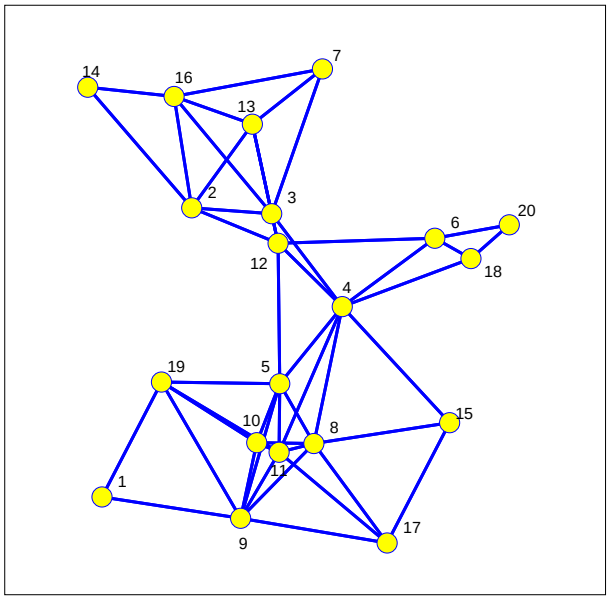

Figure 11.1: A connected network topology consisting of $N = 20$ agents employing the averaging rule (11.148).

Table 11.1: MSD values predicted by expressions (11.178) and (11.156) at the larger step-size value, $\mu = 0.01$.

algorithm	**result (11.178)**	**result (11.156)**
consensus strategy (7.14)	-42.00 dB	-44.34 dB
CTA diffusion strategy (7.22)	-42.00 dB	-44.34 dB
ATC diffusion strategy (7.23)	-43.42 dB	-44.34 dB

It is observed in Figure 11.3 that the learning curves tend to the same MSD value predicted by the theoretical expression (11.156), which provides a good approximation for the performance of distributed strategies for small step-sizes. However, it is observed in Figure 11.4 that once the step-size value is increased, differences in MSD performance arise among the algorithms, with ATC diffusion exhibiting the lowest (i.e., best) MSD value. The horizontal lines in this second figure represent the MSD levels that are predicted by future expression (11.178). This latter expression reflects the effect of higher-order terms in $\mu_{\max}$ and generally leads to an enhanced representation for the error variance of the distributed strategies, while expression (11.156), which

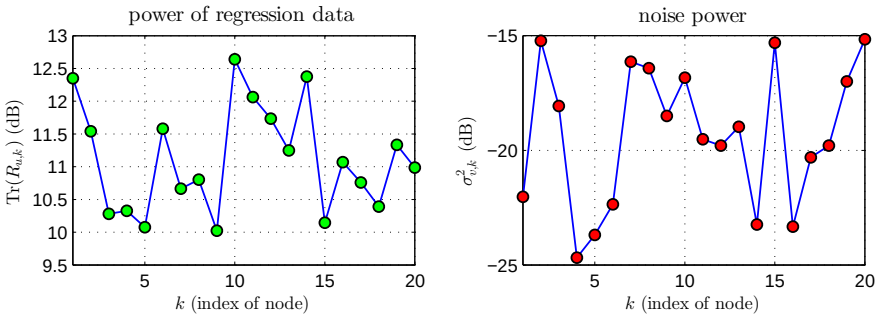

Figure 11.2: Regression data power (left) and measurement noise profile (right) across all agents in the network. The covariance matrices are assumed to be of the form $R_{u,k} = \sigma_{u,k}^2 I_M$, and the noise and regression data are Gaussian distributed in this simulation.

is the basis for the results in this example, is an expression for the MSD that is accurate to first-order in $\mu_{\max}$. Table 11.1 lists the MSD values that are predicted by expressions (11.178) and (11.156) at the larger step-size value, $\mu = 0.01$.

■

Example 11.5 (Is cooperation always beneficial?)**.** We continue with the discussion from Example 11.3 over MSE networks. If each agent in the network were to estimate w^o on its own in a non-cooperative manner by running its individual LMS learning rule (3.125), then we know from (4.186) that each agent will attain the MSD level shown below:

$$\text{MSD}_{\text{ncop},k} \;=\; \frac{\mu M}{2}\,\sigma_{v,k}^2 \tag{11.157}$$

along with the average performance across all N agents given by:

$$\text{MSD}_{\text{ncop,av}} \;=\; \frac{\mu M}{2}\left(\frac{1}{N}\sum_{k=1}^{N}\sigma_{v,k}^2\right) \tag{11.158}$$

Now assume A is doubly stochastic. Comparing (11.145) with (11.158), it is obvious that

$$\text{MSD}_{\text{dist,av}} \;=\; \frac{1}{N}\,\text{MSD}_{\text{ncop,av}} \tag{11.159}$$

which shows that, for MSE networks, the consensus and diffusion strategies outperform the average performance of the non-cooperative strategy by a factor of N. But how do the performance metrics of an agent compare to

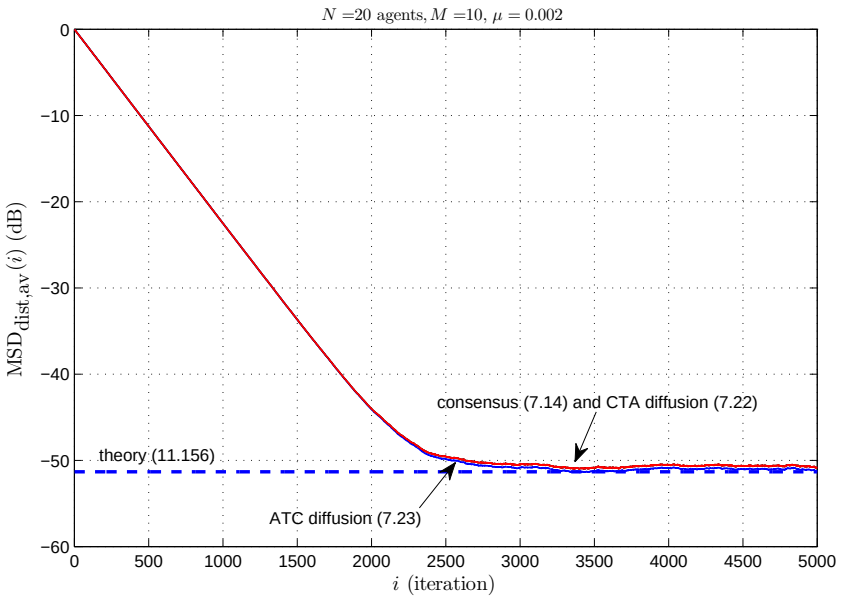

Figure 11.3: Evolution of the learning curves for three strategies, namely, consensus (7.14), CTA diffusion (7.22), ATC diffusion (7.23), for the smaller step-size at $\mu = 0.002$.

each other in the distributed and non-cooperative modes of operation? From (11.145) and (11.157) we observe that if the noise variance is uniform across all agents, i.e., $\sigma_{v,k}^2 \equiv \sigma_v^2$, then the MSD of each individual agent in the distributed solution will be smaller by the same factor N than their non-cooperative performance. However, when the noise profile varies across the agents, then the performance metrics of an individual agent in the distributed and non-cooperative solutions cannot be compared directly: one can be larger than the other depending on the noise profile. For example, for $N = 2$, $\sigma_{v,1}^2 = 1$, and $\sigma_{v,2}^2 = 9$, agent 1 will not benefit from cooperation while agent 2 will. ■

Example 11.6 (MSD performance of MSE networks — Case III)**.** We reconsider the setting of Examples 8.8 and 8.11, which deals with a *variation* of MSE networks where the data model at each agent is instead assumed to be given by

$$\boldsymbol{d}_k(i) \;=\; \boldsymbol{u}_{k,i} w_k^o \;+\; \boldsymbol{v}_k(i) \tag{11.160}$$

with the model vectors, w_k^o, being possibly different at the various agents. We explained in Example 8.11 that the gradient noise process at agent k is given

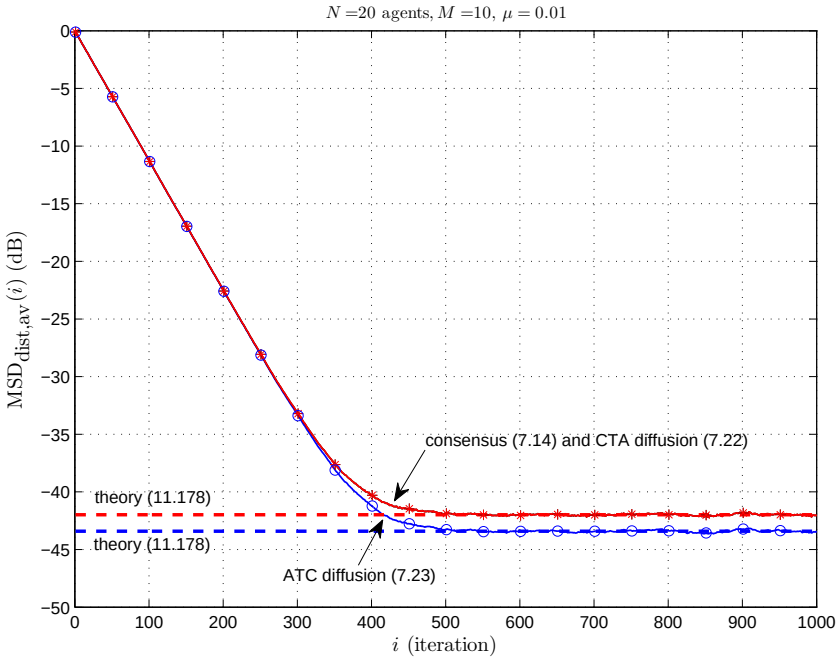

Figure 11.4: Evolution of the learning curves for three strategies, namely, consensus (7.14), CTA diffusion (7.22), ATC diffusion (7.23), for the larger step-size at $\mu = 0.01$.

by expression (8.127), namely,

$$\boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1}) \;=\; \frac{2}{h}\left(R_{u,k} - \boldsymbol{u}_{k,i}^{*}\boldsymbol{u}_{k,i}\right)\left(w_k^o - \boldsymbol{\phi}_{k,i-1}\right) \;-\; \frac{2}{h}\boldsymbol{u}_{k,i}^{*}\boldsymbol{v}_k(i) \quad (11.161)$$

By repeating the arguments of Example 8.8 for the general distributed strategy (8.5), we can similarly show that the limit point, $w^{\star}$, of the network is given by a relation similar to (8.86), namely,

$$w^{\star} \;=\; \left(\sum_{k=1}^{N} q_k R_{u,k}\right)^{-1}\left(\sum_{k=1}^{N} q_k R_{u,k} w_k^o\right) \quad (11.162)$$

where the positive scalars $\{q_k\}$ are the entries of the vector q defined by (8.50). Using (11.161) we can evaluate the second-order moment $R_{s,k}$ defined by (11.8) as follows. We introduce the difference

$$z_k \;\stackrel{\Delta}{=}\; w_k^o - w^{\star}, \quad k = 1, 2, \ldots, N \quad (11.163)$$

It is clear that $z_k = 0$ when all w_k^o coincide at the same location w^o, in which case we get $w^\star = w^o$. In general though, the perturbation vectors, $\{z_k\}$ need not be zero. From (11.161), and using the conditions imposed on the regression data and noise processes across the agents from Example 6.3, we find that

$$R_{s,k} \;=\; \frac{4}{h^2}\mathbb{E}\left(R_{u,k} - \boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}\right) z_k z_k^* \left(R_{u,k} - \boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}\right) \;+\; \frac{4}{h^2}\sigma_{v,k}^2 R_{u,k} \tag{11.164}$$

The first term on the right-hand side involves a fourth-order moment in the regression data. To evaluate this term in closed-form, we assume that the regression data is circular and Gaussian-distributed. In that case, it is known that for any $M \times M$ Hermitian matrix W_k it holds that [206, p.11]:

$$\mathbb{E}\left(\boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^* W_k \boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^*\right) \;=\; R_{u,k}\mathrm{Tr}(W_k R_{u,k}) \;+\; \frac{2}{h}R_{u,k}W_k R_{u,k} \tag{11.165}$$

This expression shows how the (weighted) fourth-order moment of the process $\boldsymbol{u}_{k,i}$ is determined by its second-order moment, $R_{u,k}$. Let

$$W_k = z_k z_k^* \tag{11.166}$$

which is a rank-one nonnegative definite Hermitian matrix. Expanding the first term on the right-hand side of (11.164) and using (11.165), we conclude that

$$R_{s,k} = \frac{4}{h^2}\sigma_{v,k}^2 R_{u,k} \;+\; \frac{4}{h^2}R_{u,k}\mathrm{Tr}(W_k R_{u,k}) \;+\; \frac{4}{h^2}\left(\frac{2}{h} - 1\right) R_{u,k}W_k R_{u,k} \tag{11.167}$$

In particular, for complex data, the above result evaluates to the following using $h = 2$:

$$R_{s,k} \;=\; \sigma_{v,k}^2 R_{u,k} \;+\; R_{u,k}\|z_k\|_{R_{u,k}}^2 \quad \text{(complex data)} \tag{11.168}$$

Each agent k in the network is associated with an individual cost of the form $J_k(w) = \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2$. We now assume that the regression covariance matrices are of the form $R_{u,k} = \sigma_{u,k}^2 I_M$. In this case, expression (11.168) for $R_{s,k}$ simplifies to

$$\begin{aligned} R_{s,k} \;&=\; \left(\sigma_{v,k}^2 + \sigma_{u,k}^2\|z_k\|^2\right)\sigma_{u,k}^2 I_M \\ &\triangleq\; \bar{\sigma}_{v,k}^2\sigma_{u,k}^2 I_M \quad \text{(complex data)} \end{aligned} \tag{11.169}$$

where we introduced the modified noise variance

$$\bar{\sigma}_{v,k}^2 \;\triangleq\; \sigma_{v,k}^2 + \sigma_{u,k}^2\|z_k\|^2 \tag{11.170}$$

Consequently, the expressions for $\{H_k, G_k\}$ become (compare with (11.152)):

$$H_k = \sigma_{u,k}^2 \begin{bmatrix} I_M & 0 \\ 0 & I_M \end{bmatrix}, \qquad G_k = \bar{\sigma}_{v,k}^2 \sigma_{u,k}^2 \begin{bmatrix} I_M & \times \\ \times & I_M \end{bmatrix} \tag{11.171}$$

We can assess the performance of the resulting consensus network (cf. Example 7.2) or diffusion network (cf. Example 7.3) by substituting these values into (11.138), and using $h = 2$ for complex data, to get:

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{M}{2} \left(\sum_{k=1}^{N} \mu_k^2 p_k^2 \bar{\sigma}_{v,k}^2 \sigma_{u,k}^2 \right) \left(\sum_{k=1}^{N} \mu_k p_k \sigma_{u,k}^2 \right)^{-1} \tag{11.172}$$

If the combination matrix A happens to be doubly stochastic, then $p = \mathbb{1}/N$. Substituting $p_k = 1/N$ into (11.172) gives

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{M}{2N} \left(\sum_{k=1}^{N} \mu_k^2 \bar{\sigma}_{v,k}^2 \sigma_{u,k}^2 \right) \left(\sum_{k=1}^{N} \mu_k \sigma_{u,k}^2 \right)^{-1} \tag{11.173}$$

On the other hand, if the combination weights $\{a_{\ell k}\}$ are selected according to the averaging rule (11.148), we would then substitute (11.150) into (11.153) to give

$$\begin{aligned} \mathrm{MSD}_{\mathrm{dist},k} &= \mathrm{MSD}_{\mathrm{dist,av}} \\ &= \frac{M}{2} \left(\sum_{k=1}^{N} n_k \right)^{-1} \left(\sum_{k=1}^{N} \mu_k^2 n_k^2 \bar{\sigma}_{v,k}^2 \sigma_{u,k}^2 \right) \left(\sum_{k=1}^{N} \mu_k n_k \sigma_{u,k}^2 \right)^{-1} \end{aligned} \tag{11.174}$$

If the step-sizes are uniform across all agents, the above expression becomes

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{\mu M}{2} \left(\sum_{k=1}^{N} n_k \right)^{-1} \left(\sum_{k=1}^{N} n_k^2 \bar{\sigma}_{v,k}^2 \sigma_{u,k}^2 \right) \left(\sum_{k=1}^{N} n_k \sigma_{u,k}^2 \right)^{-1} \tag{11.175}$$

We illustrated this result numerically earlier in Figure 8.5 while discussing the convergence of the network towards its Pareto limit point.

■

Example 11.7 (Higher-order MSD terms)**.** We explained earlier in Sec. 4.5, while motivating the definition of the MSD metric, that expressions of the form (11.37) help assess the size of the error variance, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, in steady-state and for sufficiently small step-sizes (i.e., in the slow adaptation regime).

The computation leads to an expression for the MSD that is *first-order* in $\mu_{\max}$, as can be ascertained from (11.118).

If we revisit the derivation of (11.118) in the proof of Lemma 11.3, we will observe that this expression was obtained by eliminating the contribution of the higher-order term, $O(\mu_{\max}^2)$, which appears in the expansion (11.120). We can motivate an alternative expression for assessing the size of the error variance, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, by retaining the higher-order term that is available (i.e., known) rather than neglecting it. It is expected that, by doing so, the resulting performance expression will generally provide a more accurate representation for the error variance, especially at larger step-sizes; we illustrated this behavior already in the simulations of Example 11.4 — recall Figure 11.4. The alternative performance expression can be motivated as follows.

Similarly to (4.83)–(4.84), the argument that led to (11.45) would establish the following two expressions for the limit superior and limit inferior of the error variance at each agent k (see, e.g., (11.107) and (11.109)):

$$\limsup_{i\to\infty} \frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}^{e}\|^2 = \frac{1}{h}\mathrm{Tr}(\mathcal{J}_k\mathcal{X}) + O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.176}$$

$$\liminf_{i\to\infty} \frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}^{e}\|^2 = \frac{1}{h}\mathrm{Tr}(\mathcal{J}_k\mathcal{X}) - O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.177}$$

with the same common positive constant $\mathrm{Tr}(\mathcal{J}_k\mathcal{X})$; this constant is equal to the quantity that appears on the left-hand side of (11.120). Relations (11.176)–(11.177) indicate that we can also employ the quantity $\frac{1}{h}\mathrm{Tr}(\mathcal{J}_k\mathcal{X})$ to assess the size of the error variance, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, in steady-state for small step-sizes. Subsequently, by averaging over all agents, we can similarly use the quantity $\frac{1}{hN}\mathrm{Tr}(\mathcal{X})$ to assess the size of the network error variance, $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i}\|^2$, also in steady-state and for small step-sizes. If we recall (11.58), then this argument suggests the following alternative expressions for evaluating the network error variance:

$$\mathrm{MSD}_{\mathrm{dist,av}} = \frac{1}{hN}\sum_{n=0}^{\infty}\mathrm{Tr}\left[\mathcal{B}^n\mathcal{Y}\,(\mathcal{B}^*)^n\right] \tag{11.178}$$

$$= \frac{1}{hN}\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}}(I-\mathcal{F})^{-1}\mathrm{bvec}\left(I_{hMN}\right) \tag{11.179}$$

where we continue to use the notation MSD to represent this value. As we already know from the proof of Lemma 11.3, if we expand the right-hand side of (11.179) in terms of powers of $\mu_{\max}$, then the first term in this expansion (i.e., the one that is linear in $\mu_{\max}$) will be given by expression (11.118).

■

11.4 Excess-Risk Performance

We can similarly determine closed-form expressions for the excess-risk performance of the individual agents and for the network.

Theorem 11.4 (Network ER performance)**.** Consider a network of N interacting agents running the distributed strategy (8.46) with a primitive matrix $P = A_1 A_o A_2$. Assume the aggregate cost (9.10) and the individual costs, $J_k(w)$, satisfy the conditions in Assumptions 6.1 and 10.1. Assume further that the first and fourth-order moments of the gradient noise process satisfy the conditions of Assumption 8.1 with the second-order moment condition (8.115) replaced by the fourth-order moment condition (8.121). Assume also (11.11). Then, it holds that

$$\limsup_{i\to\infty} \frac{1}{2}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i-1}^{e}\|_{\bar{H}}^2 = \frac{1}{2}\mathrm{Tr}(\mathcal{Q}_k\mathcal{X}) + O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.180}$$

$$\limsup_{i\to\infty} \frac{1}{2N}\left(\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}^{e}\|_{(I_N\otimes\bar{H})}^2\right) = \frac{1}{2N}\mathrm{Tr}(\bar{\mathcal{H}}\mathcal{X}) + O\left(\mu_{\max}^{1+\gamma_m}\right) \tag{11.181}$$

for the same quantities defined earlier in Theorem 11.2 and where

$$\bar{\mathcal{H}} = I_N \otimes \bar{H} = \mathrm{diag}\{\bar{H}, \bar{H}, \ldots, \bar{H}\} \tag{11.182}$$

$$\mathcal{Q}_k = \mathrm{diag}\{\, 0_{hM}, \ldots, 0_{hM}, \bar{H}, 0_{hM}, \ldots, 0_{hM}\,\} \tag{11.183}$$

with the matrix $\bar{H}$ defined by (11.36) appearing in the $k-$block location of $\mathcal{Q}_k$. Moreover, it further holds that

$$\mathrm{Tr}(\mathcal{Q}_k\mathcal{X}) = \left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}}(I-\mathcal{F})^{-1}\mathrm{bvec}\left(\mathcal{Q}_k\right) \tag{11.184}$$

$$\mathrm{Tr}(\bar{\mathcal{H}}\mathcal{X}) = \left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}}(I-\mathcal{F})^{-1}\mathrm{bvec}\left(\bar{\mathcal{H}}\right) \tag{11.185}$$

and, for large enough i, the convergence rate of the excess-risk measure towards its steady-state region (11.180) is given by the same expression (11.47). Furthermore, the ER performance for the individual agents and for the network are given by:

$$\mathrm{ER}_{\mathrm{dist},k} = \mathrm{ER}_{\mathrm{dist,av}} = \frac{h}{4}\left(\sum_{k=1}^{N} q_k\right)^{-1}\mathrm{Tr}\left(\sum_{k=1}^{N} q_k^2 R_{s,k}\right) \tag{11.186}$$

Proof. We start from relation (11.84) but select Σ now as the solution to the following Lyapunov equation:

$$\Sigma - \mathcal{B}^*\Sigma\mathcal{B} = \bar{\mathcal{H}} \tag{11.187}$$

and repeat the argument that led to (11.106)–(11.107) to conclude that expressions (11.180)–(11.181) hold.

With regards to expression (11.186), we first note from (11.35) and (11.180) that we need to evaluate the limit:

$$\mathrm{ER}_{\mathrm{dist},k} = \mu_{\max} \cdot \left(\lim_{\mu_{\max}\to 0} \limsup_{i\to\infty} \frac{1}{\mu_{\max}} \left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}} (I-\mathcal{F})^{-1}\mathrm{bvec}(\mathcal{Q}_k) \right) \tag{11.188}$$

We focus on the right-most factor inside the above expression. Using the low-rank factorization (9.244), we have

$$\begin{aligned} \left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}} (I-\mathcal{F})^{-1}\mathrm{bvec}(\mathcal{Q}_k) \;&=\; O(\mu_{\max}^2) + \\ \left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}} (p\otimes I_{2M})\otimes_b(p\otimes I_{2M})\, Z^{-1} \left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right)\otimes_b\left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right) \mathrm{bvec}\left(\mathcal{Q}_k\right) \end{aligned} \tag{11.189}$$

Using the block Kronecker product property (11.86), it can be verified that

$$\left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right)\otimes_b\left(\mathbb{1}^{\mathsf{T}}\otimes I_{2M}\right)\mathrm{bvec}(\mathcal{Q}_k) \;=\; \mathrm{vec}(\bar{H}) \tag{11.190}$$

Let $x = Z^{-1}\mathrm{vec}(\bar{H})$. Then, the same argument that led to (11.128) will show that the $2M\times 2M$ matrix $X = \mathrm{unvec}(x)$ is the unique solution to the Lyapunov equation

$$\left(\sum_{k=1}^{N} q_k\right)\bar{H}X \;+\; X\bar{H}\left(\sum_{k=1}^{N} q_k\right) \;=\; \bar{H} \tag{11.191}$$

so that

$$X \;=\; \frac{1}{2}\left(\sum_{k=1}^{N} q_k\right)^{-1} I_{2M} \tag{11.192}$$

Repeating the derivation that led to (11.132) we arrive at

$$\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}} (I-\mathcal{F})^{-1}\mathrm{bvec}(\mathcal{H}) = \frac{1}{2}\left(\sum_{k=1}^{N} q_k\right)^{-1} \mathrm{Tr}\left(\sum_{k=1}^{N} q_k^2 G_k\right) + O(\mu_{\max}^2) \tag{11.193}$$

Substituting into the right-hand side of (11.188) and evaluating the limit we arrive at (11.186) after recalling from (11.12) that

$$\mathrm{Tr}(G_k) \;=\; \begin{cases} R_{s,k} & \text{(real data)} \\ 2R_{s,k} & \text{(complex data)} \end{cases} \tag{11.194}$$

□

Example 11.8 (ER performance of consensus and diffusion networks)**.** We specialize the result of Theorem 11.4 to the same consensus and diffusion strategies from Example 11.2. In this case we get

$$\mathrm{ER}_{\mathrm{dist},k} = \mathrm{ER}_{\mathrm{dist,av}} = \frac{h}{4}\mathrm{Tr}\left[\left(\sum_{k=1}^{N}\mu_k p_k\right)^{-1}\left(\sum_{k=1}^{N}\mu_k^2 p_k^2 R_{s,k}\right)\right] \tag{11.195}$$

where $h = 1$ for real data and $h = 2$ for complex data. When the step-sizes are uniform across all agents, $\mu_k \equiv \mu$, and using the fact that the entries p_k add up to one, the above expression simplifies to

$$\mathrm{ER}_{\mathrm{dist},k} \;=\; \mathrm{ER}_{\mathrm{dist,av}} \;=\; \frac{\mu h}{4}\left(\sum_{k=1}^{N}p_k^2 R_{s,k}\right) \tag{11.196}$$

■

Example 11.9 (Performance of diffusion learner)**.** We generalize the scenario of Example 7.4 and consider a collection of N learners cooperating to minimize some arbitrary strongly-convex function $J(w)$ over a strongly-connected network, namely,

$$w^o \;\triangleq\; \arg\min_{w}\; J(w) \tag{11.197}$$

where $J(w)$ is the average of some loss measure, say, $J(w) = \mathbb{E}\,Q(w;\boldsymbol{x}_{k,i})$. As before, each learner k receives a streaming sequence of real-valued data vectors $\{\boldsymbol{x}_{k,i}, i = 1,2,\ldots\}$ that arise from some fixed distribution $\mathcal{X}$. We assume the agents run a consensus or diffusion strategy, say, the ATC diffusion strategy (7.19):

$$\left\{\begin{array}{rcl}\boldsymbol{\psi}_{k,i} &=& \boldsymbol{w}_{k,i-1} - \mu_k\nabla_{w^{\mathsf{T}}}Q(\boldsymbol{w}_{k,i-1};\boldsymbol{x}_{k,i})\\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i}\end{array}\right. \tag{11.198}$$

The gradient noise vector corresponding to each individual agent k is given by

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) \;=\; \nabla_{w^{\mathsf{T}}}Q(\boldsymbol{w}_{k,i-1};\boldsymbol{x}_{k,i}) \;-\; \nabla_{w^{\mathsf{T}}}\mathbb{E}\,Q(\boldsymbol{w}_{k,i-1};\boldsymbol{x}_{k,i}) \tag{11.199}$$

so that

$$\boldsymbol{s}_{k,i}(w^o) \;=\; \nabla_{w^{\mathsf{T}}}Q(w^o;\boldsymbol{x}_{k,i}) \tag{11.200}$$

Since we are assuming the distribution of the random process $\boldsymbol{x}_{k,i}$ is stationary and fixed across all agents, it follows that

$$R_{s,k} \;=\; \mathbb{E}\,\nabla_{w^{\mathsf{T}}}Q(w^o;\boldsymbol{x}_{k,i})\,[\nabla_{w^{\mathsf{T}}}Q(w^o;\boldsymbol{x}_{k,i})]^{\mathsf{T}} \;\equiv\; R_s, \qquad k = 1,2,\ldots,N \tag{11.201}$$

Substituting into (11.186), and using $h = 1$ for real data, we conclude that the excess-risk of the diffusion solution (and of consensus as well) is given by

$$\mathrm{ER}_{\mathrm{dist,av}} \;=\; \frac{1}{4}\left(\sum_{k=1}^{N}\mu_k p_k\right)^{-1}\left(\sum_{k=1}^{N}\mu_k^2 p_k^2\right)\mathrm{Tr}\,(R_s) \tag{11.202}$$

If we assume uniform step-sizes, $\mu_k \equiv \mu$ for $k = 1, 2, \ldots, N$, and use the fact that the $\{p_k\}$ add up to one, then expression (11.202) reduces to

$$\mathrm{ER}_{\mathrm{dist,av}} \;=\; \frac{\mu}{4}\left(\sum_{k=1}^{N}p_k^2\right)\mathrm{Tr}\,(R_s) \tag{11.203}$$

For comparison purposes, we reproduce below ER expression (5.98) for the centralized solution from Example 5.3:

$$\mathrm{ER}_{\mathrm{cent}} \;=\; \frac{\mu}{4}\left(\frac{1}{N}\right)\mathrm{Tr}(R_s) \tag{11.204}$$

For doubly-stochastic combination matrices A, it holds that $p_k = 1/N$ so that (11.203) reduces to (11.204).

We illustrate these results numerically for the logistic risk function (7.24) from Example 7.4, namely,

$$J(w) \;\triangleq\; \frac{\rho}{2}\|w\|^2 \;+\; \mathbb{E}\left\{\ln\left(1+e^{-\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}}w}\right)\right\} \tag{11.205}$$

Figure 11.5 shows the connected network topology with $N = 20$ agents used for this simulation. All agents are assumed to employ the same step-size parameter, i.e., $\mu_k \equiv \mu$, and they have non-trivial self-loops so that the neighborhood of each agent includes the agent itself. The resulting network is therefore strongly-connected.

The corresponding consensus, CTA diffusion, and ATC diffusion strategies with uniform step-sizes across the agents take the following forms:

$$\left\{\begin{array}{rcl}\boldsymbol{\psi}_{k,i-1} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i} \quad (\underline{\text{consensus}})\\ \boldsymbol{w}_{k,i} &=& (1-\rho\mu)\boldsymbol{\psi}_{k,i-1} + \mu\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}\left(\dfrac{1}{1+e^{\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}}\boldsymbol{w}_{k,i-1}}}\right)\end{array}\right. \tag{11.206}$$

and

$$\left\{\begin{array}{rcl}\boldsymbol{\psi}_{k,i-1} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i} \quad (\underline{\text{CTA diffusion}})\\ \boldsymbol{w}_{k,i} &=& (1-\rho\mu)\boldsymbol{\psi}_{k,i-1} + \mu\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}\left(\dfrac{1}{1+e^{\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}}\boldsymbol{\psi}_{k,i-1}}}\right)\end{array}\right. \tag{11.207}$$

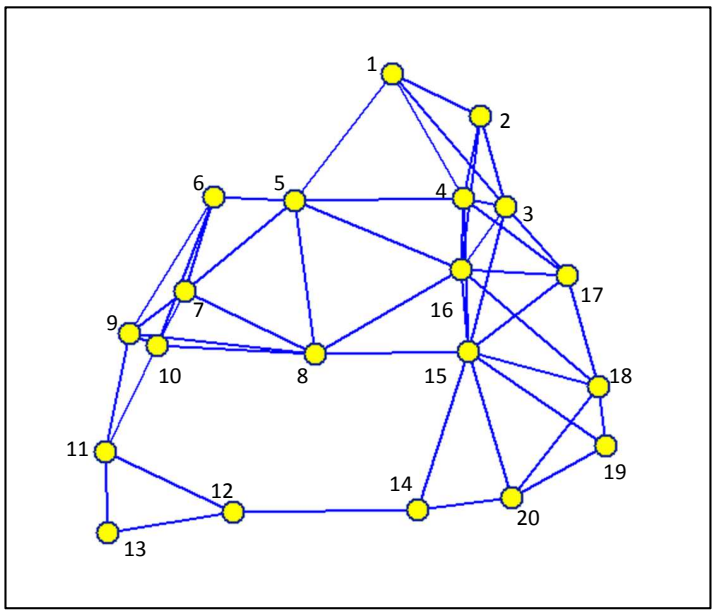

Figure 11.5: A connected network topology consisting of $N = 20$ agents employing the Metropolis rule (8.100). Each agent k is assumed to belong its neighborhood $\mathcal{N}_k$.

and

$$\begin{cases} \boldsymbol{\psi}_{k,i} &= (1-\rho\mu)\boldsymbol{w}_{k,i-1} + \mu\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}\left(\dfrac{1}{1+e^{\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}}\boldsymbol{w}_{k,i-1}}}\right) \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \quad \text{(ATC diffusion)} \end{cases} \tag{11.208}$$

where the combination weights $\{a_{\ell k}\}$ arise from the Metropolis rule (8.100). This rule leads to a doubly-stochastic matrix, A, so that the entries of the Perron eigenvector are given by $p_k = 1/N$. In this way, the ER performance level (11.203) for the above distributed strategies reduces to

$$\mathrm{ER}_{\mathrm{dist,av}} \;=\; \frac{\mu}{4}\left(\frac{1}{N}\right)\mathrm{Tr}(R_s) \tag{11.209}$$

Figures 11.6 and 11.7 plot the evolution of the ensemble-average learning curves, $\mathbb{E}\,\{J(\boldsymbol{w}_{i-1}) - J(w^o)\}$, for consensus, ATC diffusion, and CTA diffusion for two choices of the step-size parameter: a smaller value at $\mu = 1\times 10^{-4}$ and a second value that is three times larger at $\mu = 3\times 10^{-4}$. The curves

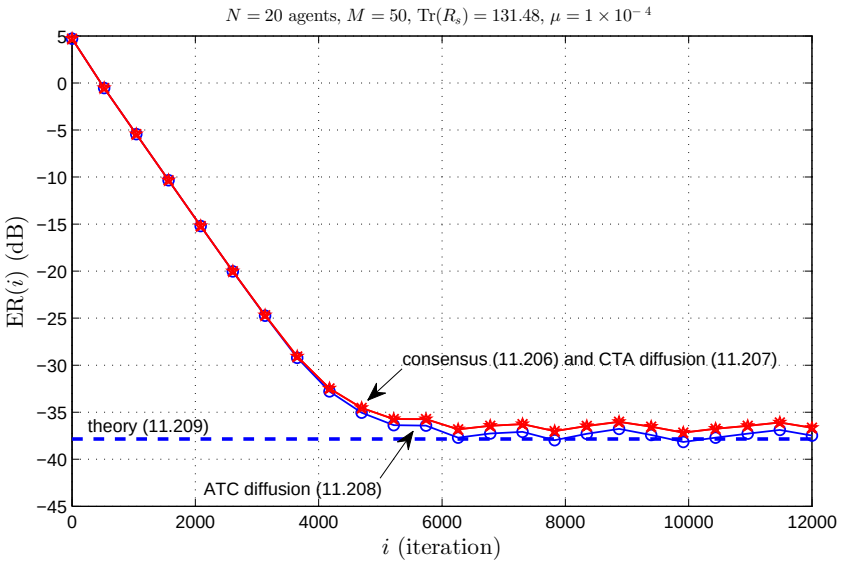

Figure 11.6: Evolution of the learning curves for three strategies, namely, consensus (11.206), CTA diffusion (11.207), and ATC diffusion (11.208), with all agents employing the smaller step-size $\mu = 1 \times 10^{-4}$.

are obtained by averaging the trajectories $\{J(\boldsymbol{w}_{i-1}) - J(w^o)\}$ over 100 repeated experiments. The labels on the vertical axes in the figures refer to the learning curves by writing $\mathrm{ER}_{\mathrm{dist,av}}(i)$, with an iteration index i. Each experiment involves running the consensus (11.206) or diffusion (11.207)–(11.208) logistic recursions with $\rho = 10$ and $h = 1$ for real data $\{\boldsymbol{\gamma}_k(i), \boldsymbol{h}_{k,i}\}$, where the dimension of the feature vectors $\{\boldsymbol{h}_{k,i}\}$ is $M = 50$. The data used for the simulation originate from the alpha data set [223]; we use the first 50 features for illustration purposes so that $M = 50$. To generate the trajectories for the experiments in this example, the optimal w^o and the gradient noise covariance matrix, R_s, are first estimated off-line by applying a batch algorithm to all data points. For the data used in this experiment we have $\mathrm{Tr}(R_s) \approx 131.48$.

It is observed in Figure 11.6 that the learning curves tend towards the ER value predicted by the theoretical expression (11.209), which provides a good approximation for the performance of distributed strategies for small step-sizes. However, it is observed in Figure 11.7 that once the step-size value is increased, differences in ER performance arise among the algorithms, with ATC diffusion exhibiting the lowest (i.e., best) ER value. The horizontal lines in the second figure represent the ER levels that are predicted by the future expression (11.210). This latter expression reflects the effect of higher-order

terms in $\mu_{\max}$ and generally leads to an enhanced representation for the mean excess cost, while expression (11.209), which is the basis for the results in this example, is an expression for the ER that is accurate to first-order in $\mu_{\max}$.

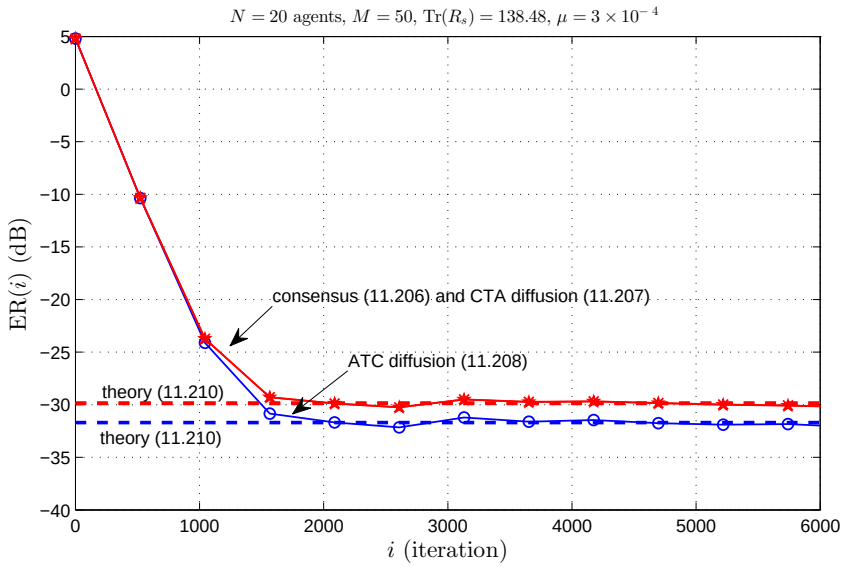

Figure 11.7: Evolution of the learning curves for three strategies, namely, consensus (11.206), CTA diffusion (11.207), and ATC diffusion (11.208), with all agents employing the larger step-size $\mu = 3 \times 10^{-4}$.

■

Example 11.10 (Higher-order ER terms)**.** We explained earlier following (11.39) that the ER metric (11.33) assesses the size of the mean fluctuation of the normalized aggregate cost, $\mathbb{E}\left\{\bar{J}^{\mathrm{glob},\star}(\boldsymbol{w}_{k,i-1}) - \bar{J}^{\mathrm{glob},\star}(w^\star)\right\}$, in steady-state and for sufficiently small step-sizes (i.e., in the slow adaptation regime). The computation leads to an expression for the ER that is *first-order* in $\mu_{\max}$, as can be ascertained from (11.186).

If we revisit the derivation of (11.186) in the proof of Theorem 11.3, we will observe that this expression was obtained by eliminating the contribution of the higher-order term, $O(\mu_{\max}^2)$, which appears in the expansion (11.189). We can motivate an alternative expression for assessing the size of the mean cost fluctuation by retaining the higher-order term that is available (i.e., known) rather than neglecting it. It is expected that, by doing so, the resulting performance expression will generally provide a more accurate representation for the mean cost fluctuation, especially at larger step-sizes; we illustrated this

behavior in Figure 11.7. In a manner similar to Example 11.7, we can motivate the following enhanced expression for the excess mean cost, which reflects contributions from higher-order powers of $\mu_{\max}$ as well:

$$\mathrm{ER}_{\mathrm{dist,av}} \;=\; \frac{1}{2N}\left(\mathrm{bvec}\left(\mathcal{Y}^{\mathsf{T}}\right)\right)^{\mathsf{T}}(I-\mathcal{F})^{-1}\mathrm{bvec}\left(\bar{\mathcal{H}}\right) \tag{11.210}$$

where we continue to use the notation ER to represent this value. As we already know from the proof of Theorem 11.3, if we expand the right-hand side of (11.210) in terms of powers of $\mu_{\max}$, then the first term in this expansion (i.e., the one that is linear in $\mu_{\max}$) will be given by expression (11.186). ■

11.5 Comparing Consensus and Diffusion Strategies

Using results from the previous sections, we can compare some performance properties of diffusion and consensus networks. Recall from (8.7)–(8.10) that the consensus and diffusion strategies correspond to the following choices for $\{A_o, A_1, A_2\}$ in terms of a single combination matrix A in the general description (8.46):

$$\text{consensus:} \qquad A_o = A, \;\; A_1 = I_N = A_2 \tag{11.211}$$

$$\text{CTA diffusion:} \qquad A_1 = A, \;\; A_2 = I_N = A_o \tag{11.212}$$

$$\text{ATC diffusion:} \qquad A_2 = A, \;\; A_1 = I_N = A_o \tag{11.213}$$

Example 11.11 (Diffusion outperforms consensus over MSE networks)**.** Expression (11.138) indicates that the MSD performance of the consensus and diffusion strategies are identical to first-order in the step-size parameters, as already anticipated by the results in Figures 11.3 and 11.4. We now examine the MSD performance level more closely by considering higher-order terms as well. More specifically, we resort to the alternative expression (11.178).

The following example is a generalization of a similar discussion from [248]. Let us consider a situation in which all agents in a strongly-connected network employ the same step-size, i.e., $\mu_k \equiv \mu$, and that the diffusion and consensus strategies from (8.46) are implemented with the same combination matrix, A. Without loss in generality, we consider the case of real-valued data. Let us assume further that the Hessian matrices of all individual costs, $J_k(w)$, evaluate to the same value at the reference point $w^\star$, namely,

$$\nabla_w^2 J_k(w^\star) \;\equiv\; H, \quad k = 1, 2, \ldots, N \tag{11.214}$$

for some constant matrix H. We also assume that the gradient noise variances $\{G_k\}$ approach the same value in steady-state apart from some scaling to account for the possibility of different noise power levels across the agents, i.e., we assume that the $\{G_k\}$ have the form:

$$G_k \;\equiv\; \sigma_{v,k}^2\, G, \quad k = 1, 2, \ldots, N \tag{11.215}$$

for some constant matrix G. For example, these two conditions on $\{\nabla_w^2 J_k(w^\star),\; G_k\}$ are readily satisfied by the class of MSE networks defined earlier in Example 6.3 when the regression covariance matrices are uniform across all agents, $R_{u,k} \equiv R_u$ for $k = 1, 2, \ldots, N$. Indeed, if we write down an expression similar to (8.15) for the gradient noise process at each agent k, namely,

$$\boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1}) \;=\; 2\left(R_u - \boldsymbol{u}_{k,i}^{\mathsf{T}}\boldsymbol{u}_{k,i}\right)\widetilde{\boldsymbol{\phi}}_{k,i-1} \;-\; 2\boldsymbol{u}_{k,i}^{\mathsf{T}}\boldsymbol{v}_k(i) \tag{11.216}$$

then we conclude that

$$R_{s,k} \;\triangleq\; \lim_{i\to\infty} \mathbb{E}\left[\,\boldsymbol{s}_{k,i}(w^\star)\boldsymbol{s}_{k,i}^{\mathsf{T}}(w^\star)\,|\boldsymbol{\mathcal{F}}_{i-1}\right] \;=\; 4\sigma_{v,k}^2 R_u \tag{11.217}$$

so that, using the definitions (11.12), we obtain for the case of real-data:

$$\nabla_w^2 J_k(w^\star) \;=\; 2R_u \;\equiv\; H, \quad G_k \;=\; 4\sigma_{v,k}^2 R_u \;\equiv\; \sigma_{v,k}^2 G \tag{11.218}$$

with $G = 2H$ in this case.

We are interested in comparing the MSD performance of diffusion and consensus networks under conditions (11.214)–(11.215). If desired, we can also compare against the performance of the non-cooperative solution. For this latter comparison to be meaningful, we would need to assume that all individual costs, $J_k(w)$, have the same minimizer so that the distributed and the non-cooperative implementations would be seeking the same minimizer. If we were only interested in comparing the consensus and diffusion strategies, then there is no need to assume that the individual costs have the same minimizer; the argument given below would still apply.

We collect the noise power scalings into an $N \times N$ diagonal matrix

$$R_v = \mathrm{diag}\{\sigma_{v,1}^2,\; \sigma_{v,2}^2, \ldots,\; \sigma_{v,N}^2\} \tag{11.219}$$

Then, it holds from (11.53) and (11.215) that $\mathcal{S}$ can be expressed as the Kronecker product:

$$\mathcal{S} \;=\; R_v \otimes G \tag{11.220}$$

Using the series representation (11.178) we have

$$\mathrm{MSD}_{\mathrm{dist,av}} \;=\; \frac{1}{hN}\sum_{n=0}^{\infty} \mathrm{Tr}\left[\mathcal{B}^n \mathcal{Y}\,(\mathcal{B}^*)^n\right] \tag{11.221}$$

where $h = 1$ for real data and, from the expressions in Theorem 11.2, the matrices $\mathcal{B}$ and $\mathcal{Y}$ are given by the following relations for the various strategies:

$$\left\{\begin{array}{lcllcl} \mathcal{B}_{\rm ncop} &=& I_N \otimes (I_{hM} - \mu H), & \mathcal{Y}_{\rm ncop} &=& \mu^2(R_v \otimes G) \\ \mathcal{B}_{\rm cons} &=& A^{\mathsf{T}} \otimes I_{hM} - \mu(I_{hM} \otimes H), & \mathcal{Y}_{\rm cons} &=& \mu^2(R_v \otimes G) \\ \mathcal{B}_{\rm atc} &=& A^{\mathsf{T}} \otimes (I_{hM} - \mu H), & \mathcal{Y}_{\rm atc} &=& \mu^2(A^{\mathsf{T}} R_v A \otimes G) \\ \mathcal{B}_{\rm cta} &=& A^{\mathsf{T}} \otimes (I_{hM} - \mu H), & \mathcal{Y}_{\rm cta} &=& \mu^2(R_v \otimes G) \end{array}\right. \tag{11.222}$$

We already know from Example 10.1 that, in general, $\rho(\mathcal{B}_{\rm diff}) \leq \rho(\mathcal{B}_{\rm ncop})$ so that diffusion strategies have a stabilizing effect. For the current data structure, it holds that these spectral radii are equal. Indeed, since A is a left-stochastic matrix, its spectral radius is given by $\rho(A) = 1$. Then,

$$\begin{aligned} \rho(\mathcal{B}_{\rm diff}) &= \rho[A^{\mathsf{T}} \otimes (I_{hM} - \mu H)] \\ &= \rho(A)\,\rho(I_{hM} - \mu H) \\ &= \rho(I_{hM} - \mu H) \\ &= \rho(\mathcal{B}_{\rm ncop}) \end{aligned} \tag{11.223}$$

On the other hand, let $\lambda_\ell(A)$ denote any of the eigenvalues of A. Since we know that $1 \in \{\lambda_\ell(A)\}$, it then follows:

$$\begin{aligned} \rho(\mathcal{B}_{\rm ncop}) &= \max_{1 \leq m \leq 2M} |1 - \mu\lambda_m(H)| \\ &\leq \max_{1 \leq \ell \leq N} \max_{1 \leq m \leq 2M} |\lambda_\ell(A) - \mu\lambda_m(H)| \\ &\overset{(8.40)}{=} \rho(\mathcal{B}_{\rm cons}) \end{aligned} \tag{11.224}$$

In other words, we arrive at the following conclusion for the scenario under study:

$$\rho(\mathcal{B}_{\rm diff}) = \rho(\mathcal{B}_{\rm ncop}) \leq \rho(\mathcal{B}_{\rm cons}) \tag{11.225}$$

It follows from this result that the convergence rate of the diffusion network is generally superior to the convergence rate of the consensus network.

Not only the convergence rate is superior, but the MSD performance of the diffusion network is also superior. To see this, we first note that for consensus implementations, it is customary to employ a doubly-stochastic matrix A (see Appendix E in [208]). For example, a left-stochastic A that is also symmetric will be doubly-stochastic. For the derivation that follows, we shall therefore assume that A is symmetric, i.e., $A = A^{\mathsf{T}}$; the argument can be extended to matrices A that are "close-to-symmetric" (i.e., diagonalizable with left-eigenvectors $\{x_k\}$ that are practically orthogonal to each other) [248]. It is sufficient for this example to consider the case of symmetric combination policies, A.

Since A is now diagonalizable, it admits a Jordan canonical decomposition of the form [27, 99, 104, 113]:

$$A^{\mathsf{T}} = XDX^{-1} \tag{11.226}$$

where D is a diagonal matrix with the eigenvalues of A, and X is a similarity transformation. Let $\{x_n\}$ denote the columns of X and let $\{y_n^*\}$ denote the rows of X^{-1}. Then, it follows from (11.226) and the fact that $XX^{-1} = I_N$ that

$$\begin{cases} A^{\mathsf{T}}x_n = \lambda_n(A)x_n \\ y_\ell^* A^{\mathsf{T}} = \lambda_\ell(A)y_\ell^* \\ y_\ell^* x_k = \delta_{\ell k} \\ \ell, k = 1, 2, \ldots, N \end{cases} \tag{11.227}$$

so that the $\{x_n\}$ correspond to the right eigenvectors of A^{T} and the $\{y_m^*\}$ correspond to the left eigenvectors of A^{T}. We assume the eigenvectors $\{x_n\}$ are normalized to satisfy

$$\|x_n\|^2 = 1, \quad n = 1, 2, \ldots, N \tag{11.228}$$

Since A is symmetric, then X is an orthonormal matrix, i.e.,

$$x_\ell^* x_k = \delta_{\ell k} \tag{11.229}$$

Under conditions (11.214)–(11.215), and for sufficiently small step-size μ to ensure mean-square stability, we now verify that diffusion networks lead to better MSD performance (i.e., smaller MSD values) than consensus networks. In particular, we verify that the ATC diffusion strategy achieves the lowest network MSD in comparison to the other strategies:

$$\mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{atc}} \leq \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cta}} \leq \mathrm{MSD}_{\mathrm{ncop,av}} \tag{11.230}$$

$$\mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{atc}} \leq \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cons}} \tag{11.231}$$

Furthermore, if it holds that

$$1 \leq \mu\lambda_{\min}(H) < 2 \tag{11.232}$$

then we verify that the consensus strategy is the worst even in comparison to the non-cooperative strategy:

$$\mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{atc}} \leq \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cta}} \leq \mathrm{MSD}_{\mathrm{ncop,av}} \leq \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cons}} \tag{11.233}$$

To see this, we introduce the eigen-decompositions of the matrices A and H into (11.221) and compare the resulting MSD expressions for the various strategies. Let $\{\lambda_m(H) > 0\}$ denote the eigenvalues of the Hermitian and

positive-definite matrix H with orthonormal eigenvectors denoted by $\{z_m\}$ $(m = 1, 2, \ldots, hM)$:

$$Hz_m = \lambda_m(H)z_m, \quad m = 1, 2, 3, \ldots, hM \tag{11.234}$$

Substituting the eigen-decompositions of A from (11.227) and H from (11.234) into (11.221) gives, after some algebra:

$$\text{MSD}^{\text{atc}}_{\text{dist,av}} = \frac{\mu^2}{hN} \sum_{k=1}^{N} \sum_{m=1}^{hM} \frac{|\lambda_k(A)|^2 \, \|y_k\|^2_{R_v} \, \|z_m\|^2_G}{1 - |\lambda_k(A)|^2 \, [1 - \mu\lambda_m(H)]^2} \tag{11.235}$$

$$\text{MSD}^{\text{cta}}_{\text{dist,av}} = \frac{\mu^2}{hN} \sum_{k=1}^{N} \sum_{m=1}^{hM} \frac{\|y_k\|^2_{R_v} \, \|z_m\|^2_G}{1 - |\lambda_k(A)|^2 \, [1 - \mu\lambda_m(H)]^2} \tag{11.236}$$

$$\text{MSD}^{\text{cons}}_{\text{dist,av}} = \frac{\mu^2}{hN} \sum_{k=1}^{N} \sum_{m=1}^{hM} \frac{\|y_k\|^2_{R_v} \, \|z_m\|^2_G}{1 - |\lambda_k(A) - \mu\lambda_m(H)|^2} \tag{11.237}$$

$$\text{MSD}_{\text{ncop,av}} = \frac{\mu^2}{hN} \sum_{k=1}^{N} \sum_{m=1}^{hM} \frac{\|y_k\|^2_{R_v} \, \|z_m\|^2_G}{1 - (1 - \mu\lambda_m(H))^2} \tag{11.238}$$

Now note that since $|\lambda_k(A)| \leq 1$, it is obvious that

$$\text{MSD}^{\text{atc}}_{\text{dist,av}} \leq \text{MSD}^{\text{cta}}_{\text{dist,av}} \leq \text{MSD}_{\text{ncop,av}} \tag{11.239}$$

To compare ATC diffusion and consensus, it can be verified that the ratio of each term on the right-hand side of (11.235) to the corresponding term in (11.237) is smaller or equal to one [248]:

$$\frac{|\lambda_k(A)|^2 \, \left(1 - |\lambda_k(A) - \mu\lambda_m(H)|^2\right)}{1 - |\lambda_k(A)|^2 \, (1 - \mu\lambda_m(H))^2} \leq 1 \tag{11.240}$$

so that

$$\text{MSD}^{\text{atc}}_{\text{dist,av}} \leq \text{MSD}^{\text{cons}}_{\text{dist,av}} \tag{11.241}$$

We can further verify that the performance of the consensus strategy is worse than the non-cooperative strategy when the step-size satisfies $1 \leq \mu\lambda_{\min}(H) < 2$. This result is established by verifying that the ratio of the individual terms appearing in the sums (11.237)-(11.238) is upper bounded by one [248]:

$$\frac{1 - |\lambda_k(A) - \mu\lambda_m(H)|^2}{1 - (1 - \mu\lambda_m(H))^2} \leq 1 \tag{11.242}$$

■

Example 11.12 (MSD performance of consensus and diffusion networks)**.** The following example specializes the results of Example 11.11 to the case of MSE networks from Example 6.3. We reconsider the two-agent network from Example 10.2 with both agents running either the LMS consensus strategy (7.13) or the LMS diffusion strategies (7.22)–(7.23) albeit on real data (for which $h = 1$). We assume

$$\mu_1 = \mu_2 \equiv \mu \tag{11.243}$$

$$R_{u,1} = R_{u,2} \equiv \sigma_u^2 I_{hM} \tag{11.244}$$

$$0 < \mu\sigma_u^2 < 1 \tag{11.245}$$

The second condition (11.244) ensures that $H = 2\sigma_u^2 I_M$. The third condition (11.245) ensures that both agents are individually stable in the mean since the matrix $\mathcal{B}_{\rm ncop} = I_N \otimes (I_{hM} - \mu H)$ from Example 11.11 will be stable.

The eigenvalues of A defined by (10.129) are at $\lambda_1(A) = 1$ and $\lambda_2(A) = 1 - a - b$. Using the notation of Example 11.11, this situation corresponds to the case

$$\left\{\begin{array}{rcl} R_v &=& \mathrm{diag}\{\sigma_{v,1}^2, \sigma_{v,2}^2\} \\ G &=& 4\sigma_u^2\, I_M \\ H &=& 2\sigma_u^2\, I_M \end{array}\right. \tag{11.246}$$

In this case, expressions (11.235)–(11.238) reduce to (using $h = 1$ for real data):

$$\mathrm{MSD}_{\rm dist,av}^{\rm atc} = 2\mu^2\sigma_u^2 M\left[\frac{y_1^* R_v y_1}{1-(1-2\mu\sigma_u^2)^2} + \frac{y_2^* R_v y_2 (1-a-b)^2}{1-(1-a-b)^2(1-2\mu\sigma_u^2)^2}\right] \tag{11.247}$$

$$\mathrm{MSD}_{\rm dist,av}^{\rm cta} = 2\mu^2\sigma_u^2 M\left[\frac{y_1^* R_v y_1}{1-(1-2\mu\sigma_u^2)^2} + \frac{y_2^* R_v y_2}{1-(1-a-b)^2(1-2\mu\sigma_u^2)^2}\right] \tag{11.248}$$

$$\mathrm{MSD}_{\rm dist,av}^{\rm cons} = 2\mu^2\sigma_u^2 M\left[\frac{y_1^* R_v y_1}{1-(1-2\mu\sigma_u^2)^2} + \frac{y_2^* R_v y_2}{1-(1-a-b-2\mu\sigma_u^2)^2}\right] \tag{11.249}$$

$$\mathrm{MSD}_{\rm ncop,av} = 2\mu^2\sigma_u^2 M\left[\frac{y_1^* R_v y_1}{1-(1-2\mu\sigma_u^2)^2} + \frac{y_2^* R_v y_2}{1-(1-2\mu\sigma_u^2)^2}\right] \tag{11.250}$$

Note that the first terms inside the brackets of (11.247)-(11.250) are the same. Then, it can be verified that these MSD values are related as follows depending

on the region in space where the parameters (a, b) lie:

$$\begin{cases} \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cons}} \leq \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cta}}, & \text{if} \quad 0 \leq a+b \leq \frac{1-2\mu\sigma_u^2}{1-\mu\sigma_u^2} \\ \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cons}} \geq \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cta}}, & \text{if} \quad \frac{1-2\mu\sigma_u^2}{1-\mu\sigma_u^2} \leq a+b < 2(1-\mu\sigma_u^2) \\ \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cons}} \leq \mathrm{MSD}_{\mathrm{ncop,\ av}}, & \text{if} \quad 0 \leq a+b \leq 2(1-2\mu\sigma_u^2) \\ \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cons}} \geq \mathrm{MSD}_{\mathrm{ncop,\ av}}, & \text{if} \quad 2(1-2\mu\sigma_u^2) \leq a+b < 2(1-\mu\sigma_u^2) \end{cases} \tag{11.251}$$

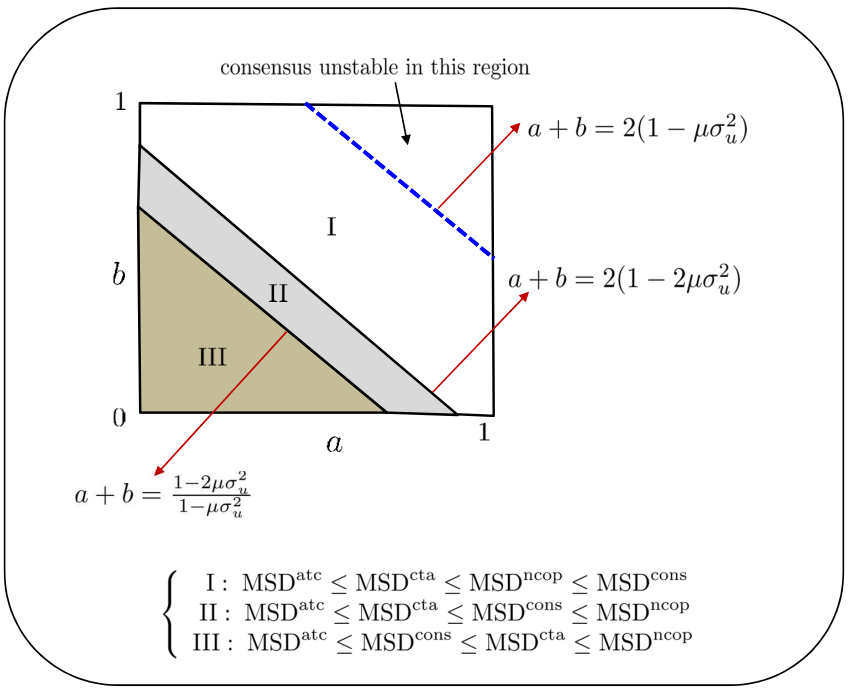

Figure 11.8: Comparison of the network MSD for $N = 2$ agents operating on complex-valued data. The consensus strategy is unstable when a and b lie above the dashed line in region I; it performs well in region III. ATC diffusion is superior in all three regions.

For example, the first relation can be established as follows:

$$
\begin{aligned}
&\mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cons}} \;\leq\; \mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{cta}} \\
&\Leftrightarrow \quad (1-a-b-2\mu\sigma_u^2)^2 \leq (1-a-b)^2(1-2\mu\sigma_u^2)^2 \\
&\Leftrightarrow \quad (a+b)^2 - 2(a+b)(1-2\mu\sigma_u^2) \leq [-2(a+b)+(a+b)^2](1-2\mu\sigma_u^2)^2 \\
&\Leftrightarrow \quad (a+b)^2\left[1-(1-2\mu\sigma_u^2)^2\right] - 2(a+b)(1-2\mu\sigma_u^2)[1-(1-2\mu\sigma_u^2)] \leq 0 \\
&\Leftrightarrow \quad 0 \leq (a+b) \leq \frac{4(1-2\mu\sigma_u^2)\mu\sigma_u^2}{1-(1-2\mu\sigma_u^2)^2} \\
&\Leftrightarrow \quad 0 \leq (a+b) \leq \frac{1-2\mu\sigma_u^2}{1-\mu\sigma_u^2} \qquad (11.252)
\end{aligned}
$$

and similarly for the other inequalities. We can therefore divide the $a \times b$ plane into three regions I, II, and III, as shown in Figure 11.8, where each region represents one possible relation among the MSD levels of the various strategies. The ATC diffusion strategy is seen to be superior in all regions, while the consensus strategy is worse than the non-cooperative strategy in region I and is also unstable in the mean for values of (a, b) lying above the dashed line in that region, i.e., for $a + b > 2(1 - \mu\sigma_u^2)$, as can be verified by following an argument similar to (10.135).

■

Example 11.13 (Higher-order terms in the MSD expression)**.** Continuing with Example 11.12, we can rework expression (11.247) for $\mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{atc}}$ into a more familiar form (and similarly for the other expressions). Thus, consider the eigenvectors $\{x_n, y_m\}$ defined by (11.227). Since A is left-stochastic, we have $A^{\mathsf{T}}\mathbb{1} = \mathbb{1}$. Note, however, from the definition of the eigenvectors $\{x_n\}$ that they need to satisfy the normalization condition (11.228). This means that we can select the first eigenvector as

$$x_1 = \frac{1}{\sqrt{N}}\,\mathbb{1} \qquad (11.253)$$

It then follows from the condition $y_1^* x_1 = 1$ that

$$y_1^*\mathbb{1} = \sqrt{N} \qquad (11.254)$$

so that the entries of the right-eigenvector y_1 add up to $\sqrt{N}$. Now recall from definition (11.136) for the Perron eigenvector p that its entries must add up to

one. Both p and y_1 are right-eigenvectors for A associated with the eigenvalue at one. Therefore, p and y_1 are related as follows:

$$p = \frac{1}{\sqrt{N}}\, y_1 \tag{11.255}$$

Using this result, and the fact that μ is sufficiently small and that we are dealing with a two-agent network in this example (so that $N = 2$), we can rewrite (11.247) to first-order in μ as follows:

$$\begin{aligned}
\mathrm{MSD}^{\mathrm{atc}}_{\mathrm{dist,av}} &= 2\mu^2\sigma_u^2 M\, \frac{y_1^* R_v y_1}{4\mu\sigma_u^2 - 4\mu^2\sigma_u^4} \\
&= 2\mu M\, \frac{N p^* R_v p}{4 - 4\mu\sigma_u^2} \\
&\approx \frac{\mu M}{2}\, 2\left(\sum_{k=1}^{2} p_k^2 \sigma_{v,k}^2\right), \quad \text{since } N = 2 \text{ and } \mu \text{ is small} \\
&= \mu M \sum_{k=1}^{2} p_k^2 \sigma_{v,k}^2
\end{aligned} \tag{11.256}$$

and we recover the analogue of expression (11.144) for real-data.

■

12

Benefits of Cooperation

Example 11.5 focused on MSE networks with quadratic costs and showed that for adaptation and learning under doubly-stochastic combination policies, it is not necessarily the case that every agent will benefit from cooperation with its neighbors. Some agents can see their performance degraded relative to what they would have attained had they operated independently of the other agents and in a non-cooperative manner. We verify in this chapter that the same conclusion holds for more general costs: doubly-stochastic combination policies enhance the average network performance albeit at the possible expense of some individual agents having their performance degrade relative to the non-cooperative scenario. One useful question to consider is whether it is possible to select combination matrices, A, that ensure that distributed (consensus or diffusion) networks will outperform the non-cooperative strategy both in terms of the overall average performance *and* the individual agent performance. The choice of A will generally need to be left-stochastic. We again recall that in order to carry a meaningful comparison with non-cooperative implementations, it is necessary to assume that all individual costs, $J_k(w)$, share the *same* global minimizer so that $w^\star = w^o$. It is also necessary to assume uniform step-sizes

across all agents since the performance of the non-cooperative agents is influenced by the step-sizes. Similarly, a meaningful comparison between distributed and centralized implementations requires that they employ the same step-size parameter and that both implementations approach the same limit point and, therefore, we also need to have $w^\star = w^o$. For these reasons, we shall assume in the sequel that

$$\mu_k \equiv \mu, \quad k = 1, 2, \ldots, N \tag{12.1}$$

For ease of reference we recall the expressions for the MSD performance of distributed (consensus and diffusion), centralized, and non-cooperative strategies for sufficiently small step-sizes, for both individual agents (when applicable) and for the average network performance:

$$\mathrm{MSD}_{\mathrm{cent}} = \frac{\mu}{2Nh}\,\mathrm{Tr}\left[\left(\sum_{k=1}^{N} H_k\right)^{-1}\left(\sum_{k=1}^{N} G_k\right)\right] \tag{12.2}$$

$$\mathrm{MSD}_{\mathrm{ncop},k} = \frac{\mu}{2h}\,\mathrm{Tr}\left(H_k^{-1} G_k\right) \tag{12.3}$$

$$\mathrm{MSD}_{\mathrm{ncop,av}} = \frac{\mu}{2Nh}\,\mathrm{Tr}\left[\sum_{k=1}^{N} H_k^{-1} G_k\right] \tag{12.4}$$

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{\mu}{2h}\mathrm{Tr}\left[\left(\sum_{k=1}^{N} p_k H_k\right)^{-1}\left(\sum_{k=1}^{N} p_k^2 G_k\right)\right] \tag{12.5}$$

In the analysis that follows, we assume that the various strategies are employing the *same* construction for their gradient vectors and that the moment matrices $\{G_k\}$ can be taken to be the same in all implementations. The matrices $\{H_k, G_k\}$ are defined by (11.12) in terms of the Hessian matrices of the individual costs, evaluated at $w^\star = w^o$, and in terms of the second-order moments of the gradient noise processes across the agents.

12.1 Doubly-Stochastic Combination Policies

Consider first the case in which the combination matrix, A, used by the consensus strategy (7.9) and the diffusion strategies (7.18) and (7.19) is

doubly stochastic. Then, the Perron eigenvector p defined by (11.136) is given by $p = \mathbb{1}/N$ so that all its entries are equal to $1/N$. In this case, expressions (12.2) and (12.5) lead to the conclusion that:

$$\mathrm{MSD}_{\mathrm{dist},k} \;=\; \mathrm{MSD}_{\mathrm{dist,av}} \;=\; \mathrm{MSD}_{\mathrm{cent}} \tag{12.6}$$

That is, the distributed consensus and diffusion strategies are able to attain the same MSD performance level as the centralized solution. Since we already showed in (5.80) that the centralized solution outperforms the non-cooperative solution, we conclude that the distributed solutions also outperform the non-cooperative solution:

$$\mathrm{MSD}_{\mathrm{dist,av}} \;=\; \mathrm{MSD}_{\mathrm{cent}} \;\leq\; \mathrm{MSD}_{\mathrm{ncop,av}} \tag{12.7}$$

Result (12.7) is in terms of the average network performance (obtained by averaging the MSD levels of the individual agents). In this way, the result establishes that the average MSD performance of the distributed solution is superior (i.e., lower) than the average MSD performance attained by the agents in a non-cooperative implementation. This conclusion motivates the following inquiry: is the improvement in network performance attained at the expense of deterioration in the performance of some of the agents? In other words, will the performance of some agents in the distributed solution become worse than what it would be if they operate independently? If this is the case, then result (12.7) would mean that in moving from non-cooperation to cooperation, some agents see their performance improve while other agents see their performance degrade in such a manner that the net effect for the network is a better (i.e., lower) average MSD value. We now verify that this is indeed the case for doubly-stochastic combination policies.

From (12.3) and (12.5) we observe that, to first-order in the step-size parameter, the MSD of the individual agents in the distributed implementation will be smaller (and, hence, better) than the MSD of the individual agents in the non-cooperative implementation only when for each $k = 1, 2, \ldots, N$:

$$\frac{1}{N}\,\mathrm{Tr}\left[\left(\sum_{k=1}^{N} H_k\right)^{-1}\left(\sum_{k=1}^{N} G_k\right)\right] \;\leq\; \mathrm{Tr}(H_k^{-1}G_k) \tag{12.8}$$

Unfortunately, this condition may or may not hold as illustrated by the next example. Agents for which the condition is violated would experience deterioration in their MSD level from cooperation. Before presenting the example, though, we mention that there are situations where condition (12.8) holds for all agents, in which case all agents will benefit from cooperation. This happens, for example, when the Hessian matrices, H_k, and the gradient noise covariances, G_k, are uniform across the agents, namely, when

$$H_k \equiv H, \quad G_k \equiv G, \quad k = 1, 2, \ldots, N \tag{12.9}$$

The condition also holds when the following two requirements hold for each $k = 1, 2, \ldots, N$:

$$H_k \equiv H \tag{12.10}$$

$$\frac{1}{N} \operatorname{Tr}\left[\sum_{k=1}^{N} H^{-1} G_k\right] \leq N \operatorname{Tr}(H^{-1} G_k) \tag{12.11}$$

We summarize the main conclusion so far in the following statement. We illustrated this conclusion earlier in Example 11.5.

Lemma 12.1 (Doubly-stochastic combination policies)**.** Assume all agents employ the same step-size parameter and that the individual costs are strongly-convex and their minimizers coincide with each other. For doubly stochastic combination matrices it holds that

$$\text{MSD}_{\text{dist,av}} = \text{MSD}_{\text{cent}} \leq \text{MSD}_{\text{ncop,av}} \tag{12.12}$$

Example 12.1 (Doubly-stochastic policies over MSE networks)**.** We reconsider the setting of Example 11.4, which deals with MSE networks operating on real-valued data and refer to the strongly-connected network of Figure 11.1 with $N = 20$ agents. We assume uniform step-sizes, $\mu_k \equiv \mu = 6 \times 10^{-4}$, and uniform regression covariance matrices of the form $R_{u,k} = \sigma_u^2 I_M$ where $\sigma_u^2 = 2$. In this setting, we have

$$H_k = 2\sigma_u^2 I_M \equiv H, \qquad G_k = 4\sigma_{v,k}^2 \, \sigma_u^2 I_M, \quad \theta_k^2 = 2M\sigma_{v,k}^2 \tag{12.13}$$

We consider two scenarios. In the first case, the agents run the ATC diffusion strategy (7.23) with the Metropolis combination weights (8.100), namely,

$$\left\{\begin{array}{rcl} \boldsymbol{\psi}_{k,i} &=& \boldsymbol{w}_{k,i-1} + 2\mu \boldsymbol{u}_{k,i}^{\mathsf{T}} \left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}\right] \\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{\psi}_{\ell,i} \end{array}\right. \tag{12.14}$$

The Metropolis weights result in a doubly-stochastic combination matrix, A, so that $p_k = 1/N$. In the second case, the agents transfer the data to a fusion center running the centralized strategy (5.13), i.e.,

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu \left(\frac{1}{N}\sum_{k=1}^{N} 2\boldsymbol{u}_{k,i}^{\mathsf{T}}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{i-1})\right) \tag{12.15}$$

The resulting MSD performance levels are given by expressions (12.2) and (12.5), which in the current setting reduce to (using $h = 1$ for real data):

$$\mathrm{MSD}_{\mathrm{cent}} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{\mu M}{N}\left(\frac{1}{N}\sum_{k=1}^{N}\sigma_{v,k}^2\right) \tag{12.16}$$

We illustrate these results numerically in Figure 12.1 for the two algorithms listed above running on complex-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ generated according to the model $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i)$, with $M = 10$ and where the noise profile is the same one shown earlier in the left plot of Figure 11.2. The unknown vector w^o is generated randomly and its norm is normalized to one. Figure 12.1 plots the evolution of the ensemble-average learning curves, $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ for diffusion and $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ for centralized and weighted centralized. The curves are obtained by averaging simulated trajectories over 100 repeated experiments. The label on the vertical axis in the figure refers to the learning curves by writing $\mathrm{MSD}(i)$, with an iteration index i. It is observed both strategies tend towards the same MSD level that is predicted by the theoretical expression (12.16). ■

12.2 Left-Stochastic Combination Policies

The previous analysis shows that under doubly-stochastic combination policies, cooperation among the agents enhances the *network* MSD performance albeit possibly at the expense of deterioration in the performance of some *individual* agents. A useful question to consider is whether it is possible to select combination matrices A that will ensure that distributed (consensus or diffusion) networks will outperform the

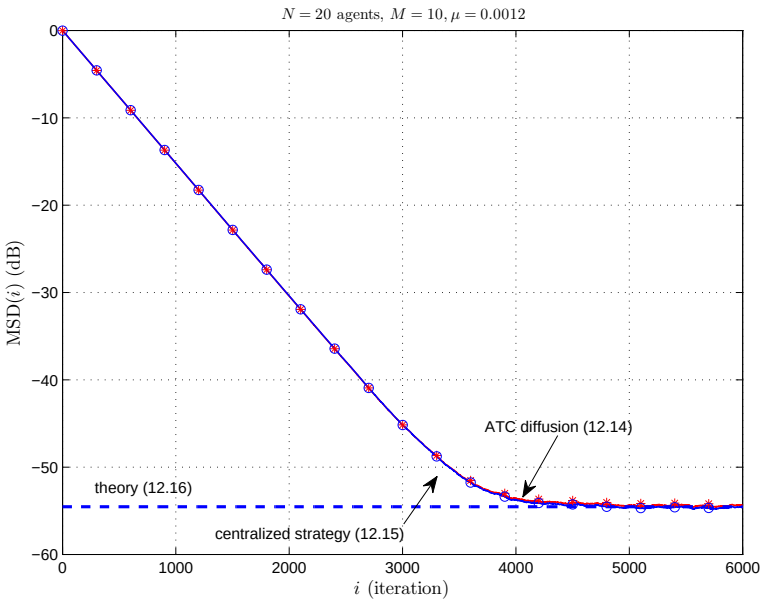

Figure 12.1: Evolution of the learning curves for two strategies: ATC diffusion (12.14) with Metropolis combination weights vs. centralized (12.15).

non-cooperative strategy *both* in terms of the overall network performance *and* the individual agent performance. We need to search over the larger set of left-stochastic matrices A since we already know that doubly-stochastic matrices A may not be sufficient to guarantee this property.

From expression (12.3) we observe that the performance of each agent in the non-cooperative mode of operation is dependent on its Hessian matrix, H_k. We therefore focus on the important special case in which these Hessian matrices are uniform across the agents:

$$H_k \equiv H, \quad k = 1, 2, \ldots, N \tag{12.17}$$

As explained earlier, this scenario is common in important situations of interest such as the MSE networks of Example 6.3 and in machine learning applications where all agents minimize the same cost function as in Examples 7.4 and 11.9. For a given network topology, we then consider the problem of minimizing the MSD level of the distributed

strategies under these conditions, namely,

$$A^o \triangleq \arg\min_{A\in\mathbb{A}} \ \mathrm{Tr}\left(\sum_{k=1}^{N} p_k^2 H^{-1} G_k\right) \quad \text{subject to} \ \ Ap = p, \ \ \mathbb{1}^{\mathsf{T}} p = 1, \ \ p_k > 0 \tag{12.18}$$

where the symbol $\mathbb{A}$ denotes the set of all $N\times N$ primitive left-stochastic matrices A whose entries $\{a_{\ell k}\}$ satisfy conditions (7.10). To solve the above problem, we start by introducing the nonnegative scalars:

$$\theta_k^2 \triangleq \mathrm{Tr}(H^{-1}G_k), \quad k = 1, 2, \ldots, N \tag{12.19}$$

and refer to them as gradient-noise factors (since they incorporate information about the gradient noise moments, G_k). Comparing with (12.3), the scalar θ_k^2 is seen to be proportional to the MSD level at agent k in the non-cooperative mode of operation. Interpreting every $A \in \mathbb{A}$ as the probability transition matrix of an irreducible aperiodic Markov chain [169, 186], and using a construction procedure developed in [42, 106], it was argued in [276] that one choice for an optimal A^o that solves optimization problems of the form (12.18) is the following left-stochastic matrix (which we refer to as the Hastings combination rule).

Lemma 12.2 (Hastings rule)**.** The following combination matrix, denoted by A^o with a superscript o, is a solution to the optimization problem (12.18):

$$a_{\ell k}^o = \begin{cases} \dfrac{\theta_k^2}{\max\{\, n_k\theta_k^2, \ n_\ell\theta_\ell^2 \,\}}, & \ell \in \mathcal{N}_k\backslash\{k\} \\ 1 - \displaystyle\sum_{m\in\mathcal{N}_k\backslash\{k\}} a_{mk}^o, & \ell = k \end{cases} \tag{12.20}$$

where $n_k = |\mathcal{N}_k|$ denotes the cardinality of $\mathcal{N}_k$ or the degree of agent k (i.e., number of its neighbors). The entries of the corresponding Perron eigenvector are given by

$$p_k^o = \frac{1}{\theta_k^2}\left(\sum_{\ell=1}^{N}\frac{1}{\theta_\ell^2}\right)^{-1}, \quad k = 1, 2, \ldots, N \tag{12.21}$$

Proof. We first consider the optimization problem (12.18) without the eigenvector constraint, $Ap = p$, and minimize instead over the positive scalars $\{p_k\}$:

$$p_k^o \triangleq \arg\min_{p_k} \sum_{k=1}^{N} p_k^2 \theta_k^2 \quad \text{subject to } \mathbb{1}^{\mathsf{T}} p = 1, \;\; p_k > 0 \tag{12.22}$$

It is easy to verify that the solution to this problem is given by (12.21). Next, we verify that the matrix A^o defined by (12.20) is a left-stochastic primitive matrix that has $p^o = \mathrm{col}\{p_k^o\}$ as its Perron eigenvector.

To begin with, it is straightforward to verify from (12.20) that A^o is left-stochastic. We now establish that $A^o p^o = p^o$, i.e., for every $1 \le \ell \le N$:

$$\sum_{k=1}^{N} a_{\ell k}^o p_k^o \;=\; p_\ell^o \tag{12.23}$$

For this purpose, we note first that for any $\ell \neq k$, the following balanced relation holds:

$$\begin{aligned} a_{\ell k}^o \, p_k^o &= \left(\frac{\theta_k^2}{\max\{ \, n_k \theta_k^2, \; n_\ell \theta_\ell^2 \, \}} \right) \frac{1}{\theta_k^2} \left(\sum_{\ell=1}^{N} \frac{1}{\theta_\ell^2} \right)^{-1} \\ &= \left(\frac{1}{\max\{ \, n_k \theta_k^2, \; n_\ell \theta_\ell^2 \, \}} \right) \left(\sum_{\ell=1}^{N} \frac{1}{\theta_\ell^2} \right)^{-1} \\ &= a_{k\ell}^o \, p_\ell^o \end{aligned} \tag{12.24}$$

so that

$$\begin{aligned} \sum_{k=1}^{N} a_{\ell k}^o p_k^o &= \sum_{k \neq \ell} a_{\ell k}^o p_k^o \; + \; a_{\ell\ell} p_\ell^o \\ &\overset{(12.24)}{=} \sum_{k \neq \ell} a_{k\ell}^o p_\ell^o \; + \; a_{\ell\ell} p_\ell^o \\ &= \sum_{k=1} a_{k\ell}^o p_\ell^o \\ &= \left(\sum_{k=1} a_{k\ell}^o \right) p_\ell^o \\ &= p_\ell^o \quad (\text{since } A^o \text{ is left-stochastic}) \end{aligned} \tag{12.25}$$

It remains to show that A^o is primitive. To do so, and in view of Lemma 6.1, it is sufficient to show that $a_{kk}^o > 0$ for some k. This property actually holds

for all diagonal entries a_{kk}^o in this case. Indeed, note that since

$$a_{\ell k}^o \;=\; \frac{\theta_k^2}{\max\{\, n_k\theta_k^2,\; n_\ell\theta_\ell^2 \,\}} \;\leq\; \frac{\theta_k^2}{n_k\theta_k^2} \;\leq\; \frac{1}{n_k} \tag{12.26}$$

we get

$$\begin{aligned}
\sum_{k\neq\ell} a_{\ell k}^o \;&=\; \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} a_{\ell k}^o \\
&\leq\; \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} \frac{1}{n_k} \\
&=\; \frac{n_k-1}{n_k}
\end{aligned} \tag{12.27}$$

which implies that

$$\begin{aligned}
a_{kk}^o \;&=\; 1 - \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} a_{\ell k}^o \\
&\geq\; 1 - \frac{n_k-1}{n_k} \\
&=\; \frac{1}{n_k} \\
&>\; 0
\end{aligned} \tag{12.28}$$

□

The Hastings rule is a fully-distributed solution — each agent k only needs to obtain the products $\{n_\ell\theta_\ell^2\}$ from its neighbors to compute the combination weights $\{a_{\ell k}^o\}$. Substituting (12.21) into (12.18), we find that the resulting optimal value for the distributed network MSD is:

$$\mathrm{MSD}_{\mathrm{dist,av}}^o \;=\; \frac{\mu}{2h}\left(\sum_{\ell=1}^N \frac{1}{\theta_\ell^2}\right)^{-1} \tag{12.29}$$

At the same time, it follows from (12.5) that the MSD performance of the distributed network for any doubly-stochastic (d.s.) matrix A is:

$$\mathrm{MSD}_{\mathrm{dist,av}}^{\mathrm{d.s.}} \;=\; \frac{\mu}{2N^2h}\left(\sum_{\ell=1}^N \theta_\ell^2\right) \tag{12.30}$$

Now, using the following algebraic property [206], which is valid for any scalars $\{\theta_\ell^2\}$:

$$N^2 \leq \left(\sum_{\ell=1}^{N} \theta_\ell^2\right)\left(\sum_{\ell=1}^{N} \frac{1}{\theta_\ell^2}\right) \tag{12.31}$$

we conclude that

$$\mathrm{MSD}^{o}_{\mathrm{dist,av}} \;\leq\; \mathrm{MSD}^{\mathrm{d.s.}}_{\mathrm{dist,av}} \;\leq\; \mathrm{MSD}_{\mathrm{ncop,av}} \tag{12.32}$$

so that, as expected, the MSD of the distributed (consensus or distributed) network with the optimal left-stochastic matrix, A^o, is also superior to the MSD of the non-cooperative network. More importantly, though, this optimal choice for A leads to the following performance level at the individual agents in the distributed solution:

$$\begin{aligned}
\mathrm{MSD}^{o}_{\mathrm{dist},k} &= \frac{\mu}{2h}\left(\sum_{\ell=1}^{N}\frac{1}{\theta_\ell^2}\right)^{-1} \\
&\leq \frac{\mu}{2h}\left(\frac{1}{\theta_k^2}\right)^{-1} \\
&\overset{(12.3)}{=} \mathrm{MSD}_{\mathrm{ncop},k}, \quad k=1,2,\ldots,N
\end{aligned} \tag{12.33}$$

so that, to first-order in the step-size parameter, the individual agent performance in the optimized distributed network is improved across all agents relative to the non-cooperative case:

$$\mathrm{MSD}^{o}_{\mathrm{dist},k} \;\leq\; \mathrm{MSD}_{\mathrm{ncop},k}, \quad k=1,2,\ldots,N \tag{12.34}$$

We summarize the main conclusion in the following statement.

Lemma 12.3 (Left-stochastic combination policies). Assume all agents employ the same step-size parameter and that the individual costs are strongly-convex and their minimizers coincide with each other. Assume further that the Hessian matrices evaluated at the optimal solution, w^o, are uniform across all agents as in (12.17). For the left-stochastic Hastings policy (12.20) it holds that

$$\mathrm{MSD}^{o}_{\mathrm{dist,av}} \;\leq\; \mathrm{MSD}^{\mathrm{d.s.}}_{\mathrm{dist,av}} \;\leq\; \mathrm{MSD}_{\mathrm{ncop,av}} \tag{12.35}$$

$$\mathrm{MSD}^{o}_{\mathrm{dist},k} \;\leq\; \mathrm{MSD}_{\mathrm{ncop},k}, \quad k=1,2,\ldots,N \tag{12.36}$$

Example 12.2 (Optimal combination policy for MSE networks)**.** Let us reconsider the setting of Example 11.3, which deals with MSE networks. We assume uniform step-sizes and uniform regression covariances, i.e., $\mu_k \equiv \mu$ and $R_{u,k} \equiv R_u$ for $k = 1, 2, \ldots, N$. In this setting we have

$$H_k = \begin{bmatrix} R_u & 0 \\ 0 & R_u^{\mathsf{T}} \end{bmatrix} \equiv H, \qquad G_k = \sigma_{v,k}^2 \begin{bmatrix} R_u & \times \\ \times & R_u^{\mathsf{T}} \end{bmatrix}, \quad \theta_k^2 = 2M\sigma_{v,k}^2 \tag{12.37}$$

For these values of $\{H_k, G_k\}$, the optimization problem (12.18) reduces to

$$\begin{aligned} A^o \;\triangleq\; & \arg\min_{A \in \mathbb{A}} \sum_{k=1}^{N} p_k^2 \sigma_{v,k}^2 \\ & \text{subject to } \; Ap = p, \;\; \mathbb{1}^{\mathsf{T}} p = 1, \;\; p_k > 0 \end{aligned} \tag{12.38}$$

which is of course the same problem we would be motivated to optimize had we started from the MSD expression (11.147). Using (12.20), an optimal solution is given by

$$a_{\ell k}^o \;=\; \begin{cases} \dfrac{\sigma_{v,k}^2}{\max\{\, n_k \sigma_{v,k}^2,\; n_\ell \sigma_{v,\ell}^2 \,\}}, & \ell \in \mathcal{N}_k \backslash \{k\} \\ 1 - \displaystyle\sum_{m \in \mathcal{N}_k \backslash \{k\}} a_{mk}^o, & \ell = k \end{cases} \tag{12.39}$$

with

$$\mathrm{MSD}_{\mathrm{dist},k}^o = \mathrm{MSD}_{\mathrm{dist,av}}^o \;=\; \frac{\mu M}{2} \left(\sum_{\ell=1}^{N} \frac{1}{\sigma_{v,\ell}^2} \right)^{-1} \tag{12.40}$$

Note that

$$\mathrm{MSD}_{\mathrm{dist},k}^o \;\leq\; \frac{\mu M}{2} \left(\frac{1}{\sigma_{v,k}^2} \right)^{-1} \overset{(12.3)}{=} \mathrm{MSD}_{\mathrm{ncop},k} \tag{12.41}$$

so that the individual agent performance in the optimized distributed network is improved across all agents relative to the non-cooperative case.

■

Example 12.3 (Optimal MSD combination policy for online learning)**.** We revisit Example 11.9, which deals with a collection of N learners. Using the notation of that example we have that, in this case, the gradient-noise factors $\{\theta_k^2\}$ are now uniform:

$$\theta_k^2 \equiv \theta^2 \;=\; \mathrm{Tr}(H^{-1} R_s) \tag{12.42}$$

Substituting into expression (12.20) for Hastings rule, we find that the optimal combination coefficients reduce to the following so-called Metropolis rule, which we encountered earlier in Example 8.9:

$$a_{\ell k}^{o} = \begin{cases} \dfrac{1}{\max\{\, n_k,\, n_\ell \,\}}, & \ell \in \mathcal{N}_k \backslash \{k\} \\ 1 - \displaystyle\sum_{m \in \mathcal{N}_k \backslash \{k\}} a_{mk}^{o}, & \ell = k \end{cases} \tag{12.43}$$

Therefore, the optimal combination policy happens to be doubly-stochastic in this case. Observe that the above combination coefficients now depend solely on the degrees of the agents (i.e., the extent of their connectivity). Moreover, from (12.29) and using $h = 1$ for real data, the optimal MSD value is given by

$$\text{MSD}_{\text{dist,av}}^{o} = \frac{\mu}{4} \left(\frac{1}{N} \right) \text{Tr}(H^{-1} R_s) \tag{12.44}$$

which, in this case, agrees with the performance of the centralized solution given by (12.2). On the other hand, for arbitrary left-stochastic combination matrices A, the MSD performance of the distributed (consensus and diffusion) solutions can be deduced from (12.5) and would be given by

$$\text{MSD}_{\text{dist,av}} = \frac{\mu}{4} \left(\sum_{k=1}^{N} p_k^2 \right) \text{Tr}(H^{-1} R_s) \tag{12.45}$$

■

12.3 Comparison with Centralized Solutions

The third question we consider in this chapter is to compare the optimal MSD performance of the distributed consensus and diffusion solutions (resulting from the use of the Hastings rule (12.20)), with the MSD performance of the centralized solution under the same condition (12.17) of uniform Hessian matrices. In this case, from expressions (12.2) and (12.29), the MSD levels of the centralized and (optimized) distributed solutions are given by:

$$\text{MSD}_{\text{cent}} = \frac{\mu}{2N^2 h} \left(\sum_{\ell=1}^{N} \theta_\ell^2 \right) \tag{12.46}$$

$$\text{MSD}_{\text{dist,av}}^{o} = \frac{\mu}{2h} \left(\sum_{\ell=1}^{N} \frac{1}{\theta_\ell^2} \right)^{-1} \tag{12.47}$$

Using the inequality (12.31) again, we readily conclude that, to first-order in the step-size parameter,

$$\mathrm{MSD}^{o}_{\mathrm{dist,av}} \;\leq\; \mathrm{MSD}_{\mathrm{cent}} \tag{12.48}$$

so that the optimized distributed network running the consensus strategy (7.9) or the diffusion strategies (7.18) or (7.19) with the Hasting combination rule (12.20) *outperforms* the centralized solution (5.22), which we repeat below for ease of reference

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \mu \left(\frac{1}{N} \sum_{k=1}^{N} \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{i-1}) \right), \quad i \geq 0 \tag{12.49}$$

The conclusion that the distributed solutions outperform the centralized solution may seem puzzling at first. However, this result follows from the fact that the optimized combination coefficients (12.20) for the distributed implementations exploit information about the gradient noise factors, $\{\theta_\ell^2\}$. This information is not used by the centralized algorithm (12.49). We can of course modify (12.49) to include information about the gradient noise factors as well.

Weighted Centralized Strategy

One way to modify the centralized solution (12.49) is as follows [279]. We incorporate the positive weighting coefficients $\{p_k^o\}$ into the centralized update equation:

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \mu \left(\sum_{k=1}^{N} p_k^o \, \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{i-1}) \right), \quad i \geq 0 \tag{12.50}$$

where the p_k^o were defined earlier in (12.21):

$$p_k^o \;\triangleq\; \frac{1}{\theta_k^2} \left(\sum_{\ell=1}^{N} \frac{1}{\theta_\ell^2} \right)^{-1}, \quad k = 1, 2, \ldots, N \tag{12.51}$$

The MSD performance of the weighted centralized solution (12.50) can be verified to match that of the optimized distributed solution (12.47). Indeed, compared with (12.49), we can interpret algorithm (12.50) as corresponding to the centralized stochastic gradient implementation

that would result from minimizing instead the following modified global cost

$$J^{\text{glob,b}}(w) \triangleq \sum_{k=1}^{N} J_k^b(w) \tag{12.52}$$

where each individual cost is a scaled version of the original cost:

$$J_k^b(w) \triangleq Np_k^o \, J_k(w) \tag{12.53}$$

In this way, the gradient noise vectors that result from using the modified costs $\{J_k^b(w)\}$ will be scaled by the same factors $\{Np_k^o\}$ relative to the gradient noise vectors that result from using the original costs $\{J_k(w)\}$. Specifically, if we denote the individual gradient noise process corresponding to implementation (12.49) by

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1}) = \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{i-1}) - \nabla_{w^*} J_k(\boldsymbol{w}_{i-1}) \tag{12.54}$$

then the gradient noise process that corresponds to implementation (12.50) will be given by

$$\begin{aligned} \boldsymbol{s}_{k,i}^b(\boldsymbol{w}_{i-1}) &\triangleq \widehat{\nabla_{w^*} J}_k^b(\boldsymbol{w}_{i-1}) - \nabla_{w^*} J_k^b(\boldsymbol{w}_{i-1}) \\ &= Np_k^o \, \boldsymbol{s}_{k,i}(\boldsymbol{w}_{i-1}) \end{aligned} \tag{12.55}$$

under the reasonable expectation that the gradient vector approximation, $\widehat{\nabla_{w^*} J}_k^b(\boldsymbol{w}_{i-1})$, is similarly scaled by Np_k^o. Consequently, the limiting moment matrices corresponding to the new gradient noise vectors, $\{\boldsymbol{s}_{k,i}^b(w^o)\}$, will be scaled multiples of the moment matrices corresponding to the previous gradient noise vectors $\{\boldsymbol{s}_{k,i}(w^o)\}$, i.e.,

$$R_{s,k}^b = (Np_k^o)^2 \, R_{s,k} \tag{12.56}$$

$$R_{q,k}^b = (Np_k^o)^2 \, R_{q,k}, \quad k = 1, 2, \ldots, N \tag{12.57}$$

It follows from definition (5.56) that the matrices $\{H^b, G_k^b\}$ for the weighted centralized solution (12.50) are related to the matrices $\{H, G_k\}$ for the original centralized solution (12.49) as follows:

$$H^b = Np_k^o H \tag{12.58}$$

$$G_k^b = (Np_k^o)^2 \, G_k, \quad k = 1, 2, \ldots, N \tag{12.59}$$

and, therefore, the corresponding gradient noise factors $\{\theta_k^2, (\theta_k^b)^2\}$ are related as

$$\left(\theta_k^b\right)^2 = N p_k^o \theta_k^2, \quad k = 1, 2, \ldots, N \tag{12.60}$$

Substituting into (12.46) we find that the MSD level for the weighted centralized solution, denoted by MSD_{wcen} is given by

$$\begin{aligned}
\text{MSD}_{\text{wcen}} &= \frac{\mu}{2N^2 h} \sum_{\ell=1}^{N} \left(\theta_\ell^b\right)^2 \\
&= \frac{\mu}{2N^2 h} \sum_{\ell=1}^{N} N p_\ell^o \theta_\ell^2 \\
&\overset{(12.51)}{=} \frac{\mu}{2h} \left(\sum_{\ell=1}^{N} \frac{1}{\theta_\ell^2}\right)^{-1} \\
&\overset{(12.47)}{=} \text{MSD}_{\text{dist,av}}^o
\end{aligned} \tag{12.61}$$

We conclude that it is possible to modify the centralized solution into the weighted form (12.50) such that the MSD performance of the optimal distributed solution matches the MSD performance of the weighted centralized solution.

Example 12.4 (Comparing distributed and centralized solutions)**.** We reconsider the setting of Example 11.3, which deals with MSE networks. We assume uniform step-sizes, $\mu_k \equiv \mu = 0.001$, and real-valued data with uniform regression covariance matrices of the form $R_{u,k} = \sigma_u^2 I_M$ where σ_u^2 is chosen randomly from within the range $[1, 2]$. In this setting, we have

$$H_k = 2\sigma_u^2 I_M \equiv H, \qquad G_k = 4\sigma_{v,k}^2 \, \sigma_u^2 I_M, \quad \theta_k^2 = 2M\sigma_{v,k}^2 \tag{12.62}$$

We consider three scenarios. In the first case, the agents run the ATC diffusion strategy (7.23), namely,

$$\begin{cases}
\boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{k,i-1} + 2\mu \boldsymbol{u}_{k,i}^{\mathsf{T}} \left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1}\right] \\
\boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k}^o \, \boldsymbol{\psi}_{\ell,i}
\end{cases} \tag{12.63}$$

where the combination weights $\{a_{\ell k}^o\}$ are the Hastings weights from (12.39). In the second case, the agents transfer the data to a fusion center running the

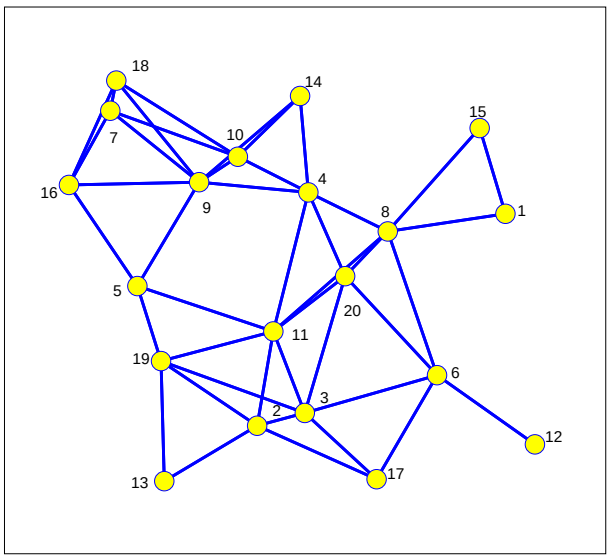

Figure 12.2: A connected network topology consisting of $N = 20$ agents employing the averaging rule (11.148).

conventional (un-weighted) centralized strategy (5.13), i.e.,

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu \left(\frac{1}{N} \sum_{k=1}^{N} 2\boldsymbol{u}_{k,i}^{\mathsf{T}} (\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{i-1}) \right) \tag{12.64}$$

In the third case, the fusion center employs a weighted centralized solution of the form:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu \left(\sum_{k=1}^{N} 2p_k^o \boldsymbol{u}_{k,i}^{\mathsf{T}} (\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{i-1}) \right) \tag{12.65}$$

where the $\{p_k^o\}$ are the entries of the Perron vector given by (12.21), which in the current setting reduces to:

$$p_k^o = \frac{1}{\sigma_{v,k}^2} \left(\sum_{\ell=1}^{N} \frac{1}{\sigma_{v,\ell}^2} \right)^{-1}, \quad k = 1, 2, \ldots, N \tag{12.66}$$

The resulting MSD performance levels are given by expressions (12.46)–

(12.47) and (12.61) using $h = 1$:

$$\text{MSD}_{\text{cent}} = \frac{\mu}{2N^2}\left(\sum_{\ell=1}^{N}\theta_\ell^2\right) \tag{12.67}$$

$$\text{MSD}_{\text{wcent}} = \text{MSD}^o_{\text{dist,av}} = \frac{\mu}{2}\left(\sum_{\ell=1}^{N}\frac{1}{\theta_\ell^2}\right)^{-1} \tag{12.68}$$

where $\theta_\ell^2 = 2M\sigma_{v,\ell}^2$.

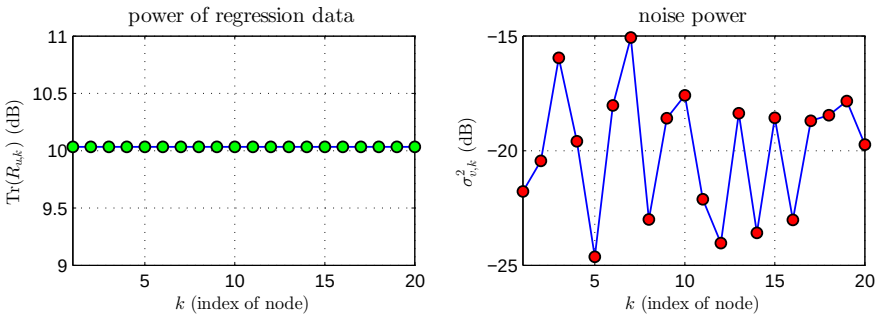

Figure 12.3: Regression data power (left) and measurement noise profile (right) across all agents in the network. The covariance matrices are assumed to be of the form $R_{u,k} = \sigma_u^2 I_M$, and the noise and regression data are Gaussian distributed in this simulation.

We illustrate these results numerically for the connected network topology shown in Figure 12.2 with $N = 20$ agents. The measurement noise variances, $\{\sigma_{v,k}^2\}$, and the power of the regression data, are shown in the plots of Figure 12.3, respectively. All agents are assumed to have a non-trivial self-loop so that the neighborhood of each agent includes the agent itself as well. The resulting network is therefore strongly-connected.

Figure 12.4 plots the resulting learning curves for the three algorithms listed above: ATC diffusion, centralized, and weighted centralized running on real-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ generated according to the model $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i)$, with $M = 10$. The unknown vector w^o is generated randomly and its norm is normalized to one. The figure plots the evolution of the ensemble-average learning curves, $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ for diffusion and $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ for centralized and weighted centralized. The curves are obtained by averaging simulated trajectories over 100 repeated experiments. The labels on the vertical axes in the figures refer to the learning curves by writing $\text{MSD}(i)$, with an iteration index i. It is seen in the figure that the MSD level that is attained

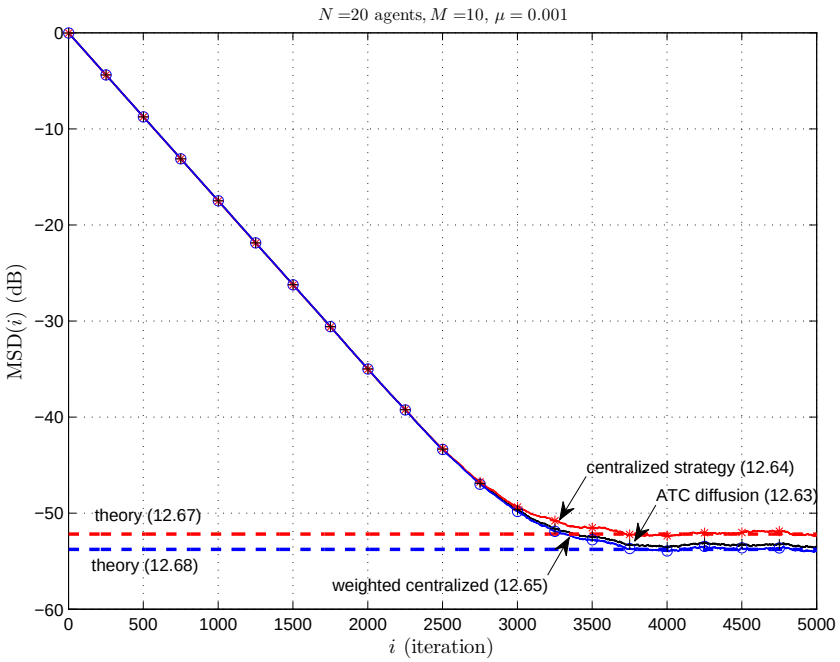

Figure 12.4: Evolution of the learning curves for ATC diffusion (12.63), un-weighted centralized strategy (12.64), and weighted centralized strategy (12.65).

by the diffusion strategy is better (lower) than the MSD level that is attained by the un-weighted centralized strategy, in agreement with the theoretical result (12.48). On the other hand, the same figure shows that the weighted centralized solution (12.65) eliminates the degradation in performance, again in agreement with the theoretical result (12.61).

■

12.4 Excess-Risk Performance

We focused in the previous sections on the MSD performance measure. The same conclusions extend to the ER performance measure and, therefore, we shall be brief. To begin with, for a meaningful comparison with the non-cooperative solution, we shall assume in this section that

all cost functions are uniform across the agents, namely,

$$J_k(w) \equiv J(w), \quad k = 1, 2, \ldots, N \tag{12.69}$$

The ER performance levels for the non-cooperative, centralized, and distributed strategies are then given by

$$\mathrm{ER}_{\mathrm{cent}} = \frac{\mu h}{4} \left(\frac{1}{N^2}\right) \mathrm{Tr}\left(\sum_{k=1}^{N} R_{s,k}\right) \tag{12.70}$$

$$\mathrm{ER}_{\mathrm{ncop},k} = \frac{\mu h}{4} \mathrm{Tr}\left(R_{s,k}\right) \tag{12.71}$$

$$\mathrm{ER}_{\mathrm{ncop,av}} = \frac{\mu h}{4} \left(\frac{1}{N}\right) \mathrm{Tr}\left(\sum_{k=1}^{N} R_{s,k}\right) \tag{12.72}$$

$$\mathrm{ER}_{\mathrm{dist},k} = \mathrm{ER}_{\mathrm{dist,av}} = \frac{\mu h}{4} \mathrm{Tr}\left(\sum_{k=1}^{N} p_k^2 R_{s,k}\right) \tag{12.73}$$

For doubly-stochastic combination matrices, and to first-order in the step-size parameter, it again holds that

$$\mathrm{ER}_{\mathrm{dist,av}} = \mathrm{ER}_{\mathrm{cent}} = \frac{1}{N} \mathrm{ER}_{\mathrm{ncop,av}} \tag{12.74}$$

This result is in terms of the average network performance (obtained by averaging the ER levels of the individual agents). In this way, the result establishes that the average ER performance of the distributed solution is $N-$fold superior (i.e., lower) than the average ER performance attained by the agents in a non-cooperative solution. However, from (12.71) and (12.73) we observe that the ER of the individual agents in the distributed implementation will be smaller (and, hence, better) than the ER of the individual agents in the non-cooperative implementation only when for each $k = 1, 2, \ldots, N$:

$$\frac{1}{N} \sum_{k=1}^{N} \mathrm{Tr}\left(R_{s,k}\right) \leq N \mathrm{Tr}(R_{s,k}) \tag{12.75}$$

Unfortunately, this condition may or may not hold. For example, if all the $\{R_{s,k}\}$ are uniform across the agents, then the condition is clearly satisfied and the performance of all individual agents will im-

prove through cooperation. On the other hand, if we consider the example $N = 2$, $R_{s,1} = rI_M$ and $R_{s,2} = 9rI_M$ for some $r > 0$. Then,

$$\frac{1}{N}\sum_{k=1}^{N}\mathrm{Tr}\,(R_{s,k}) \;=\; 5rI_M \tag{12.76}$$

which is larger than $2R_{s,1}$ but smaller than $2R_{s,2}$. In this case, agent 2 will benefit from cooperation while agent 1 will not.

We can then seek a left-stochastic policy that optimizes the ER level by solving

$$\begin{aligned} A^o \;&\triangleq\; \arg\min_{A\in\mathbb{A}}\; \mathrm{Tr}\left(\sum_{k=1}^{N} p_k^2 R_{s,k}\right) \\ &\text{subject to}\;\; Ap = p,\;\; \mathbb{1}^{\mathsf{T}}p = 1,\;\; p_k > 0 \end{aligned} \tag{12.77}$$

The solution to (12.77) can be obtained in a manner similar to the solution of the earlier problem (12.18). The only difference is that the parameters θ_k^2 should now be defined as follows:

$$\theta_k^2 \;\triangleq\; \mathrm{Tr}(R_{s,k}), \quad k = 1, 2, \ldots, N \tag{12.78}$$

in terms of the moment matrices $\{R_{s,k}\}$ alone — compare with (12.19). These parameters can then be used in (12.20) to construct the corresponding Hastings combination rule. The resulting (optimized) ER value will be

$$\mathrm{ER}^{o}_{\mathrm{dist,av}} \;=\; \frac{\mu h}{4}\left(\sum_{\ell=1}^{N}\frac{1}{\theta_\ell^2}\right)^{-1} \tag{12.79}$$

and it again holds that

$$\mathrm{ER}^{o}_{\mathrm{dist,av}} \;\leq\; \mathrm{ER}^{\mathrm{d.s.}}_{\mathrm{dist,av}} \;=\; \frac{1}{N}\mathrm{ER}_{\mathrm{ncop,av}} \tag{12.80}$$

so that, as expected, the ER of the distributed (consensus or distributed) network with an optimal left-stochastic matrix, A^o, is also superior to the ER of the non-cooperative scenario. More importantly, though, this optimal choice for A leads again to

$$\mathrm{ER}^{o}_{\mathrm{dist},k} \;\leq\; \mathrm{ER}_{\mathrm{ncop},k}, \quad k = 1, 2, \ldots, N \tag{12.81}$$

so that the individual agent performance in the optimized distributed network is improved across all agents relative to the non-cooperative case.

Example 12.5 (Comparing distributed and centralized learners)**.** We reconsider the numerical example at the end of Example 11.11, which deals with logistic networks operating on real data $\{\boldsymbol{\gamma}_k(i), \boldsymbol{h}_{k,i}\}$ originating from the alpha data set [223]. We consider the same network topology shown earlier in Figure 11.5 with $N = 20$ agents employing uniform step-sizes, $\mu_k \equiv \mu$. We already know from the result of Example 12.3 that the (optimal) Hastings rule reduces to the Metropolis rule (12.43), which is doubly-stochastic. Therefore, the entries of the corresponding Perron eigenvector are $p_k^o = 1/N$.

In this example, we compare the performance of two algorithms, ATC diffusion and the weighted centralized strategy, for the minimization of the (regularized) logistic risk function (11.205). The algorithms take the following form in this case:

$$\left\{\begin{array}{rcll} \boldsymbol{\psi}_{k,i} & = & (1-\rho\mu_k)\boldsymbol{w}_{k,i-1} + \mu\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}\left(\dfrac{1}{1+e^{\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}}\boldsymbol{w}_{k,i-1}}}\right) & \\ \boldsymbol{w}_{k,i} & = & \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} & \text{(ATC diffusion)} \end{array}\right. \tag{12.82}$$

and

$$\boldsymbol{w}_i = (1-\rho\mu)\boldsymbol{w}_{i-1} + \frac{\mu}{N}\sum_{k=1}^{N}\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}\left(\frac{1}{1+e^{\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}}\boldsymbol{w}_{i-1}}}\right) \quad \text{(weigh. centr.)} \tag{12.83}$$

In this case, and since the combination policy is doubly-stochastic, the ER performance of both algorithms will tend towards similar values. Using expression (12.79) with $h = 1$ for real data, this level is given by

$$\mathrm{ER}_{\mathrm{cent}} = \mathrm{ER}_{\mathrm{dist,av}}^{o} = \frac{\mu}{4}\left(\sum_{\ell=1}^{N}\frac{1}{\theta_\ell^2}\right)^{-1} = \frac{\mu}{4N}\mathrm{Tr}(R_s) \tag{12.84}$$

where we used (12.78) to note that

$$\theta_\ell^2 \equiv \theta^2 = \mathrm{Tr}(R_s) \tag{12.85}$$

Figure 12.5 plots the evolution of the ensemble-average learning curves, $\mathbb{E}\,\{J(\boldsymbol{w}_{i-1}) - J(w^o)\}$, for the above ATC diffusion and weighted centralized strategies using $\mu = 1 \times 10^{-4}$. The curves are obtained by averaging the trajectories $\{J(\boldsymbol{w}_{i-1}) - J(w^o)\}$ over 100 repeated experiments. The label on

the vertical axis in the figure refers to the learning curves by writing $\mathrm{ER}(i)$, with an iteration index i. Each experiment involves running the diffusion strategy (12.82) or the weighted centralized strategy (12.83) with $\rho = 10$. To generate the trajectories for the experiments in this example, the optimal w^o and the gradient noise covariance matrix, R_s, are first estimated off-line by applying a batch algorithm to all data points. For the data used in this experiment we have $\mathrm{Tr}(R_s) \approx 131.48$. It is observed in the figure that the learning curves tend towards the ER value predicted by the theoretical expression (12.84).

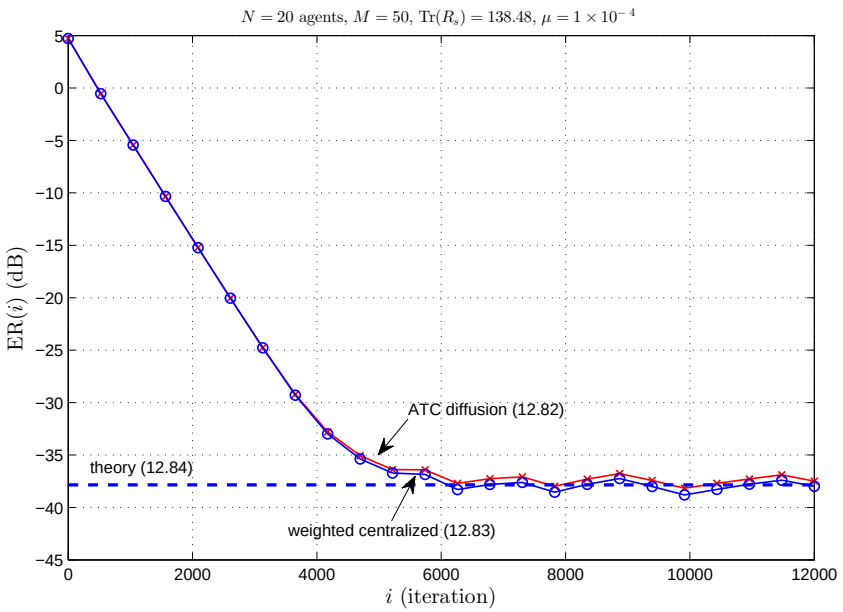

Figure 12.5: Evolution of the learning curves for the diffusion and weighted centralized strategies (12.82)–(12.83), with all agents employing the step-size $\mu = 1 \times 10^{-4}$.

■

13

Role of Informed Agents

We assumed in our presentation so far that all agents in the network have *continuous* access to data measurements and are able to evaluate their gradient vector approximations. However, it is observed in nature that the behavior of biological networks is often driven more heavily by a small fraction of informed agents as happens, for example, with bees and fish [12, 22, 125, 219]. This phenomenon motivates us to examine in this chapter multi-agent networks where only a *fraction* of the agents are informed, while the remaining agents are uninformed.

13.1 Informed and Uninformed Agents

Informed agents are defined as those agents that are capable of evaluating their gradient vector approximation continuously from streaming data and of performing the two tasks of adapting their iterates and consulting with their neighbors. Uninformed agents, on the other hand, are incapable of performing adaptation but can still participate in the consultation process with their neighbors. In this way, uninformed agents continue to assist in the diffusion of information across the network and act primarily as relay agents. We illustrate these two definitions

by considering a strongly-connected network running, for example, the ATC diffusion strategy (7.19). When an agent k is informed, it employs a strictly positive step-size and performs the two steps of adaptation and combination:

$$\text{(informed)} \qquad \left\{ \begin{array}{rcl} \boldsymbol{\psi}_{k,i} &=& \boldsymbol{w}_{k,i-1} - \dfrac{2\mu}{h}\widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{array} \right. \tag{13.1}$$

where $h = 1$ for real data and $h = 2$ for complex data. When an agent is uninformed, we set its step-size parameter to zero, $\mu_k = 0$, so that they are unable to perform the adaptation step but continue to perform the aggregation step. Their update equations therefore reduce to

$$\text{(uninformed)} \qquad \left\{ \begin{array}{rcl} \boldsymbol{\psi}_{k,i} &=& \boldsymbol{w}_{k,i-1} \\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{array} \right. \tag{13.2}$$

which collapse into the more compact form:

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1} \tag{13.3}$$

Although unnecessary for our treatment, we will assume for simplicity of presentation that the step-size parameter is uniform and equal to μ across all informed agents:

$$\mu_k = \left\{ \begin{array}{ll} \mu, & \text{(informed agent)} \\ 0, & \text{(uninformed agent)} \end{array} \right. \tag{13.4}$$

We will also focus on diffusion and consensus networks. Recall from (8.7)–(8.10) that the consensus and diffusion strategies correspond to the following choices for $\{A_o, A_1, A_2\}$ in terms of a single combination matrix A in the general description (8.46):

$$\text{consensus:} \qquad A_o = A, \;\; A_1 = I_N = A_2 \tag{13.5}$$

$$\text{CTA diffusion:} \qquad A_1 = A, \;\; A_2 = I_N = A_o \tag{13.6}$$

$$\text{ATC diffusion:} \qquad A_2 = A, \;\; A_1 = I_N = A_o \tag{13.7}$$

13.2 Conditions on Cost Functions

We recall the definition of the aggregate cost function for the case when all agents are informed:

$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} J_k(w) \tag{13.8}$$

Let $\mathcal{N}_I$ denote the set of indices of informed agents in the network:

$$\mathcal{N}_I \triangleq \{k : \text{such that } \mu_k = \mu > 0\} \tag{13.9}$$

The number of elements in $\mathcal{N}_I$ is denoted by

$$N_I = |\mathcal{N}_I| \tag{13.10}$$

The remaining agents are uninformed. We assume the network has at least one informed agent so that $N_I \geq 1$.

Now, observe from the definitions of informed and uninformed agents that if some agent k_o happens to be uninformed, then information about its gradient vector and, hence, cost function $J_{k_o}(w)$, is excluded from the overall learning process. For this reason, the effective global cost that the network will be minimizing is redefined as

$$J^{\text{glob,eff}}(w) \triangleq \sum_{k \in \mathcal{N}_I} J_k(w) \tag{13.11}$$

where the sum is over the individual costs of the informed agents. Clearly, if the individual costs share a common minimizer (which is the situation of most interest to us in this chapter), then the global minimizers of (13.8) and (13.11) will coincide. In general, though, the minimizers of these global costs may be different, and the minimizer of (13.11) will change with the set $\mathcal{N}_I$. For this reason, whenever necessary, we shall write $w^o(\mathcal{N}_I)$ to highlight the dependency of the minimizer of (13.11) on the set of informed agents.

In this chapter, whenever we refer to the global cost, we will be referring to the *effective* global cost (13.11) since entries from uninformed agents are excluded. It is this global cost, along with the individual costs of the informed agents, that we now need to assume to satisfy the conditions in Assumption 6.1. Specifically, the individual cost functions,

$J_k(w)$ for $k \in \mathcal{N}_I$, are each twice-differentiable and convex, with at least one of them being ν_d-strongly convex. Moreover, the effective aggregate cost function, $J^{\text{glob,eff}}(w)$, is also twice-differentiable and satisfies

$$0 < \frac{\nu_d}{h} I_{hM} \;\leq\; \nabla_w^2 J^{\text{glob,eff}}(w) \;\leq\; \frac{\delta_d}{h} I_{hM} \tag{13.12}$$

for some positive parameters $\nu_d \leq \delta_d$. In other words, conditions that we introduced in the earlier chapters on the cost functions $\{J^{\text{glob}}(w),\, J_k(w), k = 1, 2, \ldots, N\}$ will now need to be satisfied by the informed agents and by the effective global cost, $\{J^{\text{glob,eff}}(w),\, J_k(w),\, k \in \mathcal{N}_I\}$. For example, the smoothness condition (10.1) on the individual cost functions will now be required to be satisfied by the informed agents. Likewise, the gradient noise processes at the informed agents will need to satisfy the conditions in Assumption 8.1 or the fourth-order moment condition (8.121), as well as the smoothness condition (11.10) on their covariance matrices.

The limit point of the network will continue to be denoted by $w^\star$ and it is now defined as unique minimum of the following weighted aggregate cost function, $J^{\text{glob,eff},\star}(w)$, from (8.53), namely,

$$J^{\text{glob,eff}\star}(w) \;\triangleq\; \sum_{k \in \mathcal{N}_I} \mu_k p_k J_k(w) \tag{13.13}$$

where the sum is again defined over the set of informed agents, and where the $\{p_k\}$ are the entries of the Perron eigenvector of the primitive combination matrix A:

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0 \tag{13.14}$$

The limit vector, $w^\star$, that results from (13.13) is again dependent on the set of informed agents. For this reason, whenever necessary, we shall also write $w^\star(\mathcal{N}_I)$ to highlight the dependency of the minimizer of (13.13)on $\mathcal{N}_I$.

Under these adjustments, with requirements now imposed on the informed agents and with the network still assumed to be strongly-connected, it can be verified that the multi-agent network continues to be stable in the mean-square sense and in the mean sense, namely, for

all agents $k = 1, 2, \ldots, N$ (informed and uninformed alike):

$$\limsup_{i\to\infty} \|\mathbb{E}\,\widetilde{\boldsymbol{w}}_{k,i}\| = O(\mu) \tag{13.15}$$

$$\limsup_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 = O(\mu) \tag{13.16}$$

These facts are justified as follows. With regards to mean-square-error stability, we refer to the general proof in step (c) of Theorem 9.1. The two main differences that will occur if we repeat the argument relate to expressions (9.33) and (9.58), which now become

$$\boldsymbol{D}_{11,i-1} = \sum_{k\in\mathcal{N}_I} \mu p_k \boldsymbol{H}_{k,i-1}^{\mathsf{T}} \tag{13.17}$$

$$0 = \sum_{k\in\mathcal{N}_I} \mu p_k b_k^e \tag{13.18}$$

with the sums evaluated over the set of informed agents. It will continue to holds that $\boldsymbol{D}_{11,i-1} > 0$ in view of condition (13.12). Likewise, result (13.18) will hold in view of (13.13) from which we conclude that $w^\star$ now satisfies

$$\sum_{k\in\mathcal{N}_I} \mu p_k \nabla_w J_k(w^\star) = 0 \tag{13.19}$$

With regards to mean stability, if we refer to the proof of Theorem 9.3, we will again conclude that the matrix $\mathcal{B}$ remains stable since the matrix D_{11} defined by (9.195) will now become

$$D_{11} = \sum_{k\in\mathcal{N}_I} \mu p_k H_k^{\mathsf{T}} \tag{13.20}$$

and it remains positive-definite.

13.3 Mean-Square-Error Performance

The results in the sequel reveal some interesting facts about adaptation and learning in the presence of informed and uninformed agents [213, 247, 250]. For example, it will be seen that when the set of informed agents is enlarged, the convergence rate of the network will become faster albeit at the expense of possible deterioration in mean-square-error performance. In other words, the MSD and ER performance metrics do not necessarily improve with a larger proportion of

informed agents. The arguments in this chapter extend the presentation from [213] to the case of complex-valued arguments.

Thus, consider strongly-connected networks running the consensus or diffusion strategies (7.9), (7.18), or (7.19). We recall from expression (11.118) that, when all agents are informed, the MSD performance of these distributed solutions is given by:

$$\mathrm{MSD}_{\mathrm{dist,av}} \;=\; \frac{\mu}{2h}\,\mathrm{Tr}\left[\left(\sum_{k=1}^{N} p_k H_k\right)^{-1}\left(\sum_{k=1}^{N} p_k^2 G_k\right)\right] \tag{13.21}$$

We also recall from (11.139) that the convergence rate of the error variances, $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, towards this MSD value is given by

$$\alpha_{\mathrm{dist}} \;=\; 1 - 2\mu\,\lambda_{\min}\left\{\sum_{k=1}^{N} p_k H_k\right\} \;+\; o(\mu) \tag{13.22}$$

in terms of the smallest eigenvalue of the sum of weighted Hessian matrices. In the above expression, the parameter $\alpha_{\mathrm{dist}} \in (0,1)$ and the smaller the value of α_{dist} is, the faster the convergence behavior becomes.

If we now consider the case where some agents are uninformed, and repeat the derivation that led to (11.47) and (11.118), we will find that the same result still hold if we set $\mu_k = 0$ for the uninformed agents [68, 213, 247, 250], namely,

$$\alpha_{\mathrm{dist}} \;=\; 1 - 2\mu\,\lambda_{\min}\left\{\sum_{k\in\mathcal{N}_I} p_k H_k\right\} + o(\mu) \tag{13.23}$$

and

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} \;=\; \frac{\mu}{2h}\,\mathrm{Tr}\left[\left(\sum_{k\in\mathcal{N}_I} p_k H_k\right)^{-1}\left(\sum_{k\in\mathcal{N}_I} p_k^2 G_k\right)\right] \tag{13.24}$$

where the sums are over the set $k \in \mathcal{N}_I$.

Observe now that since the entries of p are positive for primitive left-stochastic matrices A, it is clear from (13.23) that, for small step-sizes, if the set of informed agents is enlarged from $\mathcal{N}_I$ to

$$\mathcal{N}_I^{'} \supset \mathcal{N}_I \tag{13.25}$$

then the convergence rate improves (i.e., faster convergence with α_{dist} becoming smaller). However, from (13.24), the network MSD may decrease, remain unchanged, or increase depending on the values of $\{H_k, G_k\}$. This situation is illustrated in Figure 13.1.

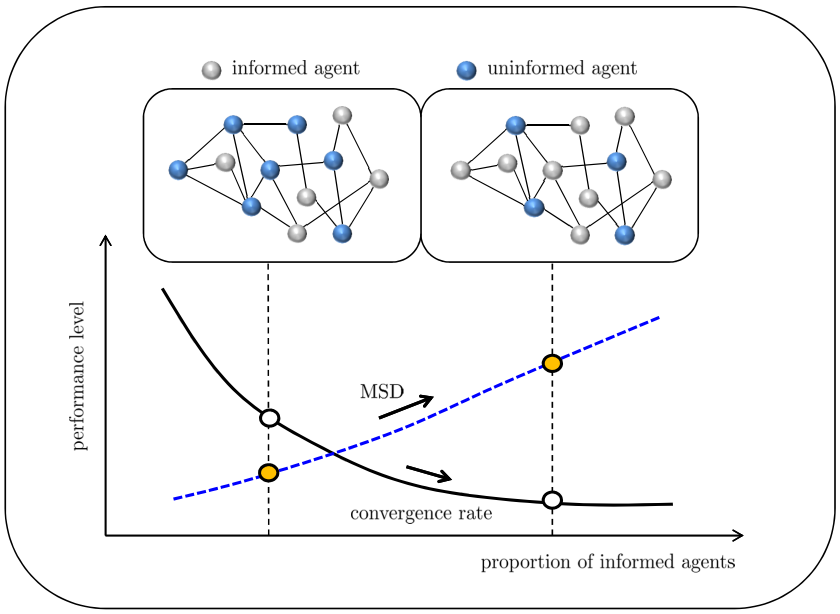

Figure 13.1: Enlarging the set of informed agents improves the convergence rate but does not necessarily improve the MSD network performance.

Note that the previous statements compare the convergence rates and MSD levels relative to the minimizers $w^\star(\mathcal{N}_I)$ and $w^\star(\mathcal{N}_I^{'})$ of the weighted effective costs (13.13) that would correspond to the sets $\mathcal{N}_I$ and $\mathcal{N}_I^{'}$. These minimizers are generally different and, therefore,

these comparisons amount to determining how well and how fast the network configuration, $\mathcal{N}_I$ or $\mathcal{N}_I^{'}$, converge towards their respective limit points. The next example describes the useful scenario when the two minimizers, $w^\star(\mathcal{N}_I)$ and $w^\star(\mathcal{N}_I^{'})$, coincide since the corresponding individual costs will share a common minimizer.

Example 13.1 (Role of informed agents over MSE networks). For the MSE network of Example 6.3 with uniform step-sizes and uniform covariance matrices, i.e., $\mu_k \equiv \mu$ and $R_{u,k} \equiv R_u > 0$, we have

$$H_k = \begin{bmatrix} R_u & 0 \\ 0 & R_u^{\mathsf{T}} \end{bmatrix} \equiv H, \quad G_k = \sigma_{v,k}^2 \begin{bmatrix} R_u & \times \\ \times & R_u^{\mathsf{T}} \end{bmatrix} \tag{13.26}$$

Moreover, all costs $J_k(w)$ share the same minimizer so that $w^\star = w^o$ for any set of informed agents. Using $h = 2$ for complex data, it follows that expressions (13.23) and (13.24) reduce to

$$\alpha_{\text{dist}} \approx 1 - 2\mu\,\lambda_{\min}(R_u)\left(\sum_{k\in\mathcal{N}_I} p_k\right) \tag{13.27}$$

$$\text{MSD}_{\text{dist,av}} = \frac{\mu M}{h}\left(\sum_{k\in\mathcal{N}_I} p_k\right)^{-1}\left(\sum_{k\in\mathcal{N}_I} p_k^2\sigma_{v,k}^2\right) \tag{13.28}$$

where the symbol $\approx$ in the expression for α_{dist} signifies that we are ignoring the higher-order term $o(\mu)$ for sufficiently small step-sizes. It is now clear that if the set of informed agents is enlarged to $\mathcal{N}_I^{'} \supset \mathcal{N}_I$, then the convergence rate improves (i.e., faster convergence with α_{dist} becoming smaller). However, from (13.28), the network MSD may decrease, remain unchanged, or increase depending on the values of the noise variances $\{\sigma_{v,k}^2\}$ at the new informed agents. We illustrate this behavior by considering two cases of interest.

Assume first that A is doubly-stochastic. Then, $p_k = 1/N$ and the above expressions reduce to:

$$\alpha_{\text{dist}} \approx 1 - 2\mu\left(\frac{N_I}{N}\right)\lambda_{\min}(R_u) \tag{13.29}$$

$$\text{MSD}_{\text{dist,av}} = \frac{\mu M}{h}\,\frac{1}{N}\left(\frac{1}{N_I}\sum_{k\in\mathcal{N}_I}\sigma_{v,k}^2\right) \tag{13.30}$$

It is seen that if we add a new informed agent of index $k' \notin \mathcal{N}_I$, then the convergence rate improves because N_I increases but the MSD performance of

the network will get worse if

$$\left(\frac{1}{N_I+1}\sum_{k\in\mathcal{N}_{I+1}}\sigma_{v,k}^2\right) \;>\; \left(\frac{1}{N_I}\sum_{k\in\mathcal{N}_I}\sigma_{v,k}^2\right) \tag{13.31}$$

where $\mathcal{N}_{I+1}=\mathcal{N}_I\cup\{k'\}$ or, equivalently, if

$$\sigma_{v,k'}^2 \;>\; \frac{1}{N_I}\sum_{k\in\mathcal{N}_I}\sigma_{v,k}^2 \tag{13.32}$$

That is, the MSD performance gets worse if the incoming noise power at the newly added agent is worse than the average noise power of the existing informed agents.

Let us consider next the case in which the combination weights $\{a_{\ell k}\}$ are selected according to the averaging rule (which is left-stochastic):

$$a_{\ell k} \;=\; \begin{cases} 1/n_k, & \ell\in\mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{13.33}$$

in terms of the degrees of the various agents. Recall that n_k is equal to the number of neighbors that agent k has. It can be verified that the Perron eigenvector p is given by:

$$p \;=\; \left(\sum_{k=1}^{N} n_k\right)^{-1}\begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{bmatrix} \tag{13.34}$$

In this case, expressions (13.27) and (13.28) reduce to

$$\alpha_{\text{dist}} \;\approx\; 1-2\mu\,\lambda_{\min}(R_u)\left(\frac{\sum_{k\in\mathcal{N}_I} n_k}{\sum_{k=1}^{N} n_k}\right) \tag{13.35}$$

$$\text{MSD}_{\text{dist,av}} \;=\; \frac{\mu M}{h}\left(\frac{1}{\sum_{k=1}^{N} n_k}\right)\left(\frac{1}{\sum_{k\in\mathcal{N}_I} n_k}\right)\left(\sum_{k\in\mathcal{N}_I} n_k^2\sigma_{v,k}^2\right) \tag{13.36}$$

It is again seen that if we add a new informed agent $k'\notin\mathcal{N}_I$, then the convergence rate improves. However, the MSD performance of the network will get worse if

$$\left(\frac{1}{\sum_{k\in\mathcal{N}_{I+1}} n_k}\right)\left(\sum_{k\in\mathcal{N}_{I+1}} n_k^2\sigma_{v,k}^2\right) \;>\; \left(\frac{1}{\sum_{k\in\mathcal{N}_I} n_k}\right)\left(\sum_{k\in\mathcal{N}_I} n_k^2\sigma_{v,k}^2\right) \tag{13.37}$$

or, equivalently, if

$$n_{k'}\,\sigma^2_{v,k'} \;>\; \left(\sum_{k\in\mathcal{N}_I} n_k\right)^{-1} \left(\sum_{k\in\mathcal{N}_I} n_k^2\sigma^2_{v,k}\right) \tag{13.38}$$

where the degrees of the agents are now involved in the inequality *in addition* to the noise variances. The above condition can be expressed in terms of a *weighted* harmonic mean as follows. Introduce the inverse variables

$$x_k \;\triangleq\; \frac{1}{n_k\sigma^2_{v,k}},\; k\in\mathcal{N}_I \tag{13.39}$$

which consist of the inverses of the noise variances scaled by n_k. Let x_H denote the weighted harmonic mean of these variables, with weights $\{n_k\}$, which is defined as

$$x_H \;\triangleq\; \left(\sum_{k\in\mathcal{N}_I} n_k\right)\left(\sum_{k\in\mathcal{N}_I} \frac{n_k}{x_k}\right)^{-1} \tag{13.40}$$

Then, condition (13.38) is equivalent to stating that

$$x'_k \;\triangleq\; \frac{1}{n_{k'}\,\sigma^2_{v,k'}} \;<\; x_H \tag{13.41}$$

That is, the MSD performance will get worse if the new inverse variable, x'_k, is smaller than the weighted harmonic mean of the inverse variables $\{x_k\}$ associated with the existing informed agents.

We illustrate these results numerically for the case of the averaging rule (13.33) with uniform step-sizes across the agents set at $\mu_k \equiv \mu = 0.002$. Figure 13.2 shows two versions of the connected network topology with $N = 20$ agents used in the simulations. In one version, the topology has 14 informed agents and 6 uninformed agents. In the second version, two of the previously uninformed agents are transformed back to the informed state so that the topology now ends up with 16 informed agents. The measurement noise variances, $\{\sigma^2_{v,k}\}$, and the power of the regression data, assumed uniform and of the form $R_{u,k} = \sigma^2_u I_M$, are shown in the right and left plots of Figure 13.3, respectively.

Figure 13.4 plots the evolution of the ensemble-average learning curves, $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, for the ATC diffusion strategy (13.1)–(13.2). The curves are obtained by averaging the trajectories $\{\frac{1}{N}\|\widetilde{\boldsymbol{w}}_i\|^2\}$ over 200 repeated experiments. The label on the vertical axis in the figure refers to the learning curve $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ by writing $\mathrm{MSD}_{\mathrm{dist,av}}(i)$, with an iteration index i. Each experiment involves running the ATC diffusion strategy (13.1)–(13.2) with $h = 2$ on complex-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ generated according to the model

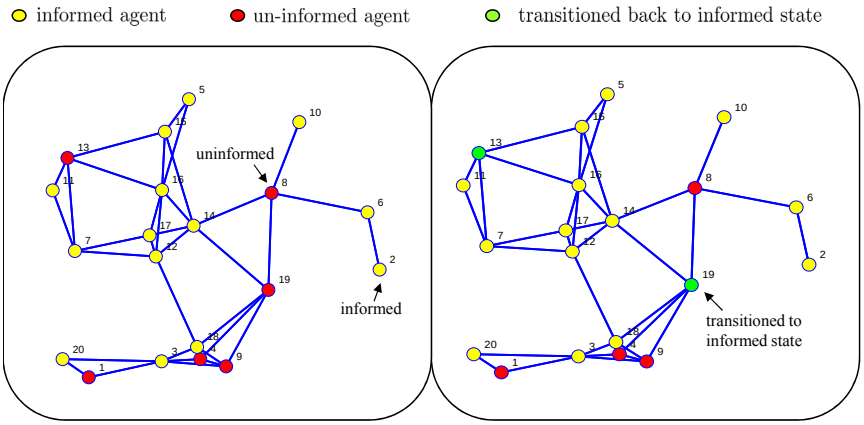

Figure 13.2: A connected network topology consisting of $N = 20$ agents employing the averaging rule (13.33). Two simulations are performed in this example. In one simulation, the topology on the left is used with 14 informed agents and 6 uninformed agents. In a second simulation, the topology on the right is used where two of the previously uninformed agents are transformed back to the informed state.

$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i)$, with $M = 10$. The unknown vector w^o is generated randomly and its norm is normalized to one. The solid horizontal lines in the figure represent the theoretical MSD values obtained from (13.36) for the two scenarios shown in Figure 13.2, namely,

$$\text{MSD}(\mathcal{N}_I) \approx -50.19\,\text{dB}, \qquad \text{MSD}(\mathcal{N}_I^{'}) \approx -49.40\,\text{dB} \tag{13.42}$$

where $\mathcal{N}_I^{'}$ denotes the enlarged set of informed agents shown on the right-hand side of Figure 13.2. It is observed in this simulation that when the set of informed agents is enlarged by adding agents #13 and #19, the convergence rate is improved while the MSD value is degraded by about 0.79dB.

■

Example 13.2 (Performance degradation under fixed convergence rate)**.** We continue with Example 13.1 and the case of the averaging rule (13.33). The current example is based on the discussion from [250] and its purpose is to show that even if we adjust the convergence rate of the network to remain fixed and invariant to the proportion of informed agents, the MSD performance of the network can still deteriorate if the set of informed agents is enlarged. To

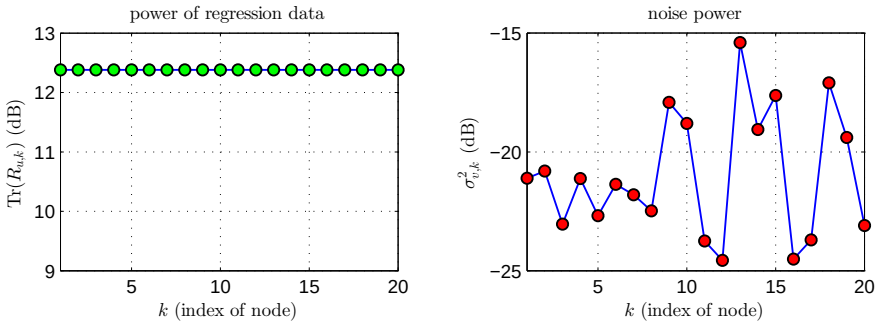

Figure 13.3: Measurement noise profile (right) and regression data power (left) across all agents in the network. The covariance matrices are assumed to be of the form $R_{u,k} = \sigma_u^2 I_M$, and the noise and regression data are Gaussian distributed in this simulation.

see this, we set the step-size to the following normalized value:

$$\mu = \mu_o \left(\sum_{k \in \mathcal{N}_I} n_k \right)^{-1} \tag{13.43}$$

for some small $\mu_o > 0$, and where the normalization is over the sum of the degrees of the informed agents. Note that this selection of μ depends on $\mathcal{N}_I$. For this choice of μ, the convergence rate given by (13.35) becomes

$$\alpha_{\text{dist}} \;\approx\; 1 - 2\mu_o \, \lambda_{\min}(R_u) \left(\sum_{k=1}^{N} n_k \right)^{-1} \tag{13.44}$$

which is independent of N_I. Therefore, no matter how the set $\mathcal{N}_I$ is adjusted, the convergence rate of the network remains fixed. At the same time, the MSD level (13.36) becomes

$$\text{MSD}_{\text{dist,av}} = \frac{\mu_o M}{2} \left(\frac{1}{\sum_{k=1}^{N} n_k} \right) \left(\frac{1}{\sum_{k \in \mathcal{N}_I} n_k} \right)^2 \left(\sum_{k \in \mathcal{N}_I} n_k^2 \sigma_{v,k}^2 \right) \tag{13.45}$$

Some straightforward algebra will show that if we add a new informed agent $k' \notin \mathcal{N}_I$, then the MSD performance of the network will get worse if the parameters $\{n'_k, \sigma^2_{v,k'}\}$ satisfy the inequality:

$$n_{k'} \;>\; 2 \left(\sum_{k \in \mathcal{N}_I} n_k \right) \left[\frac{\left(\sum_{k \in \mathcal{N}_I} n_k \right)^2 \sigma^2_{v,k'}}{\sum_{k \in \mathcal{N}_I} n_k^2 \sigma^2_{v,k}} - 1 \right]^{-1} \tag{13.46}$$

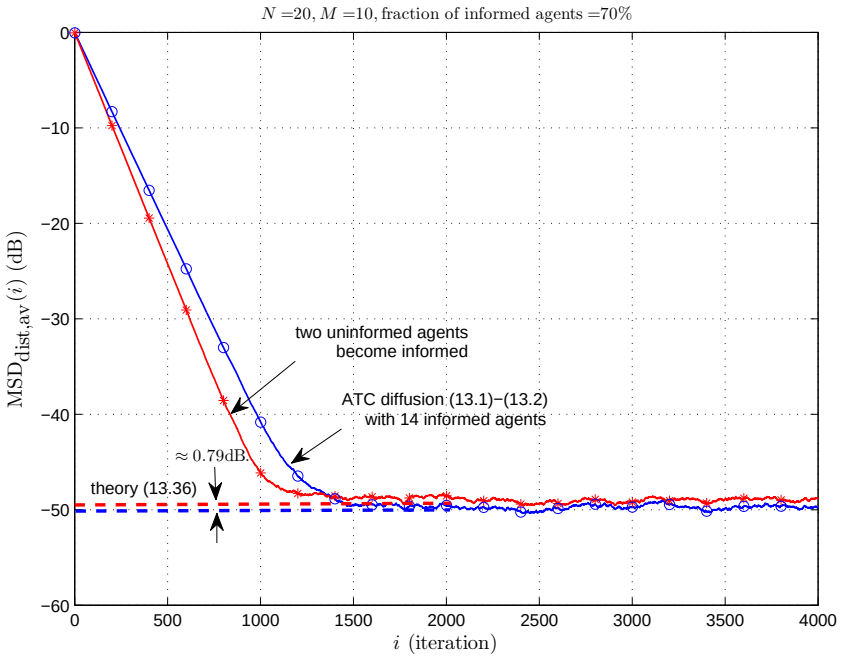

Figure 13.4: Evolution of the learning curves for the ATC diffusion strategy (13.1)–(13.2) using $\mu = 0.002$ and the averaging rule (13.33).

We now verify that there exist situations under which the above requirement is satisfied so that the network MSD will end up increasing (an undesirable effect) even though the convergence rate has been set to a constant value.

Consider first the case in which all agents have the same degree, say, $n_k \equiv n$ for all k. Then, condition (13.46) becomes

$$\sigma_{v,k'}^2 \;>\; \left(2 + \frac{1}{N_I}\right)\left(\frac{1}{N_I}\sum_{k\in\mathcal{N}_I}\sigma_{v,k}^2\right) \tag{13.47}$$

That is, if the new added noise variance is sufficiently larger than the average noise variance at the informed agents, then deterioration in performance will occur.

Our second example assumes the noise variances are uniform across all agents, say, $\sigma_{v,k}^2 \equiv \sigma_v^2$ for all k. Then, condition (13.46) becomes

$$n_k' \;>\; 2\left(\sum_{k\in\mathcal{N}_I} n_k\right)\left[\frac{\left(\sum_{k\in\mathcal{N}_I} n_k\right)^2}{\left(\sum_{k\in\mathcal{N}_I} n_k^2\right)} - 1\right]^{-1} \tag{13.48}$$

so that if the degree of the new added agent is sufficiently large, then deterioration in performance will occur. The results in these two cases suggest that it is beneficial to keep few highly noisy or highly connected agents uninformed and for them to participate only in the aggregation task (13.2) and to act as relays.

■

13.4 Controlling Degradation in Performance

The previous arguments indicate that the MSD performance need not improve with the addition of informed agents. The deterioration in network performance can be controlled through proper selection of the combination weights, for example, when the matrix A is selected according to the Hastings rule (12.20). Recall that, under the condition of uniform step-sizes and uniform Hessian matrices, and assuming *all* agents are informed, i.e.,

$$\mu_k \equiv \mu > 0, \quad H_k \equiv H, \quad k = 1, 2, \ldots, N \tag{13.49}$$

we derived earlier in (12.21) the following expression for the entries of the optimized Perron eigenvector:

$$p_k^o = \frac{1}{\theta_k^2} \left(\sum_{\ell=1}^{N} \frac{1}{\theta_\ell^2} \right)^{-1}, \quad k = 1, 2, \ldots, N \tag{13.50}$$

Now, assume the gradient noise factors, $\{\theta_k^2\}$, that result from assuming all agents are informed are known. Assume further that the partially informed network under study in this chapter (with both informed and uninformed agents) employs the Hastings rule (12.20) that would result from using the above Perron vector entries. Substituting these entries into (13.23) and (13.24) we find that the convergence rate and the MSD level of the partially informed network are now given by

$$\alpha_{\text{dist}} \approx 1 - 2\mu\, \lambda_{\min}(H) \left(\sum_{k \in \mathcal{N}_I} \frac{1}{\theta_k^2} \right) \left(\sum_{k=1}^{N} \frac{1}{\theta_k^2} \right)^{-1} \tag{13.51}$$

$$\text{MSD}_{\text{dist,av}} = \frac{\mu}{2h} \left(\sum_{k=1}^{N} \frac{1}{\theta_k^2} \right)^{-1} \tag{13.52}$$

We observe that when the agents employ the Hastings rule, the network MSD level becomes independent of N_I (and, hence, does not change with the addition of informed agents), while the convergence rate decreases (becomes faster) as the set of informed agents is enlarged (since the expression for α_{dist} depends on N_I).

13.5 Excess-Risk Performance

We can repeat the analysis of the previous sections and examine how the excess-risk (ER) performance of distributed solutions varies as a function of the fraction of informed agents in the network. The treatment is similar and so we shall be brief. In a manner similar to the study of the MSD metric, the ER performance of distributed solutions with N_I informed agents can be deduced from (11.186) and is given by:

$$\text{ER}_{\text{dist},k} \;=\; \text{ER}_{\text{dist,av}} \;=\; \frac{\mu h}{4}\left(\sum_{k\in\mathcal{N}_I} p_k\right)^{-1} \text{Tr}\left(\sum_{k\in\mathcal{N}_I} p_k^2 R_{s,k}\right) \tag{13.53}$$

where the sum of the $\{p_k\}$ does not evaluate to one anymore because this sum runs over $k \in \mathcal{N}_I$ only and not over the entire set of agents. It is again seen from (13.53) that the ER level of the network may increase, remain unchanged, or decrease with the addition of informed agents.

Example 13.3 (Role of informed agents in online learning)**.** We revisit Example 11.9, which deals with a collection of N learners. Using $h = 1$ for real data, the ER performance level for the distributed solution, using N_I informed agents with step-size $\mu_k \equiv \mu$, can be deduced from (13.53) as

$$\text{ER}_{\text{dist,av}} = \frac{\mu}{4}\left(\sum_{k\in\mathcal{N}_I} p_k\right)^{-1}\left(\sum_{k\in\mathcal{N}_I} p_k^2\right)\text{Tr}\,(R_s) \tag{13.54}$$

In particular, it is seen that if we add a new informed agent of index $k' \notin \mathcal{N}_I$, then the ER performance levels will get worse if

$$p_{k'} \;>\; \left(\sum_{k\in\mathcal{N}_I} p_k\right)^{-1}\left(\sum_{k\in\mathcal{N}_I} p_k^2\right) \tag{13.55}$$

This condition is in terms of the entries $\{p_k\}$, which are determined by the combination policy, A. We again consider two choices for the combination matrices.

Assume first that A is doubly-stochastic (such as the Metropolis rule (12.43)) so that $p_k = 1/N$. Then, condition (13.55) cannot be satisfied and we conclude that, for this case, the addition of informed agents cannot degrade network performance. Indeed, in this scenario, it can be readily seen that the ER expression (13.54) reduces to

$$\mathrm{ER}_{\mathrm{dist,av}} \;=\; \frac{\mu}{4}\left(\frac{1}{N}\right)\mathrm{Tr}\,(R_s) \tag{13.56}$$

Both of these expressions are independent of N_I; it is worth noting that in the current problem, the Hastings rule (12.20) reduces to the doubly-stochastic Metropolis rule (12.43), which explains why the ER result (13.56) is independent of N_I.

Let us consider next the case in which the combination weights $\{a_{\ell k}\}$ are selected according to the averaging rule (13.33). Using (13.34), condition (13.55) would then indicate that the network ER level will degrade if the degree of the newly added informed agent satisfies:

$$n_{k'} \;>\; \left(\sum_{k\in\mathcal{N}_I} n_k\right)^{-1}\left(\sum_{k\in\mathcal{N}_I} n_k^2\right) \tag{13.57}$$

■

14

Combination Policies

We end our exposition by commenting on the selection of the combination policy, A. Although unnecessary, we assume in this chapter that all agents are informed so that their step-sizes are strictly positive. It is clear from the performance expression (11.118) that the combination weights $\{a_{\ell k}\}$ that are used by the consensus (7.9) and diffusion strategies (7.18) and (7.19) influence the performance of the distributed solution in a direct manner. Their influence is reflected by the entries $\{p_k\}$, defined earlier through (11.136), namely,

$$\mathrm{MSD}_{\mathrm{dist},k} = \mathrm{MSD}_{\mathrm{dist,av}} = \frac{1}{2h}\mathrm{Tr}\left[\left(\sum_{k=1}^{N}\mu_k p_k H_k\right)^{-1}\left(\sum_{k=1}^{N}\mu_k^2 p_k^2 G_k\right)\right] \quad (14.1)$$

There are several ways by which the coefficients $\{a_{\ell k}\}$ can be selected. On one hand, many existing combination policies rely on static selections for these coefficients, i.e., selections that are fixed during the adaptation and learning process and do not change with time. On the other hand, the discussion will reveal that it is important to consider selections where these coefficients are also adapted over time, and are allowed to evolve dynamically alongside the learning mechanism. This

latter area of investigation is evolving steadily and there are already some useful adaptive combination policies proposed in the literature. We comment on some of them in a future section.

14.1 Static Combination Policies

To begin with, Table 14.1 is extracted from [208] and lists some common static choices for selecting the combination weights $\{a_{\ell k}\}$ for a network with N agents. In the table, the symbol $n_k = |\mathcal{N}_k|$ denotes the degree of agent k, which is equal to the size of its neighborhood, and the symbol $n_{\max}$ denotes the maximum degree across the network:

$$n_{\max} \stackrel{\Delta}{=} \max_{1 \le k \le N} n_k \tag{14.2}$$

The Laplacian rule, which appears in the second line of the table, relies on the use of the Laplacian matrix of the network and a positive scalar, β. The Laplacian matrix is a symmetric matrix whose entries are constructed as follows [41, 82, 143, 208]:

$$[\mathcal{L}]_{\ell k} = \begin{cases} n_\ell - 1, & \text{if } k = \ell \\ -1, & \text{if } k \neq \ell \text{ and } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{14.3}$$

The Laplacian matrix has several useful properties and conveys important information about the network topology [208, App. B]. For example, (a) $\mathcal{L}$ is always nonnegative-definite; (b) the entries on each of its rows add up to zero; and (c) its smallest eigenvalue is zero. Moreover, (d) the multiplicity of zero as an eigenvalue for $\mathcal{L}$ is equal to the number of connected subgraphs of the network topology. Accordingly, a graph is connected if, and only if, the second smallest eigenvalue of $\mathcal{L}$ (also called the algebraic connectivity of the graph) is nonzero.

It is observed from the constructions in Table 14.1 that the values of the combination weights $\{a_{\ell k}\}$ are solely determined by the degrees (and, hence, the extent of connectivity) of the agents. As explained in [208], while such selections may be appropriate in some applications, they can nevertheless lead to degraded performance in the context of adaptation and learning over networks [232]. This is because

these weighting schemes ignore the gradient noise profile across the network.

Table 14.1: Static selections for the combination matrix $A = [a_{\ell k}]$. The second column indicates whether the resulting matrix is left-stochastic or doubly stochastic.

Entries of combination matrix A	**Type of** A
1. Averaging rule [39]: $a_{\ell k} = \begin{cases} 1/n_k, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}$	left-stochastic
2. Laplacian rule [215, 265]: $a_{\ell k} = 1 - \beta[\mathcal{L}]_{\ell k},\ \beta > 0$	symmetric and doubly-stochastic
3. Laplacian rule using $\beta = 1/n_{\max}$: $a_{\ell k} = \begin{cases} 1/n_{\max}, & \text{if } k \neq \ell \text{ are neighbors} \\ 1 - \frac{(n_k - 1)}{n_{\max}}, & k = \ell \\ 0, & \text{otherwise} \end{cases}$	symmetric and doubly-stochastic
4. Laplacian rule using $\beta = 1/N$ (or maximum-degree rule [266]) : $a_{\ell k} = \begin{cases} 1/N, & \text{if } k \neq \ell \text{ are neighbors} \\ 1 - (n_k - 1)/N, & k = \ell \\ 0, & \text{otherwise} \end{cases}$	symmetric and doubly-stochastic
5. Metropolis rule [106, 167, 265]: $a_{\ell k} = \begin{cases} \frac{1}{\max\{n_k, n_\ell\}}, & \text{if } k \neq \ell \text{ are neighbors} \\ 1 - \sum_{m \in \mathcal{N}_k \backslash \{k\}} a_{mk}, & k = \ell \\ 0, & \text{otherwise} \end{cases}$	symmetric and doubly-stochastic
6. Relative-degree rule [58]: $a_{\ell k} = \begin{cases} n_\ell \left(\sum_{m \in \mathcal{N}_k} n_m \right)^{-1}, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}$	left-stochastic

14.2 Need for Adaptive Policies

One way to capture the gradient noise profile across the network is by means of the factors $\{\theta_k^2\}$ defined earlier in (12.19) and (12.78):

$$\theta_k^2 \triangleq \begin{cases} \mathrm{Tr}(H^{-1}G_k) & \text{(for MSD performance)} \\ \mathrm{Tr}(R_{s,k}) & \text{(for ER performance)} \end{cases} \tag{14.4}$$

where G_k is also dependent on the gradient noise variance, $R_{s,k}$, in view of definition (11.12). Now, since some agents can be noisier (with larger θ_k^2) than others, it becomes important to take into account the amount of noise that is present at the agents and to assign more or less weights to interactions with neighbors in accordance to their noise level. For example, if some agent k can determine which of its neighbors are the noisiest, then it can assign smaller combination weights to its interaction with these neighbors. One difficulty in employing this strategy is that the noise factors $\{\theta_\ell^2\}$ are unknown beforehand since their values depend on the unknown noise moments $\{G_\ell, R_{s,\ell}\}$. It therefore becomes necessary to devise noise-aware schemes that enable agents to estimate the noise factors $\{\theta_\ell^2\}$ of their neighbors in order to assist them in the process of selecting proper combination coefficients. It is also desirable for these schemes to be adaptive so that they can track variations in the noise moments over time. The techniques described in this chapter are motivated by the procedures developed in [208, 244, 280]; variations appear in [95, 270]. We first consider an example to illustrate the idea.

Example 14.1 (Noise variance estimation over MSE networks)**.** We continue with the MSE network from Example 12.1 where we assumed uniform step-sizes and uniform regression covariance matrices, i.e., $\mu_k \equiv \mu$ and $R_{u,k} \equiv R_u > 0$ for $k = 1, 2, \ldots, N$. Recall that for these networks, the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ are assumed to be related via the linear regression model:

$$\boldsymbol{d}_k(i) \;=\; \boldsymbol{u}_{k,i} w^o \;+\; \boldsymbol{v}_k(i), \quad k = 1, 2, \ldots, N \tag{14.5}$$

where the variance of the noise is denoted by $\sigma_{v,k}^2 = \mathbb{E}\,|\boldsymbol{v}_k(i)|^2$. We derived in Example 12.2 the (optimal) combination coefficients in the form of the

Hastings rule (12.39), namely,

$$a_{\ell k}^{o} \;=\; \begin{cases} \dfrac{\sigma_{v,k}^2}{\max\{\, n_k\sigma_{v,k}^2,\; n_\ell\sigma_{v,\ell}^2 \,\}}, & \ell \in \mathcal{N}_k\backslash\{k\} \\ 1 - \displaystyle\sum_{m\in\mathcal{N}_k\backslash\{k\}} a_{mk}^{o}, & \ell = k \end{cases} \tag{14.6}$$

and noted that the gradient noise factors in this case are given by $\theta_k^2 = 2M\sigma_{v,k}^2$; they are therefore proportional to the measurement noise power, $\sigma_{v,k}^2$. It is clear that rule (14.6) takes into account the size of the noise powers, $\{\sigma_{v,\ell}^2\}$, at the agents. Moreover, in this particular construction, only the noise levels of the two interacting agents are directly involved in the computation of their combination weights; no other agents from the neighborhood of agent k are involved in the calculation.

A second combination construction is motivated in [280] for MSE networks by solving an alternative optimization problem than the one that led to the Hastings rule (12.39) or (14.6). We shall describe this alternative construction further ahead in (14.27). For now, we simply state that the resulting combination rule for the case under study in this example, and which we shall refer to as the relative-variance rule [206], takes the following form:

$$a_{\ell k}^{o} \;=\; \begin{cases} \dfrac{1}{\sigma_{v,\ell}^2}\left(\displaystyle\sum_{m\in\mathcal{N}_k}\frac{1}{\sigma_{v,m}^2}\right)^{-1}, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{14.7}$$

Comparing with (14.6), we note that in this second rule, the interaction between agents k and ℓ is more broadly dependent on the noise profile across the *entire* neighborhood of agent k. In particular, neighbors with smaller noise power relative to the neighborhood are assigned larger weights.

For every agent k, both rules (14.6) and (14.7) still require knowledge of the noise variances $\{\sigma_{v,\ell}^2\}$. This information is generally unavailable but can be estimated by agent k as follows — see the derivation that leads to (14.53) in the next section. Assume, for illustration purposes, that the agents are running the ATC LMS diffusion strategy (7.23):

$$\begin{cases} \boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{k,i-1} + \mu\boldsymbol{u}_{k,i}^{*}\,[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{cases} \tag{14.8}$$

Then, agent k can estimate the noise variance, $\sigma_{v,\ell}^2$, by running the recursion:

$$\boldsymbol{\gamma}_{\ell k}^2(i) \;=\; (1-\zeta)\,\boldsymbol{\gamma}_{\ell k}^2(i-1) \;+\; \zeta\,\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2, \qquad \ell \in \mathcal{N}_k \tag{14.9}$$

where $0 < \zeta \ll 1$ is a small positive coefficient, e.g., $\zeta = 0.1$. This recursion relies on smoothing the energy of the difference between the intermediate iterate, $\boldsymbol{\psi}_{\ell,i}$, received from neighbor ℓ and the existing iterate $\boldsymbol{w}_{k,i-1}$ at agent k. The resulting energy measure provides an indication of the amount of noise that is present at agent ℓ since it can be verified that asymptotically [208] — see also (14.55):

$$\mathbb{E}\,\boldsymbol{\gamma}^2_{\ell k}(i) \;\approx\; \mu^2\sigma^2_{v,\ell}\mathrm{Tr}(R_u), \quad i \gg 1 \tag{14.10}$$

with the limit being proportional to $\sigma^2_{v,\ell}$. Therefore, the running variables $\{\boldsymbol{\gamma}^2_{\ell k}(i)\}$ can be used by agent k as scaled estimates for the noise variances. These variables can then be used in place of the noise variances in rules (14.6) and (14.7) to adapt the combination weights over time. Under this construction, each agent k ends up running n_k recursions of the form (14.9), one for each of its neighbors, in order to update the necessary variables $\{\boldsymbol{\gamma}^2_{\ell k}(i),\ \ell \in \mathcal{N}_k\}$.

■

14.3 Hastings Policy

Before discussing adaptive constructions for the combination weights, we present two combination policies that are noise-aware. We already encountered one such policy when we derived the Hastings rule earlier in Sec. 12.2 — see expression (12.20). Here we review it briefly before discussing the second policy, known as the relative variance rule. Recall that the Hastings rule was derived under the condition of uniform step-sizes and uniform Hessian matrices, namely,

$$\mu_k \equiv \mu, \quad H_k \equiv H, \quad k = 1, 2, \ldots, N \tag{14.11}$$

The rule followed from the solution to the optimization problem (12.18) and led to

$$a^o_{\ell k} \;=\; \begin{cases} \dfrac{\theta^2_k}{\max\{\, n_k\theta^2_k,\ n_\ell\theta^2_\ell \,\}}, & \ell \in \mathcal{N}_k\backslash\{k\} \\ 1 - \displaystyle\sum_{m\in\mathcal{N}_k\backslash\{k\}} a^o_{mk}, & \ell = k \end{cases} \tag{14.12}$$

Observe how the entries of this policy are dependent on the gradient-noise factors:

$$\theta^2_k \;\triangleq\; \mathrm{Tr}(H^{-1}G_k), \quad k = 1, 2, \ldots, N \tag{14.13}$$

Observe also that these factors are not only dependent on G_k but that they also depend on the Hessian matrix information, H. In comparison, the relative-variance policy described in the next section will be independent of H. Recall from the derivation in Sec. 12.2 that the above Hastings rule is a solution to the optimization problem (12.18); it therefore minimizes the network MSD. While deriving the Hastings rule in Sec. 12.2, we formulated the problem in the context of cost functions, $\{J_k(w)\}$, that share a common minimizer. In this case, the minimizer, w^o, of the aggregate cost, $J^{\text{glob}}(w)$, defined by (8.44) will be invariant under the combination policy, A. For this reason, we can interpret Hastings rule (14.12) as providing a combination policy that results in the smallest possible MSD relative to the same fixed limit point w^o.

14.4 Relative-Variance Policy

We now describe a second noise-aware policy to select the combination weights; this second rule will be independent of the Hessian matrix information, H.

Recall that the Hastings rule was derived by working with the MSD expression (12.5), which results from keeping the first-order term in the MSD expression (11.178). The second policy that we shall derive here, and which we refer to as the relative-variance policy, is instead based on working with the alternative MSD expression (11.178). The derivation of this second policy does not require the uniformity conditions (14.11). Since the MSD performance levels of the distributed (consensus and diffusion) strategies (7.9), (7.18), and (7.19) agree to first-order in the step-size parameters, we shall motivate the combination rule by considering the ATC diffusion implementation.

To begin with, we know from (11.178) that the MSD performance of the ATC diffusion network (7.19) can be evaluated by means of the following series expression for sufficiently small step-sizes:

$$\text{MSD}_{\text{dist,av}}^{\text{atc}} \;=\; \frac{1}{hN}\sum_{n=0}^{\infty}\text{Tr}\left[\mathcal{B}_{\text{atc}}^{n}\mathcal{Y}_{\text{atc}}\left(\mathcal{B}_{\text{atc}}^{*}\right)^{n}\right] \tag{14.14}$$

where $h = 1$ for real data and $h = 2$ for complex data, and where the

matrix quantities $\{\mathcal{B}_{\rm atc}, \mathcal{Y}_{\rm atc}\}$ are defined as follows:

$$\mathcal{B}_{\rm atc} = \mathcal{A}^{\mathsf{T}}\left(I_{hMN} - \mathcal{M}\mathcal{H}\right) \tag{14.15}$$

$$\mathcal{Y}_{\rm atc} = \mathcal{A}^{\mathsf{T}}\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A} \tag{14.16}$$

which in turn are defined in terms of the quantities:

$$\mathcal{M} = \mathrm{diag}\left\{\mu_1 I_{hM},\ \mu_2 I_{hM}, \ldots, \mu_N I_{hM}\right\} \tag{14.17}$$

$$\mathcal{S} = \mathrm{diag}\{G_1, G_2, \ldots, G_N\} \tag{14.18}$$

$$\mathcal{R} = \mathrm{diag}\left\{H_1, H_2, \ldots, H_N\right\} \tag{14.19}$$

$$\mathcal{A} = A \otimes I_{hM} \tag{14.20}$$

and $\otimes$ is the Kronecker product operation.

Starting from (14.14), we pose the problem of seeking a left-stochastic combination matrix A that solves:

$$\begin{aligned} A^o \ \triangleq\ & \arg\min_{A\in\mathbb{A}} \sum_{n=0}^{\infty} \mathrm{Tr}\left[\mathcal{B}_{\rm atc}^n \mathcal{Y}_{\rm atc}\left(\mathcal{B}_{\rm atc}^*\right)^n\right] \\ & \text{subject to } \ A^{\mathsf{T}}\mathbb{1} = \mathbb{1},\ \ a_{\ell k} \geq 0,\ \ a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{aligned} \tag{14.21}$$

However, solving problem (14.21) is generally non-trivial and we replace it by a more tractable problem. Specifically, we replace the cost in (14.21) by an upper bound and minimize this upper bound instead. Indeed, it is shown in [208, Sec. 8.2] that the following inequality holds for a stable matrix $\mathcal{B}_{\rm atc}$:

$$\sum_{n=0}^{\infty} \mathrm{Tr}\left[\mathcal{B}_{\rm atc}^n \mathcal{Y}_{\rm atc}\left(\mathcal{B}_{\rm atc}^*\right)^n\right] \ \leq\ c\,\mathrm{Tr}(\mathcal{Y}_{\rm atc}) \tag{14.22}$$

for some finite positive constant c that is *independent* of A. In other words, the series is upper bounded by a multiple of the trace of $\mathcal{Y}_{\rm atc}$, which happens to be the first term of the series itself. Therefore, instead of minimizing the series in (14.21), we replace the problem by that of minimizing its first term, namely,

$$\begin{aligned} \min_{A\in\mathbb{A}}\ & \mathrm{Tr}(\mathcal{Y}_{\rm atc}) \\ & \text{subject to } \ A^{\mathsf{T}}\mathbb{1} = \mathbb{1},\ \ a_{\ell k} \geq 0,\ \ a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{aligned} \tag{14.23}$$

Using definition (14.16), the trace of $\mathcal{Y}_{\rm atc}$ can be expressed in terms of the combination coefficients $\{a_{\ell k}\}$ as follows:

$$\mathrm{Tr}(\mathcal{Y}_{\rm atc}) \;=\; \sum_{k=1}^{N}\sum_{\ell=1}^{N} \mu_\ell^2\, a_{\ell k}^2\, \mathrm{Tr}(G_\ell) \tag{14.24}$$

and it is seen that problem (14.23) can be decoupled into N separate optimization problems, one for each row of A:

$$\begin{aligned} \min_{\{a_{\ell k}\}_{\ell=1}^{N}} \quad & \sum_{\ell=1}^{N} \mu_\ell^2\, a_{\ell k}^2\, \mathrm{Tr}(G_\ell), \quad k=1,\ldots,N \\ & \text{subject to } \sum_{\ell=1}^{N} a_{\ell k} = 1,\; a_{\ell k} \geq 0,\; a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{aligned} \tag{14.25}$$

With each agent ℓ, we associate the following nonnegative scalar, which is proportional to the trace of the gradient noise moment matrix G_ℓ:

$$\gamma_\ell^2 \;\triangleq\; \mu_\ell^2\, \mathrm{Tr}(G_\ell), \quad \ell = 1,2,\ldots,N \tag{14.26}$$

The factor γ_ℓ^2 so defined plays a role similar to the factor θ_ℓ^2 defined earlier in (14.13) for the Hastings rule; note that both factors contain information about the noise moment matrix, G_ℓ.

Lemma 14.1 (Relative-variance rule)**.** The following combination matrix, denoted by A^o with a superscript o, is a solution to the optimization problem (14.25):

$$a_{\ell k}^o \;=\; \begin{cases} \dfrac{1}{\gamma_\ell^2}\left(\displaystyle\sum_{m\in\mathcal{N}_k} \frac{1}{\gamma_m^2}\right)^{-1}, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{14.27}$$

In the above construction, agent k combines the iterates from its neighbors in proportion to $1/\gamma_\ell^2$. The result is physically meaningful. Agents with smaller noise power, relative to the neighborhood noise power, are assigned larger weights.

Example 14.2 (Relative-variance rule for MSE networks)**.** We return to the setting of Example 14.1, which deals with MSE networks. The agents employ uniform step-sizes and the data have uniform regression covariance matrices, i.e., $\mu_k \equiv \mu$ and $R_{u,k} \equiv R_u$ for $k = 1, 2, \ldots, N$. In this case,

$$G_k \;=\; \sigma_{v,k}^2 \left[\begin{array}{cc} R_u & \times \\ \times & R_u^{\mathsf{T}} \end{array} \right] \tag{14.28}$$

so that expression (14.27) reduces to expression (14.7), namely,

$$a_{\ell k}^o \;=\; \frac{1}{\sigma_{v,\ell}^2} \left(\sum_{m \in \mathcal{N}_k} \frac{1}{\sigma_{v,m}^2} \right)^{-1}, \quad \ell \in \mathcal{N}_k \tag{14.29}$$

If the step-sizes are not uniform across the agents, then expression (14.27) would instead reduce to

$$a_{\ell k}^o \;=\; \frac{1}{\mu_\ell^2 \sigma_{v,\ell}^2} \left(\sum_{m \in \mathcal{N}_k} \frac{1}{\mu_m^2 \sigma_{v,m}^2} \right)^{-1}, \quad \ell \in \mathcal{N}_k \tag{14.30}$$

If both the step-sizes and the covariance matrices are not uniform across the agents, then expression (14.27) would lead to:

$$a_{\ell k}^o \;=\; \frac{1}{\mu_\ell^2 \sigma_{v,\ell}^2 \mathrm{Tr}(R_{u,\ell})} \left(\sum_{m \in \mathcal{N}_k} \frac{1}{\mu_m^2 \sigma_{v,m}^2 \mathrm{Tr}(R_{u,m})} \right)^{-1}, \quad \ell \in \mathcal{N}_k \tag{14.31}$$

■

14.5 Adaptive Combination Policy

To evaluate the relative-variance weights (14.27), the agents still need to know the gradient noise factors, $\{\gamma_\ell^2\}$, defined by (14.26). We motivate in this section a procedure for estimating these factors in an adaptive manner.

To begin with, we recall the definitions of the original and weighted aggregate cost functions:

$$J^{\mathrm{glob}}(w) \;\triangleq\; \sum_{k=1}^{N} J_k(w) \tag{14.32}$$

$$J^{\mathrm{glob},\star}(w) \overset{(8.53)}{\triangleq} \sum_{k=1}^{N} q_k J_k(w) \tag{14.33}$$

whose unique minima are denoted by w^o and $w^\star$, respectively. The individual costs, $\{J_k(w)\}$, are assumed to share a common minimizer and, hence, $w^o = w^\star$, i.e.,

$$\nabla_w \, J_k(w^\star) = 0, \quad k = 1, 2, \ldots, N \tag{14.34}$$

The common minimizer assumption ensures that the location of the global solution, w^o or $w^\star$, is fixed and invariant under A. This is a useful condition especially when A is implemented in an adaptive manner and varies with time.

We illustrate the construction of the adaptive combination policy by considering the ATC diffusion strategy (7.19), which is repeated here for ease of reference:

$$\left\{ \begin{array}{rcl} \boldsymbol{\psi}_{k,i} &=& \boldsymbol{w}_{k,i-1} - \mu_k \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \, \boldsymbol{\psi}_{\ell,i} \end{array} \right. \tag{14.35}$$

A similar construction applies to the CTA diffusion strategy (7.18) and the consensus strategy (7.9). The following result forms the basis for the procedure developed in this section for estimating the factors $\{\gamma_\ell^2\}$.

Lemma 14.2 (Useful expression for γ_ℓ^2)**.** Consider a network of N interacting agents running the distributed strategy (14.35) with a primitive left-stochastic matrix A. Under the same conditions in the statement of Theorem 9.2, it holds that

$$\mathbb{E}\,\|\boldsymbol{\psi}_{\ell,i}^e - \boldsymbol{w}_{\ell,i-1}^e\|^2 \;=\; \gamma_\ell^2 \;+\; o(\mu_{\max}^2), \quad \text{for } i \gg 1 \tag{14.36}$$

Proof. Using the mean-value theorem (D.20) from the appendix and (14.34) we note that we can write at an arbitrary agent ℓ:

$$\begin{aligned} \begin{bmatrix} \nabla_{w^*} J_\ell(\boldsymbol{w}_{\ell,i-1}) \\ \nabla_{w^\mathsf{T}} J_\ell(\boldsymbol{w}_{\ell,i-1}) \end{bmatrix} &= -\left(\int_0^1 \nabla_w^2 J_\ell(w^\star - r\widetilde{\boldsymbol{w}}_{\ell,i-1}) dr \right) \widetilde{\boldsymbol{w}}_{\ell,i-1}^e \\ &\overset{(8.138)}{=} -\boldsymbol{H}_{\ell,i-1} \widetilde{\boldsymbol{w}}_{\ell,i-1}^e \end{aligned} \tag{14.37}$$

where $\widetilde{\boldsymbol{w}}_{\ell,i-1} = w^\star - \boldsymbol{w}_{\ell,i-1}$ and

$$\widetilde{\boldsymbol{w}}_{\ell,i-1}^e \;\triangleq\; \begin{bmatrix} \widetilde{\boldsymbol{w}}_{\ell,i-1} \\ \left(\widetilde{\boldsymbol{w}}_{\ell,i-1}^*\right)^\mathsf{T} \end{bmatrix} \tag{14.38}$$

Therefore, in terms of the extended vectors and replacing the approximate gradient in terms of the sum of the true gradient and the gradient noise process, we can write for any arbitrary agent ℓ:

$$
\begin{aligned}
&\|\boldsymbol{\psi}_{\ell,i}^{e} - \boldsymbol{w}_{\ell,i-1}^{e}\|^2 \\
&\overset{(14.35)}{=} \mu_\ell^2 \left\| \begin{bmatrix} \boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{e,i-1}) \\ \left(\boldsymbol{s}_{\ell,i}^{e*}(\boldsymbol{w}_{e,i-1})\right)^{\mathsf{T}} \end{bmatrix} + \begin{bmatrix} \nabla_{w^*} J_\ell(\boldsymbol{w}_{\ell,i-1}) \\ \nabla_{w^{\mathsf{T}}} J_\ell(\boldsymbol{w}_{\ell,i-1}) \end{bmatrix} \right\|^2 \\
&\overset{(14.37)}{=} \mu_\ell^2 \, \|\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1}) - \boldsymbol{H}_{\ell,i-1}\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\|^2 \\
&\overset{(14.34)}{=} \mu_\ell^2 \, \|\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1})\|^2 + \mu_\ell^2 \|\boldsymbol{H}_{\ell,i-1}\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\|^2 - \\
&\qquad 2\mu_\ell^2 \, \mathrm{Re}\left[\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e*}\boldsymbol{H}_{\ell,i-1}\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1})\right]
\end{aligned} \tag{14.39}
$$

Now, we can deduce from an argument similar to (11.30) and from (11.8) that, for $i \gg 1$, and for sufficiently small step-sizes:

$$
\mathbb{E}\,\|\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1})\|^2 \;=\; \mathrm{Tr}(G_{s,\ell}) \;+\; O(\mu_{\max}^{\gamma'/2}) \tag{14.40}
$$

where $\gamma' = \min\{\gamma, 2\}$ and $\gamma \in (0, 4]$. Likewise, we can deduce from an argument similar to (9.280) that, for small step-sizes and for $i \gg 1$:

$$
\mathbb{E}\,\|\boldsymbol{H}_{\ell,i-1}\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\|^2 \;\le\; a\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\|^4 \overset{(9.107)}{=} O(\mu_{\max}^2) \tag{14.41}
$$

for some constant a that is independent of $\mu_{\max}$. Moreover, using the inequalities $|x^*y| \le \|x\|\,\|y\|$ for any vectors x and y, and $(\mathbb{E}\,\boldsymbol{a})^2 \le \mathbb{E}\,\boldsymbol{a}^2$ for any scalar real-valued random variable $\boldsymbol{a}$, we have

$$
\begin{aligned}
&\mathbb{E}\,\left[\,\left|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e*}\boldsymbol{H}_{\ell,i-1}\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1})\right| \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \\
&\le \left\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e*}\boldsymbol{H}_{\ell,i-1}\right\| \,\mathbb{E}\,\left[\,\|\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1})\| \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right] \\
&\le \sqrt{\left\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e*}\boldsymbol{H}_{\ell,i-1}\right\|^2} \sqrt{\mathbb{E}\,\left[\,\|\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1})\|^2 \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right]} \\
&\overset{(9.280)}{\le} \sqrt{a\left\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\right\|^4} \sqrt{\mathbb{E}\,\left[\,\|\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1})\|^2 \,|\, \boldsymbol{\mathcal{F}}_{i-1}\right]} \\
&\overset{(8.118)}{\le} \sqrt{a\left\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\right\|^4} \sqrt{(\beta_\ell^2/h^2)\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\|^2 + 2\sigma_{s,\ell}^2} \\
&\le \sqrt{a}\left\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\right\|^2 \left[(\beta_\ell/h)\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\| + \sqrt{2}\sigma_{s,\ell}\right] \\
&= \frac{\sqrt{a}\beta_\ell}{h}\left\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\right\|^3 + \sqrt{2a\sigma_{s,\ell}^2}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\|^2
\end{aligned} \tag{14.42}
$$

where $h = 1$ for real data and $h = 2$ for complex data. Taking expectations of both sides of (14.42), and using (9.11) and (9.107), we conclude that for small step-sizes and for $i \gg 1$:

$$\begin{aligned}
&\mathbb{E}\,\big|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e*}\boldsymbol{H}_{\ell,i-1}\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1})\big| \\
&\qquad\leq \frac{\sqrt{a}\beta_\ell}{h}\,\mathbb{E}\,\big\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\big\|^3 \;+\; \sqrt{2a\sigma_{s,\ell}^2}\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\|^2 \\
&\qquad\leq \frac{\sqrt{a}\beta_\ell}{h}\Big(\mathbb{E}\,\big\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\big\|^4\Big)^{3/4} \;+\; \sqrt{2a\sigma_{s,\ell}^2}\,\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e}\|^2 \\
&\qquad= \frac{\sqrt{a}\beta_\ell}{h}\big(O(\mu_{\max}^2)\big)^{3/4} \;+\; \sqrt{2a\sigma_{s,\ell}^2}\,O(\mu_{\max}) \\
&\qquad= \frac{\sqrt{a}\beta_\ell}{h}O(\mu_{\max}^{3/2}) \;+\; \sqrt{2a\sigma_{s,\ell}^2}\,O(\mu_{\max}) \\
&\qquad= O(\mu_{\max})
\end{aligned} \tag{14.43}$$

Using the fact that $|\mathrm{Re}(z)| \leq |z|$ for any complex number, we deduce from (14.43) that

$$\mathbb{E}\,\big|\mathrm{Re}\,\big[\widetilde{\boldsymbol{w}}_{\ell,i-1}^{e*}\boldsymbol{H}_{\ell,i-1}\boldsymbol{s}_{\ell,i}^{e}(\boldsymbol{w}_{\ell,i-1})\big]\big| \;=\; O(\mu_{\max}) \tag{14.44}$$

Substituting these results into (14.39) we conclude that for $i \gg 1$ we can write:

$$\begin{aligned}
\mathbb{E}\,\|\boldsymbol{\psi}_{\ell,i}^{e} - \boldsymbol{w}_{\ell,i-1}^{e}\|^2 \;&=\; \mu_\ell^2\mathrm{Tr}(G_{s,\ell}) \;+\; O\left(\mu_{\max}^{\min\{3,2+\frac{\gamma'}{2}\}}\right) \\
&\overset{(14.26)}{=}\; \gamma_\ell^2 \;+\; O\left(\mu_{\max}^{\min\{3,2+\frac{\gamma}{2}\}}\right) \\
&=\; \gamma_\ell^2 \;+\; o(\mu_{\max}^2)
\end{aligned} \tag{14.45}$$

as desired.

□

Result (14.45) shows that, for sufficiently small step-sizes, if we can approximate the limiting value of the variance that appears on the left-hand side of (14.36), after sufficient iterations have elapsed, then we would be able to estimate the desired factor γ_ℓ^2. We can estimate this variance iteratively by using at least one of two constructions.

Construction I: Agent-Centered Calculation

First, observe that

$$\mathbb{E}\left\|\boldsymbol{\psi}_{\ell,i}^{e}-\boldsymbol{w}_{\ell,i-1}^{e}\right\|^{2} \;=\; 2\,\mathbb{E}\left\|\boldsymbol{\psi}_{\ell,i}-\boldsymbol{w}_{\ell,i-1}\right\|^{2} \tag{14.46}$$

where the extended $2M\times 1$ vectors $\{\boldsymbol{\psi}_{\ell,i}^{e},\boldsymbol{w}_{\ell,i-1}^{e}\}$ are replaced by the regular $M\times 1$ vectors $\{\boldsymbol{\psi}_{\ell,i},\boldsymbol{w}_{\ell,i-1}\}$. Then, agent ℓ can estimate its variance parameter by running a smoothing filter of the following form:

$$\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)=(1-\zeta_{\ell})\,\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i-1)\;+\;\zeta_{\ell}\,\|\boldsymbol{\psi}_{\ell,i}\;-\;\boldsymbol{w}_{\ell,i-1}\|^{2} \tag{14.47}$$

where the quantities $\{\boldsymbol{\psi}_{\ell,i},\boldsymbol{w}_{\ell,i-1}\}$ that are needed to run the recursion are available at agent k. In this recursion, the notation $\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)$ denotes the estimator for γ_{ℓ}^{2} that is computed by agent ℓ at iteration i. Moreover, $0<\zeta_{\ell}\ll 1$ is a positive scalar much smaller than one. Note that under expectation, expression (14.47) gives

$$\mathbb{E}\,\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)=(1-\zeta_{\ell})\,\mathbb{E}\,\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i-1)\;+\;\zeta_{\ell}\,\mathbb{E}\,\|\boldsymbol{\psi}_{\ell,i}-\boldsymbol{w}_{\ell,i-1}\|^{2} \tag{14.48}$$

so that after sufficient iterations and using (14.36):

$$\mathbb{E}\,\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)\;\approx\;\gamma_{\ell}^{2}/2,\quad\text{for } i\gg 1 \tag{14.49}$$

That is, the estimator $\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)$ converges on average to the desired measure γ_{ℓ}^{2} (scaled by $1/2$); the scaling is irrelevant because it will appear in both the numerator and denominator of the expression for $a_{\ell k}^{o}$ in the relative-variance rule (14.27) and will therefore cancel out. Each agent ℓ can then share the estimator $\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)$ with its neighbors. That is, in this implementation, agent ℓ shares both $\boldsymbol{\psi}_{\ell,i}$ and $\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)$ with its neighbors. Using the iterates $\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)$, we can then replace the relative-variance weights (14.27) by their adaptive counterparts and write:

$$\boldsymbol{a}_{\ell k}^{o}(i)\;=\;\frac{1}{\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)}\left(\sum_{m\in\mathcal{N}_{k}}\frac{1}{\widehat{\boldsymbol{\gamma}}_{m}^{2}(i)}\right)^{-1},\quad\ell\in\mathcal{N}_{k} \tag{14.50}$$

Equations (14.47) and (14.50) provide one adaptive construction for the relative-variance combination weights $\{a_{\ell k}^{o}\}$. These adaptive weights

would be used in (14.35) to evaluate $\boldsymbol{w}_{k,i}$, and the process continues. The above procedure is valid for both real and complex data.

Adaptive relative-variance rule (agent-centered)
(individual costs have a common minimizer)

for each time instant $i \geq 0$ repeat:
  for each neighbor ℓ of agent $k = 1, 2, \ldots, N$ do :

$$\begin{aligned}
&\boldsymbol{y}_{\ell,i} \triangleq \boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{\ell,i-1} \quad \text{(ATC diffusion)} \\
&\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i) = (1-\zeta_\ell)\,\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i-1) + \zeta_\ell \,\|\boldsymbol{y}_{\ell,i}\|^2 \\
&\boldsymbol{a}_{\ell k}^{o}(i) = \frac{1}{\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)} \left(\sum_{m \in \mathcal{N}_k} \frac{1}{\widehat{\boldsymbol{\gamma}}_{m}^{2}(i)} \right)^{-1}, \quad \ell \in \mathcal{N}_k
\end{aligned} \tag{14.51}$$

  end
end

Construction II: Neighbor-Centered Calculation

There is an alternative implementation where we move the estimation of the parameter γ_ℓ^2 into the neighbors of agent ℓ; this mode of operation removes the need for transmitting $\widehat{\boldsymbol{\gamma}}_{\ell}^{2}(i)$ from agent ℓ to its neighbors. This advantage, however, comes at the expense of added computations as follows. Note that agent k now only has access to the iterate $\boldsymbol{\psi}_{\ell,i}$ that it receives from its neighbor ℓ. Agent k does not have access to $\boldsymbol{w}_{\ell,i-1}$ in the ATC diffusion implementation. To overcome this difficulty, we can, for example, replace $\boldsymbol{w}_{\ell,i-1}$ by $\boldsymbol{w}_{k,i-1}$ since for $i \gg 1$, the iterates at the various agents approach $w^\star$ within $O(\mu_{\max})$ with high probability and, hence,

$$\mathbb{E}\,\left\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{\ell,i-1}\right\|^2 \;\approx\; \mathbb{E}\,\left\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\right\|^2 \tag{14.52}$$

With this substitution, agent k can now estimate the variance γ_ℓ^2 of its neighbor locally by running a smoothing filter of the following form:

$$\boldsymbol{\gamma}_{\ell k}^{2}(i) = (1-\zeta_k)\,\boldsymbol{\gamma}_{\ell k}^{2}(i-1) + \zeta_k \,\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2 \tag{14.53}$$

where the quantities $\{\boldsymbol{\psi}_{\ell,i}, \boldsymbol{w}_{k,i-1}\}$ that are needed to run the recursion are available at agent k. In this recursion, we are employing the notation $\boldsymbol{\gamma}^2_{\ell k}(i)$, with two subscripts, to denote the estimator for γ^2_ℓ that is computed by agent k at iteration i. Thus, observe that now several estimators for the same quantity γ^2_ℓ are being computed: one by each neighbor of agent ℓ. Again, under expectation, expression (14.53) gives

$$\mathbb{E}\,\boldsymbol{\gamma}^2_{\ell k}(i) = (1-\zeta_k)\,\mathbb{E}\,\boldsymbol{\gamma}^2_{\ell k}(i-1) \;+\; \zeta_k\,\mathbb{E}\,\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2 \tag{14.54}$$

so that, again, after sufficient iterations and using (14.36):

$$\mathbb{E}\,\boldsymbol{\gamma}^2_{\ell k}(i) \;\approx\; \gamma^2_\ell/2, \quad \text{for } i \gg 1 \tag{14.55}$$

That is, the estimator $\boldsymbol{\gamma}^2_{\ell k}(i)$ converges on average to the desired measure γ^2_ℓ (scaled by $1/2$); the scaling is again irrelevant. Using the iterates $\boldsymbol{\gamma}^2_{\ell k}(i)$, we can replace the relative-variance weights (14.27) by their adaptive counterparts and write:

$$\boldsymbol{a}^o_{\ell k}(i) \;=\; \frac{1}{\boldsymbol{\gamma}^2_{\ell k}(i)}\left(\sum_{m\in\mathcal{N}_k}\frac{1}{\boldsymbol{\gamma}^2_{mk}(i)}\right)^{-1}, \quad \ell\in\mathcal{N}_k \tag{14.56}$$

Equations (14.53) and (14.56) provide another adaptive construction for the relative-variance combination weights $\{a^o_{\ell k}\}$. These adaptive weights would then be used in (14.35) to evaluate $\boldsymbol{w}_{k,i}$, and the process continues.

Adaptive relative-variance rule (neighbor-centered)
(individual costs have a common minimizer)

for each time instant $i \geq 0$ repeat:
  for each neighbor ℓ of agent $k = 1, 2, \ldots, N$ do :

$$\begin{aligned}
&\boldsymbol{y}_{\ell k,i} \;\triangleq\; \boldsymbol{\psi}_{\ell,i} \;-\; \boldsymbol{w}_{k,i-1} \quad \text{(ATC diffusion)}\\
&\boldsymbol{\gamma}^2_{\ell k}(i) = (1-\zeta_k)\,\boldsymbol{\gamma}^2_{\ell k}(i-1) \;+\; \zeta_k\,\|\boldsymbol{y}_{\ell k,i}\|^2\\
&\boldsymbol{a}^o_{\ell k}(i) \;=\; \frac{1}{\boldsymbol{\gamma}^2_{\ell k}(i)}\left(\sum_{m\in\mathcal{N}_k}\frac{1}{\boldsymbol{\gamma}^2_{mk}(i)}\right)^{-1}, \quad \ell\in\mathcal{N}_k
\end{aligned} \tag{14.57}$$

  end
end

Example 14.3 (Detecting intruders and agent clustering)**.** The following example is extracted from [214]. Allowing diffusion networks to adjust their combination coefficients in real-time enables the agents to assign smaller or larger weights to their neighbors depending on how well they contribute to the inference task. This capability can be exploited by the network to exclude harmful neighbors (such as intruders) [273]. For example, over MSE networks, the ATC diffusion strategy (7.23) with the adaptive combination weights (14.57) will take the following form.

ATC diffusion with adaptive combination weights

set $\boldsymbol{\gamma}_{\ell k}^2(-1) = 0$ for all $k = 1, 2, \ldots, N$ and $\ell \in \mathcal{N}_k$.

for $i \geq 0$ and for every agent k do :

$$
\begin{aligned}
\boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{k,i-1} + \tfrac{2\mu}{h}\boldsymbol{u}_{k,i}^{*}\left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}\right] \\
\boldsymbol{\gamma}_{\ell k}^2(i) &= (1-\zeta)\,\boldsymbol{\gamma}_{\ell k}^2(i-1) + \zeta\,\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2, \quad \ell \in \mathcal{N}_k \\
\boldsymbol{a}_{\ell k}(i) &= \frac{\boldsymbol{\gamma}_{\ell k}^{-2}(i)}{\sum_{m\in\mathcal{N}_k}\boldsymbol{\gamma}_{mk}^{-2}(i)}, \quad \ell \in \mathcal{N}_k \\
\boldsymbol{w}_{k,i} &= \sum_{\ell\in\mathcal{N}_k}\boldsymbol{a}_{\ell k}(i)\boldsymbol{\psi}_{\ell,i}
\end{aligned}
\tag{14.58}
$$

end

Figure 14.1 illustrates the ability of networks running algorithm (14.58) to detect intrusion, and also to perform agent clustering. The figure shows a network with $N = 20$ agents. One of the agents, say, agent ℓ_o, is an intruder and it feeds its neighbors irrelevant data such as sending them wrong iterates $\boldsymbol{\psi}_{\ell_o,i}$. In some other applications, agent ℓ_o may not be an intruder but is simply subject to measurements $\{\boldsymbol{d}_{\ell_o}, \boldsymbol{u}_{\ell_o,i}\}$ that arise from a different model, $w^{\blacktriangle}$, than the model w^o. The figure on the left shows the state of the combination weights after 300 diffusion iterations: the thickness of the edges reflect the size of the combination weights assigned to them; thicker edges correspond to larger weights. Observe how the edges connecting to the intruder are essentially cut-off by the algorithm. The figure on the right illustrates the ability of diffusion strategies to perform agent clustering (i.e., to separate into groups agents that are influenced by two different models, $w^{\blacktriangle}$ and w^o). Agents do not know beforehand which of their neighbors are influenced by which model. They also do not know which model is influencing their own data. By allowing agents to adapt their combination coefficients on the fly, it becomes possible for the agents to cut their links over time to neighbors that

are sensing a different model than their own. The net effect is that agents end up being clustered in two groups. Cooperation between the members of the same group then leads to the estimation of $\{w^{\blacktriangle}, w^{o}\}$.

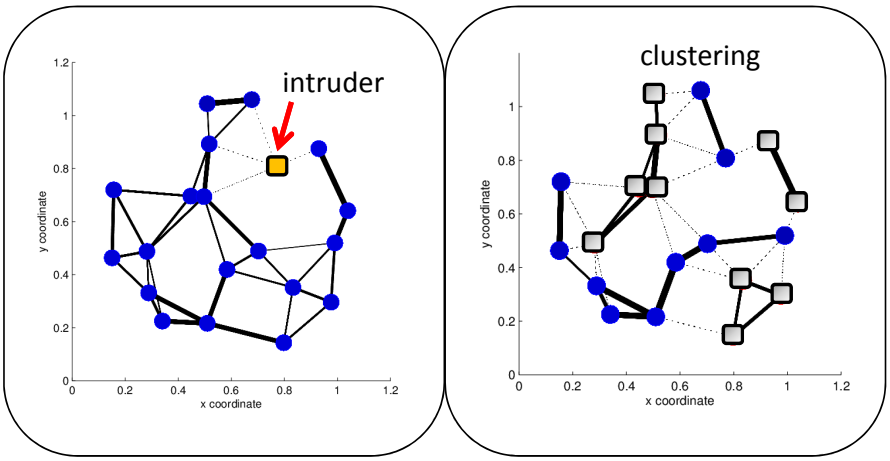

Figure 14.1: The figure on the left shows how diffusion cuts the links to the intruder. The figure on the right illustrates the clustering ability of the network.

■

Example 14.4 (Adapting combination weights over MSE networks)**.** We illustrate the performance of adaptive combination rules over MSE networks of the form described earlier in Example 6.3. We employ uniform step-sizes across the agents, $\mu_k = \mu = 0.001$. Figure 14.2 shows the connected network topology with $N = 20$ agents used for this simulation, with the measurement noise variances, $\{\sigma_{v,k}^2\}$, and the power of the regression data, assumed of the form $R_{u,k} = \sigma_{u,k}^2 I_M$, shown in the left and right plots of Figure 14.3, respectively.

Figure 14.4 plots the evolution of the ensemble-average learning curves, $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, for the ATC diffusion strategy (14.58) using four different combination rules: the left-stochastic uniform or averaging rule (11.148), the doubly-stochastic Metropolis rule (12.43), the relative-variance rule (14.31), and the adaptive combination rule (14.58) with uniform $\zeta_k = \zeta = 0.01$. The curves are obtained by averaging the trajectories $\{\frac{1}{N}\|\widetilde{\boldsymbol{w}}_i\|^2\}$ over 100 repeated experiments. The label on the vertical axis in the figure refers to the learning curves $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ by writing $\mathrm{MSD}_{\mathrm{dist,av}}(i)$, with an iteration index i. Each experiment involves running the diffusion strategy with $h = 2$ on complex-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ generated according to the model

$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i)$, with $M = 10$. The unknown vector w^o is generated randomly and its norm is normalized to one.

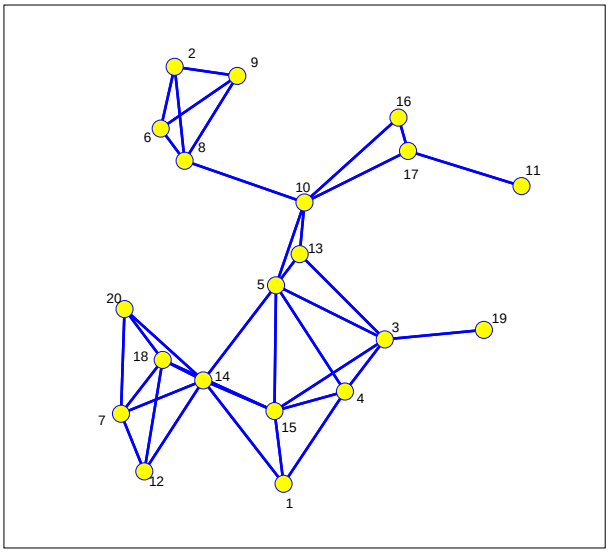

Figure 14.2: A connected network topology consisting of $N = 20$ agents employing the averaging rule (11.148).

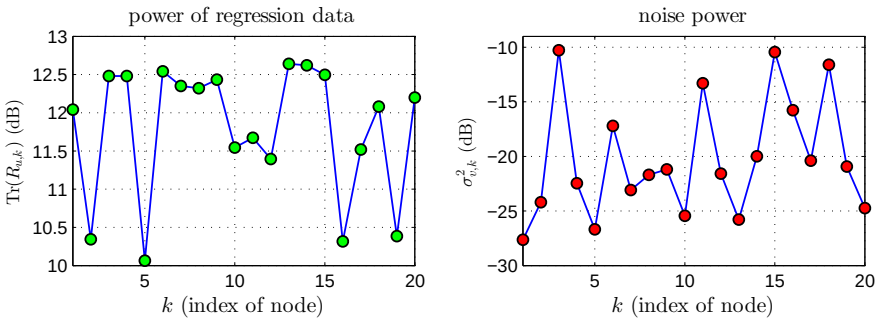

Figure 14.3: Measurement noise profile (left) and regression data power (right) across all agents. It is assumed that $R_{u,k} = \sigma^2_{u,k} I_M$, and the noise and regression data are Gaussian distributed.

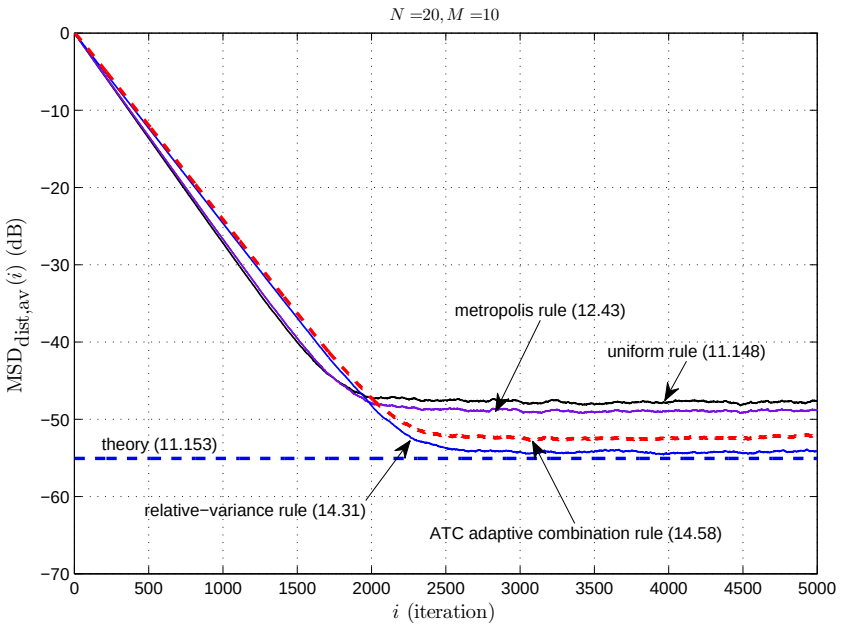

Figure 14.4: Evolution of the learning curves for the ATC diffusion strategy (14.58) using four different combination rules: the left-stochastic uniform or averaging rule (11.148), the doubly-stochastic Metropolis rule (12.43), the relative-variance rule (14.31), and the adaptive combination rule (14.58) with uniform $\zeta_k = \zeta = 0.01$.

It is further observed in the figure that the learning curve of the relative-variance rule tends to the MSD value predicted by the theoretical expression (11.153) with the entries $\{p_k\}$ corresponding to the Perron eigenvector that is associated with the combination policy (14.31), which reduces to the following expression in the example under consideration:

$$a_{\ell k}^{o} = \frac{1}{\sigma_{v,\ell}^2 \sigma_{u,\ell}^2} \left(\sum_{m \in \mathcal{N}_k} \frac{1}{\sigma_{v,m}^2 \sigma_{u,m}^2} \right)^{-1}, \quad \ell \in \mathcal{N}_k \tag{14.59}$$

It is also observed from Figure 14.4 that the adaptive rule is able to learn the noise factors $\{\gamma_\ell^2\}$ and to attain a performance level that is expected from the relative-variance rule. However, the convergence rate of the adaptive rule is clearly slower than the uniform and Metropolis rules: this is because of the additional adaptation process that is involved in learning the noise factors $\{\gamma_\ell^2\}$ and the combination coefficients $\{\boldsymbol{a}_{\ell k}(i)\}$. Schemes for speeding up the

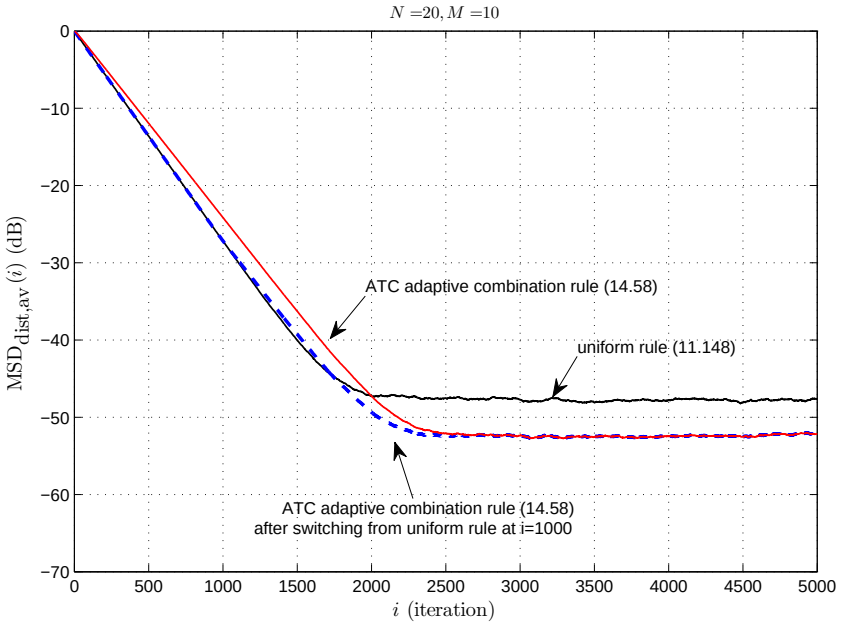

Figure 14.5: Evolution of the learning curves for the ATC diffusion strategy (14.58) using three different combination rules: the left-stochastic uniform or averaging rule (11.148), the adaptive combination rule (14.58) with uniform $\zeta_k = \zeta = 0.01$, and the same adaptive rule except that it is activated at $i = 1000$; during the initial 1000 iterations the network employs the uniform rule while the combination weights are being adapted.

convergence of the adaptive combination rule are proposed in [270] and [95]. One idea is based on training the network initially by using a static rule, such as the uniform rule, while the combination weights are being adapted and subsequently switch to the adaptive combination rule. Criteria for selecting the switching time is developed in these references. Figure 14.5 illustrates this construction where the switching time occurs at $i = 1000$. It is seen that the adaptive combination rule is able to recover the faster convergence rate of the uniform rule.

■

15

Extensions and Conclusions

This work provides an overview of strategies for adaptation, learning, and optimization over networks. Particular attention was given to the constant step-size case in order to enable solutions that are able to adapt and learn continuously from streaming data. There are of course several other important aspects of distributed strategies that were not covered in this work. Following [207, 208], we comment briefly on some of them and provide relevant references for the benefit of the reader.

15.1 Gossip and Asynchronous Strategies

It is possible to train networks whereby agents are not required to continually interact with all their neighbors at each time instant. Instead, agents may select a subset of their neighbors (or even a single neighbor) at every iteration. Figure 15.1 illustrates this situation graphically. The figure shows three successive instances of a network with the active edges highlighted by thicker lines. At each of these instants, agents select randomly a subset of their neighbors and share data with them over the selected links.

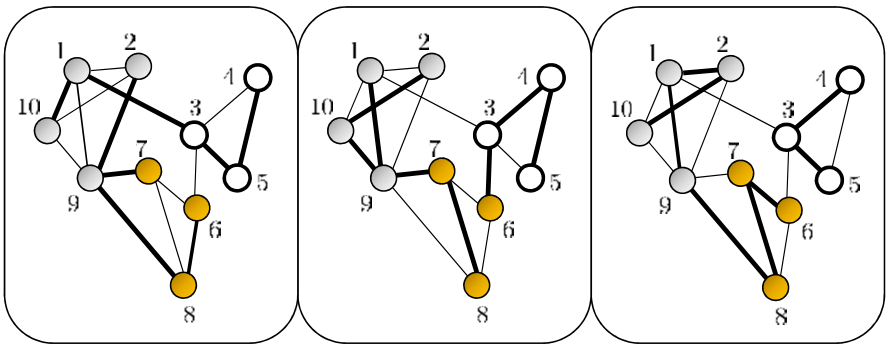

Figure 15.1: Three successive instances of a network with random selection of neighbors during the consultation process. The active edges are highlighted by the thicker lines. At each of these instants, agents select randomly a subset of their neighbors and share data with them over the selected links.

Criteria can be developed for determining which and how many neighbors to select. One simple strategy is to pick one neighbor at a time randomly, which is the case with useful gossip implementations for distributed processing (see, e.g., [14, 24, 43, 87, 137, 158, 201, 221]). For example, the ATC LMS diffusion implementation (7.23) based on the selection of a single neighbor per iteration would take the following form [201]:

$$\left\{\begin{array}{l} \boldsymbol{\psi}_{k,i} \;=\; \boldsymbol{w}_{k,i-1} \;+\; \frac{2}{h}\mu_k \boldsymbol{u}_{k,i}^{*}\,[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \\ \text{agent } k \text{ picks randomly a neighbor } \ell_o \in \mathcal{N}_k \\ \boldsymbol{w}_{k,i} \;=\; a_k\boldsymbol{\psi}_{k,i} \;+\; (1-a_k)\boldsymbol{\psi}_{\ell_o,i} \end{array}\right. \tag{15.1}$$

where $h = 2$ for complex data and $h = 1$ for real data, and where the scalar $a_k \in [0,1]$ denotes a convex combination coefficient. Useful variations that incorporate energy or game-theoretic considerations can also be pursued [102, 127]. One can also consider variations where agents share with their neighbors a subset of the entries in their vector iterates [9].

Moreover, an implicit assumption made in our presentation has been that all agents act synchronously. At every iteration, each agent completes its adaptation step before its neighbors initiate their combination steps. One can also study asynchronous implementations that

are subject to random events such as random data arrival times, random agent failures, random link failures, random topology changes, etc. There exist several studies in the literature on the performance of consensus and gossip-type strategies in response to asynchronous events or changing topologies [43, 124, 134–137, 195, 226, 242]. There are also studies in the context of diffusion strategies [158, 231, 277]. With the exception of the latter references on diffusion, most existing works investigate either pure averaging algorithms without streaming data, or assume noise-free data, or rely on the use of diminishing step-size sequences. In the works [277, 278], a fairly detailed analysis is carried out in the context of adaptation and learning with constant step-sizes. For example, the ATC diffusion update (7.19) in an asynchronous environment would take the following form:

$$\left\{\begin{array}{rcl} \boldsymbol{\psi}_{k,i} &=& \boldsymbol{w}_{k,i-1} - \boldsymbol{\mu}_k(i)\widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{a}_{\ell k}(i)\boldsymbol{\psi}_{\ell,i} \end{array}\right. \tag{15.2}$$

where the $\{\boldsymbol{\mu}_k(i), \boldsymbol{a}_{\ell k}(i)\}$ are now *time-varying* and *random* step-sizes and combination coefficients, and $\boldsymbol{\mathcal{N}}_{k,i}$ denotes the *random* neighborhood of agent k at time i. The underlying network is therefore randomly varying. Two of the main results established in [277, 278], following techniques similar to this work, are that, under some independence conditions on the random events, the asynchronous network continues to be mean-square stable for sufficiently small step-sizes. Moreover, its convergence rate and MSD performance compare well to those of the synchronous network that is constructed by employing the average values for the step-sizes and the average values for the combination coefficients, namely,

$$\alpha_{\text{async}} = \alpha_{\text{sync}} + O\left(\mu_{\max}^{1+1/N^2}\right) \tag{15.3}$$

$$\text{MSD}_{\text{async,av}} = \text{MSD}_{\text{sync,av}} + O\left(\mu_{\max}\right) \tag{15.4}$$

where $\mu_{\max}$ is now defined in terms of an upper bound on the random step-size parameters (and is sufficiently small). In other words, the convergence rate remains largely unaffected by asynchronous events at

the expense of a deterioration in the order of $O(\mu_{\max})$ in MSD performance. These results help justify the remarkable robustness and resilience properties of cooperative networks in the face of random failures at multiple levels: agents, links, data, and topology.

15.2 Noisy Exchanges of Information

We ignored in our presentation the effect of perturbations during the exchange of information among neighboring agents. These perturbations can arise from different sources, including noise over the communication links, quantization effects (e.g., [13, 87, 203]), attenuation and fading effects. To model distortions over links, one can introduce, for example, additive noise components and attenuation components into the steps involving the exchange of iterates among neighboring agents. This situation is illustrated generically in Figure 15.2 for an agent k receiving data from its neighbors $\{\ell, 4, 7\}$. The scalars $\{\boldsymbol{\gamma}_{\ell k}(i),\ \ell \in \mathcal{N}_k\}$ model attenuation or fading effects and the noise sources $\{\boldsymbol{v}_{\ell k}(i),\ \ell \in \mathcal{N}_k\}$ model additive noise components over the edges linking the neighbors to agent k. Such distortions influence the performance of distributed strategies as follows.

For example, in the diffusion LMS network of Example 7.3, the same iterate $\boldsymbol{\psi}_{\ell,i}$ is broadcast by agent ℓ to all its neighbors. When this is done, different noise sources interfere with the exchange of $\boldsymbol{\psi}_{\ell,i}$ over each of the edges that link agent ℓ to its neighbors. Thus, agent k will end up receiving the perturbed iterate:

$$\boldsymbol{\psi}_{\ell k,i} \;=\; \boldsymbol{\gamma}_{\ell k}(i)\boldsymbol{\psi}_{\ell,i} \;+\; \boldsymbol{v}_{\ell k,i}^{(\psi)} \tag{15.5}$$

where $\boldsymbol{v}_{\ell k,i}^{(\psi)}$ denotes the additive noise component over the edge from ℓ to k, and $\boldsymbol{\gamma}_{\ell k}(i)$ denotes the attenuation effect. The actual ATC diffusion implementation ends up being:

$$\begin{cases} \boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^{*}\,[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell k,i} \end{cases} \tag{15.6}$$

with the $\{\boldsymbol{\psi}_{\ell k,i}\}$ appearing in the combination step in (15.6) in place of

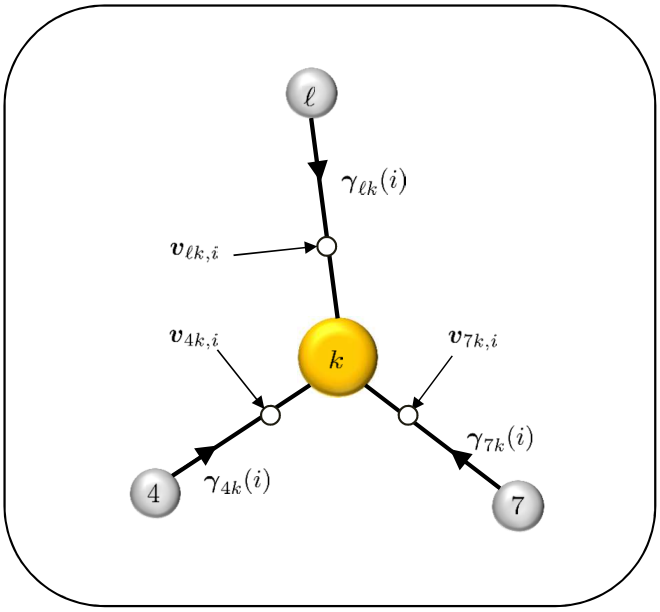

Figure 15.2: Data $\{\boldsymbol{\psi}_{\ell,i}\}$ sent to agent k from its neighbors undergo additive noise perturbations, represented by the noise sources $\{\boldsymbol{v}_{\ell k}(i),\ \ell \in \mathcal{N}_k\}$, as well as attenuation or fading effects, represented by the scaling coefficients $\{\gamma_{\ell k}(i),\ \ell \in \mathcal{N}_k\}$.

$\boldsymbol{\psi}_{\ell,i}$. It is seen that the perturbations interfere with the quality of the iterates $\{\boldsymbol{w}_{k,i}\}$. Studying the degradation in performance that results from these noisy exchanges, and developing adaptive combination rules that counter the effect of such degradation, can be pursued by extending the mean-square analysis of the earlier chapters. Readers can refer to [1, 141, 208, 244, 274, 280] for results on diffusion strategies and to [135, 166] for results on consensus strategies.

15.3 Exploiting Temporal Diversity

We can also develop distributed strategies that incorporate an additional temporal processing step besides the spatial aggregation step [76, 151, 152, 208]. The temporal step is reminiscent of momentum-type techniques proposed for gradient descent optimization [11, 21, 176, 177]

in that the agents update their states by relying on additional *past* values of their iterates besides the most recent iterates. Note, for instance, that in the LMS diffusion strategies of Example 7.3, each agent shares information locally with its neighbors through a process of spatial cooperation represented by the aggregation step. We can add a temporal dimension to this cooperative behavior as follows. For example, in the ATC LMS implementation (7.23), rather than have each agent k rely solely on the current weight iterates received from its neighbors, $\{\boldsymbol{\psi}_{\ell,i},\ \ell \in \mathcal{N}_k\}$, agent k can also be allowed to store and process its present and past weight iterates, say, L of them as in $\{\boldsymbol{\psi}_{k,j},\ j = i, i-1, \ldots, i-L+1\}$. There are several ways by which temporal processing can be added. The following equations describe one possibility for MSE networks of the form described in Example 6.3 [152]:

$$\begin{cases} \boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{k,i-1} \ + \ \frac{2}{h}\mu_k \boldsymbol{u}_{k,i}^{*}\,[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \\ \boldsymbol{\phi}_{k,i} &= \displaystyle\sum_{j=0}^{L-1} f_{kj}\boldsymbol{\psi}_{k,i-j} \qquad \text{(temporal processing)} \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\phi}_{\ell,i} \qquad \text{(spatial processing)} \end{cases} \tag{15.7}$$

where $h = 1$ for real data and $h = 2$ for complex data, and the coefficients $\{f_{kj}\}$ are chosen to satisfy

$$f_{kj} \geq 0, \quad \sum_{j=0}^{L-1} f_{kj} \ = \ 1 \tag{15.8}$$

In this way, previous weight iterates are smoothed and used to help counter the effect of noise over the communication links. Figure 15.3 illustrates the three steps of adaptation (A), temporal processing (T), and spatial processing (S) that are involved in the implementation (15.7). The order of these three steps can be interchanged, thus leading to other variations of the diffusion implementation. The version listed above is ATS diffusion, where the order of the letters in "ATS" refers to the order in which the processing steps appear in the algorithm implementation. [152].

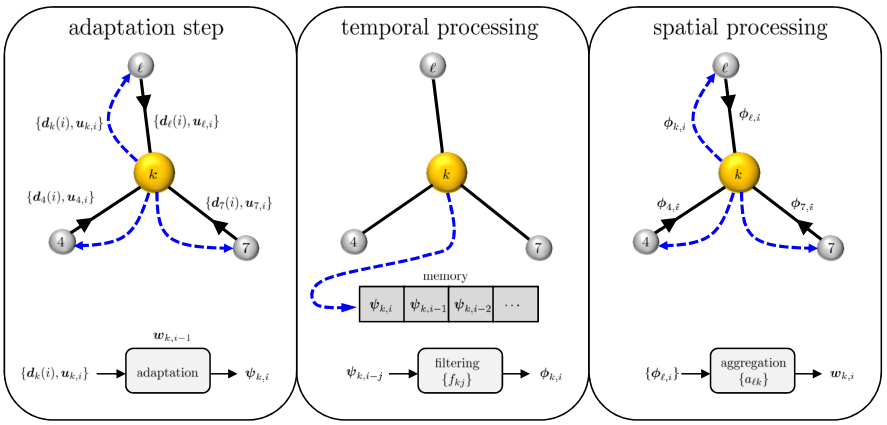

Figure 15.3: From left to right: the three steps of adaptation (A), temporal processing (T), and spatial processing (S) that are involved in the diffusion implementation (15.7).

Other possibilities for the addition of temporal processing can be pursued. For example, reference [76] starts from the CTA diffusion algorithm (7.22) and incorporates a useful *projection* step between the combination step and the adaptation step. The projection step uses the iterate, $\boldsymbol{\psi}_{k,i-1}$, at node k and projects it onto hyperslabs defined by the current and past raw data. Specifically, the algorithm from [76] has the following form:

$$\left\{\begin{array}{rcl} \boldsymbol{\psi}_{k,i-1} & = & \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{\phi}_{k,i-1} & = & \mathcal{P}'_{k,i}[\boldsymbol{\psi}_{k,i-1}] \\ \boldsymbol{w}_{k,i} & = & \boldsymbol{\phi}_{k,i-1} - \mu_k\left\{\boldsymbol{\phi}_{k,i-1} - \displaystyle\sum_{j=0}^{L-1} f_{kj}\,\mathcal{P}_{k,i-j}[\boldsymbol{\phi}_{k,i-1}]\right\} \end{array}\right. \tag{15.9}$$

where the notation $\phi = \mathcal{P}_{k,i}[\psi]$ refers to the act of projecting the vector ψ onto the hyperslab $P_{k,i}$ that consists of all $M\times 1$ vectors z satisfying

(similarly for the projection $\mathcal{P}'_{k,i}$):

$$P_{k,i} \triangleq \{ z \text{ such that } |\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} z| \leq \epsilon_k \} \tag{15.10}$$

$$P'_{k,i} \triangleq \{ z \text{ such that } |\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} z| \leq \epsilon'_k \} \tag{15.11}$$

where $\{\epsilon_k, \epsilon'_k\}$ are positive (tolerance) parameters chosen by the designer to satisfy $\epsilon'_k > \epsilon_k$. For generic values $\{d, u, \epsilon\}$, where d is a scalar and u is a row vector, the projection operator is described analytically by the following expression [222]:

$$\mathcal{P}[\psi] = \psi + \begin{cases} \frac{u^*}{\|u\|^2}[d - \epsilon - u\psi], & \text{if } d - \epsilon > u\psi \\ \frac{u^*}{\|u\|^2}[d + \epsilon - u\psi], & \text{if } d + \epsilon < u\psi \\ 0, & \text{if } |d - u\psi| \leq \epsilon \end{cases} \tag{15.12}$$

The projections that appear in (15.9) can be regarded as another example of a temporal processing step.

15.4 Incorporating Sparsity Constraints

We may also consider distributed strategies that enforce sparsity constraints on the solution vector (e.g., [74, 75, 86, 157]). For example, in the context of the MSE networks of Example 6.3, we may consider individual costs of the following modified form:

$$J_k(w) = \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w|^2 + \rho\, f(w) \tag{15.13}$$

where $f(w)$ is some real-valued convex function weighted by some parameter $\rho > 0$. The role of $f(w)$ is to help ensure that the solution vectors are sparse [17, 51, 235]. One ATC diffusion strategy for solving such problems takes the form [86]:

$$\boldsymbol{e}_k(i) = \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1} \tag{15.14}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}^*_{k,i} \boldsymbol{e}_k(i) - \rho\mu_k\, \partial f(\boldsymbol{w}_{k,i-1}) \tag{15.15}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{\psi}_{\ell,i} \tag{15.16}$$

where $\partial f(\cdot)$ denotes a sub-gradient vector for $f(w)$ relative to w. Various possibilities exist for the selection of $f(w)$ and its sub-gradient vector. One choice is

$$\partial f(w) = \text{sign}(w) \tag{15.17}$$

where the entries of the column vector $\text{sign}(w)$ are defined as follows in terms of the individual entries of w:

$$[\text{sign}(w)]_m \triangleq \begin{cases} w_m/|w_m|, & w_m \neq 0 \\ 0, & w_m = 0 \end{cases} \tag{15.18}$$

A second choice is to use instead

$$\partial f(w) = \left[\frac{\text{sign}(w_1)}{\epsilon+|w_1|} \quad \frac{\text{sign}(w_2)}{\epsilon+|w_2|} \quad \ldots \quad \frac{\text{sign}(w_M)}{\epsilon+|w_M|} \right] \tag{15.19}$$

This second choice has the advantage of selectively shrinking those components of the iterate $\boldsymbol{w}_{k,i-1}$ whose magnitudes are comparable to ϵ with little effect on components whose magnitudes are much larger than ϵ (see, e.g., [51, 73, 147]). Greedy techniques can also be used to develop useful sparsity-aware diffusion strategies, as shown in [74].

15.5 Distributed Constrained Optimization

Distributed strategies can also be developed for the solution of *constrained* convex optimization problems of the form:

$$\begin{aligned} &\min_{w} \sum_{k=1}^{N} J_k(w) \\ &\text{subject to } w \in \mathbb{W}_1 \cap \mathbb{W}_2 \cap \ldots \cap \mathbb{W}_N \end{aligned} \tag{15.20}$$

where each $J_k(w)$ is convex and each $\mathbb{W}_k$ is a convex set of points w that satisfy a collection of affine equality constraints and convex inequality constraints, say, as:

$$\mathbb{W}_k \triangleq \begin{cases} w: & h_{k,m}(w) = 0, \quad m = 1, 2, \ldots, U_k \\ & g_{k,n}(w) \leq 0, \quad n = 1, 2, \ldots, L_k \end{cases} \tag{15.21}$$

The key challenge in solving such problems in a distributed manner is that each agent k should only be aware of its cost function, $J_k(w)$, and its $L_k + U_k$ total constraints. For this reason, some available solution methods are in effect non-distributed because they require each agent to know all constraints from across the network [196]. If the feasible set and the constraints happen to be agent-independent, then such solution methods become distributed.

More generally, when solving constrained optimization problems of the form (15.20) in a distributed manner, it is customary to rely on the use of useful projection steps in order to ensure that the successive iterates that are computed by the agents satisfy the convex constraints — see, e.g., [63, 76, 153, 226, 234, 268]. An insightful overview of the use of projection methods in optimization problems is given in [234]. We already encountered one example of a projection-based solution method in (15.9). Nevertheless, solution techniques that rely on the use of projection operations require the constraint conditions to be relatively simple in order for the distributed algorithm to be able to compute the necessary projections analytically (such as projecting onto the nonnegative orthant) [153, 226, 268]. For example, the following form of the diffusion CTA strategy (7.18) with projections is used in [153]:

$$\boldsymbol{\psi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{w}_{\ell,i-1} \tag{15.22}$$

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu(i)\,\nabla_{w^{\mathsf{T}}} J_k\left(\boldsymbol{\psi}_{k,i-1}\right) \tag{15.23}$$

$$\boldsymbol{w}_{k,i} = \mathcal{P}_{\mathbb{W}_k}[\boldsymbol{\phi}_{k,i}] \tag{15.24}$$

In this construction, the main motivation is to solve a static optimization problem (in lieu of adaptation and learning). Thus, note that the *actual* gradient vector is employed in (15.23) along with a *decaying* step-size sequence. Moreover, the notation $\mathcal{P}_{\mathbb{W}_k}[\cdot]$ denotes projection onto the set $\mathbb{W}_k$; each of these sets is required to consist of "simple constraints" so that the projections can be carried out analytically. Motivated by these considerations, the work in [237, 238] develops distributed strategies that circumvent projection steps. The solution relies on the use of suitably chosen penalty functions and replaces the projection step by a stochastic approximation update that runs simultaneously with the optimization step. One form of this diffusion solution can be described as follows. We select continuous, convex, and twice-differentiable functions $\delta^{\mathrm{IP}}(x)$ and $\delta^{\mathrm{EP}}(x)$ that satisfy the properties:

$$\delta^{\rm IP}(x) \;=\; \begin{cases} 0, & x \leq 0 \\ > 0, & x > 0 \end{cases} \tag{15.25}$$

and

$$\delta^{\rm EP}(x) \;=\; \begin{cases} 0, & x = 0 \\ > 0, & x \neq 0 \end{cases} \tag{15.26}$$

with $\delta^{\rm IP}(x)$ being additionally a non-decreasing function. For example, the following continuous, convex, and twice-differentiable functions satisfy these conditions for small ρ:

$$\delta^{\rm IP}(x) = \max\left\{0, \frac{x^3}{\sqrt{x^2+\rho^2}}\right\}, \qquad \delta^{\rm EP}(x) = x^2 \tag{15.27}$$

Using the functions $\{\delta^{\rm IP}(x), \delta^{\rm EP}(x)\}$, we associate with each agent k the following penalty function, which takes into account all constraints at the agent:

$$p_k(w) \;\stackrel{\Delta}{=}\; \sum_{n=1}^{L_k} \delta^{\rm IP}\left(g_{k,n}(w)\right) \;+\; \sum_{m=1}^{U_k} \delta^{\rm EP}\left(h_{k,m}(w)\right) \tag{15.28}$$

The penalized ATC diffusion form for solving (15.20) then takes the following form for any parameter $0 < \theta < 1$ [237, 238]:

$$\boldsymbol{\psi}_{k,i} \;=\; \boldsymbol{w}_{k,i-1} - \mu\,\widehat{\nabla_{w^{\mathsf T}} J_k}(\boldsymbol{w}_{k,i-1}) \tag{15.29}$$

$$\boldsymbol{\phi}_{k,i} \;=\; \boldsymbol{\psi}_{k,i} - \mu^{1-\theta}\,\nabla_{w^{\mathsf T}} p_k(\boldsymbol{\psi}_{k,i}) \tag{15.30}$$

$$\boldsymbol{w}_{k,i} \;=\; \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\phi}_{\ell,i} \tag{15.31}$$

One of the main conclusions in [237, 238] is that, under certain conditions on the cost and penalty functions and gradient noise, and for sufficiently small step-sizes μ and a doubly-stochastic combination policy A, it holds that

$$\lim_{\mu\to 0}\,\limsup_{i\to\infty}\,\mathbb{E}\,\|w^o - \boldsymbol{w}_{k,i}\|^2 = 0 \tag{15.32}$$

where w^o denotes the unique optimal solution for (15.20) for a strongly-convex aggregate cost $J^{\rm glob}(w)$.

Following [237, 238], we illustrate the operation of the algorithm by considering the network shown in Figure 15.4 with $N = 20$ agents

running the penalized diffusion algorithm (15.29)–(15.31) using the Metropolis rule (12.43) with $\mu = 0.002$, $\theta = 0.9$, and $\rho = 0.001$. Each agent in the network is associated with a mean-square-error cost of the form $J_k(w) = \mathbb{E}\,(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w)^2$, where the observed data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ are related to each other via a linear regression model of the form:

$$\boldsymbol{d}_k(i) \;=\; \boldsymbol{u}_{k,i}w^{\bullet} + \boldsymbol{v}_k(i) \tag{15.33}$$

for some unknown model $w^{\bullet}$. To illustrate the adaptation and tracking ability of the algorithm, we associate a single linear inequality constraint with each agent. Specifically, we set $L_k = 1$, $U_k = 0$ and choose:

$$g_{k,i}(w) \;=\; b_{k,i}^{\mathsf{T}}w - z_k(i) \tag{15.34}$$

where $\{b_{k,i}, z_k(i)\}$ are allowed to change with the iteration index, i. If we introduce the block quantities:

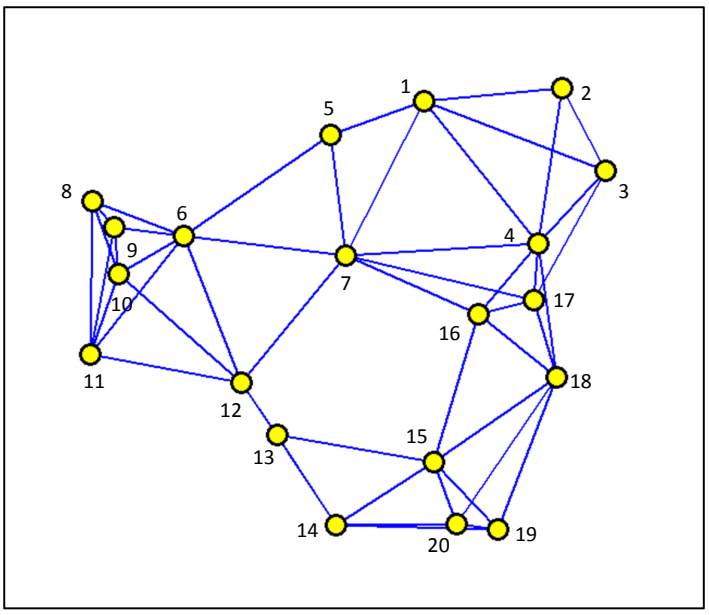

Figure 15.4: A connected network topology consisting of $N = 20$ agents running the penalized diffusion algorithm (15.29)–(15.31).

$$B_i \triangleq \mathrm{col}\{b_{1,i}^{\mathsf{T}},\, b_{2,i}^{\mathsf{T}}, \ldots, b_{N,i}^{\mathsf{T}}\} \tag{15.35}$$

$$z_i \triangleq \mathrm{col}\{z_1(i),\, z_2(i), \ldots, z_N(i)\} \tag{15.36}$$

then we have that the global optimization problem that we are interested in solving is of the form:

$$\begin{aligned} &\min_{w} \sum_{k=1}^{N} \mathbb{E}\,(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w)^2 \\ &\text{subject to } B_i w - z_i \preceq 0 \end{aligned} \tag{15.37}$$

where the notation $a \preceq b$, for two vectors a and b, indicates element-wise comparison of the entries of the vectors. While the projections associated with the constraints in this problem may be solved analytically, this setup is simply meant to illustrate the operation of the penalized diffusion algorithm and its tracking ability.

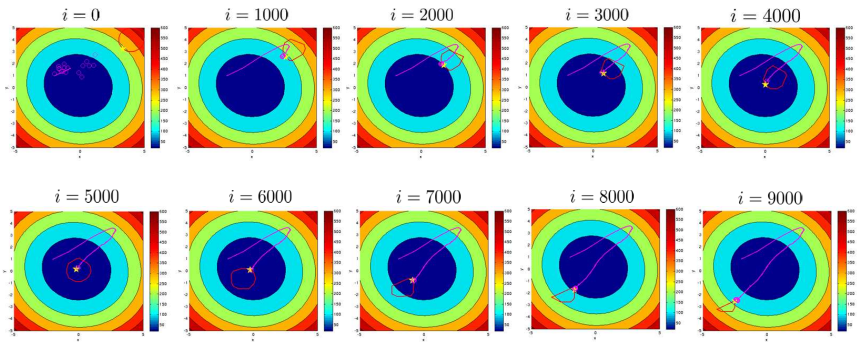

Figure 15.5: The star indicates the location of the optimal minimizer, w_i^o, which is allowed to drift in this simulation to illustrate the tracking ability of the algorithm. The polygon in the graph denotes the boundary of the feasible region for each agent. The tiny circles (e.g., in the left-most plot in the first row) illustrate the location of the iterates by the agents, the line denotes the average estimated trajectory by the network. As the constraint set changes, it is observed that the iterates are able to track the minimizer even as the feasible region shrinks and changes with time.

The statistical distribution of the random processes $\{\boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$ remain invariant for the duration of the simulation; only the constraints

drift with time. This need not be the case in general, and the diffusion algorithm can also handle non-stationary cost functions; keeping the cost function fixed facilitates the illustration of the results. The variance of the noise $\boldsymbol{v}_k(i)$ is selected randomly according to a uniform distribution from within the open interval $\sigma_{v,k}^2 \in (0,1)$. The covariance matrices $R_{u,k} = \mathbb{E}\,\boldsymbol{u}_{k,i}^{\mathsf{T}}\boldsymbol{u}_{k,i}$ are generated as $R_{u,k} = Q_k\Lambda_k Q_k^{\mathsf{T}}$, where Q_k is a randomly generated orthogonal matrix and Λ_k is a diagonal matrix with random elements also selected uniformly from within the interval $(0,1)$. The model vector $w^{\bullet} \in \mathbb{R}^2$ is chosen randomly. The constraint set is also initialized randomly and changes as time progresses.

Figure 15.5 illustrates the evolution of the iterates across the agents as time progresses. It is observed that the agents are attracted towards the feasible region from their initial position and quickly converge towards the true optimizer, w_i^o, which is initially stationary. As the constraint set changes over time, we observe that each agent's iterate changes and tracks w_i^o. The magenta line in the figure denotes the average estimated trajectory by the network.

15.6 Distributed Recursive Least-Squares

We can also apply diffusion strategies to solve recursive least-squares (RLS) problems in a distributed manner [28, 57, 58]. Consensus-based solutions also appear in [165, 266, 267]. For example, consider a collection of N agents observing data $\{d_k(i), u_{k,i}\}$, which are assumed to be related via:

$$d_k(i) \;=\; u_{k,i}w^o \;+\; v_k(i) \tag{15.38}$$

where $u_{k,i}$ is a $1 \times M$ regression vector and w^o is the $M \times 1$ unknown vector to be estimated in a least-squares sense by minimizing the global cost

$$\min_{w} \;\; \lambda^{i+1}\delta\|w\|^2 \;+\; \sum_{j=0}^{i}\lambda^{i-j}\left(\sum_{k=1}^{N}|d_k(j) - u_{k,j}w|^2\right) \tag{15.39}$$

where $0 \ll \lambda \leq 1$ is an exponential forgetting factor whose value is usually close to one. Distributed recursive least-squares (RLS) strategies of the diffusion-type for the solution of (15.39) were developed in

[57, 58] and they take the following form. Let $w_{k,i}$ denote the estimate for w^o that is computed by agent k at time i. For every agent k, we start with the initial conditions $w_{k,-1} = 0$ and $P_{k,-1} = \delta^{-1} I_M$, where $P_{k,-1}$ is an $M \times M$ matrix and $\delta > 0$ (usually a small number). Then, every agent k repeats the calculations listed in (15.40) by cooperating with its neighbors, where the symbol $\leftarrow$ denotes a sequential assignment. The scalars $\{a_{\ell k}, c_{\ell k}\}$ are nonnegative combination coefficients satisfying for all $k = 1, 2, \ldots, N$:

Diffusion RLS strategy (ATC)

step 1 (initialization by agent k)

$$\begin{aligned}
&\psi_{k,i} \leftarrow w_{k,i-1} \\
&P_{k,i} \leftarrow \lambda^{-1} P_{k,i-1}
\end{aligned}$$

step 2 (adaptation)

Update $\{\psi_{k,i}, P_{k,i}\}$ by iterating over $\ell \in \mathcal{N}_k$:

$$\begin{aligned}
\psi_{k,i} &\leftarrow \psi_{k,i} + \frac{c_{\ell k} P_{k,i} u_{\ell,i}^*}{1 + c_{\ell k} u_{\ell,i} P_{k,i} u_{\ell,i}^*} (d_{\ell,i} - u_{\ell,i} \psi_{k,i}) \\
P_{k,i} &\leftarrow P_{k,i} - \frac{c_{\ell k} P_{k,i} u_{\ell,i}^* u_{\ell,i} P_{k,i}}{1 + c_{\ell k} u_{\ell,i} P_{k,i} u_{\ell,i}^*}
\end{aligned} \tag{15.40}$$

end

step 3 (combination)

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_{\ell,i}$$

$$c_{\ell k} \geq 0, \quad \sum_{k=1}^{N} c_{\ell k} = 1, \quad c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{15.41}$$

$$a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{15.42}$$

That is, $A = [a_{\ell k}]$ is a left-stochastic matrix and $C = [c_{\ell k}]$ is a right-stochastic matrix. Figure 15.6 illustrates the exchange of information that occurs during the adaptation and combination steps in the diffusion implementation. During the adaptation step, agents exchange their data measurements $\{d_\ell(i), u_{\ell,i}\}$ with their neighbors, and during the consultation step agents exchange their intermediate iterates $\{\psi_{\ell,i}\}$.

Under some approximations, and for the special choices $\lambda = 1$ and $A = C$ (in which case A becomes doubly stochastic), the diffusion RLS strategy (15.40) can be reduced to a form given in [267] and which is described by the following equations:

$$P_{k,i}^{-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \left[P_{\ell,i-1}^{-1} + u_{\ell,i}^* u_{\ell,i} \right] \tag{15.43}$$

$$q_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \left[q_{\ell,i-1} + u_{\ell,i}^* d_\ell(i) \right] \tag{15.44}$$

$$\psi_{k,i} = P_{k,i} q_{k,i} \tag{15.45}$$

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_{\ell,i} \tag{15.46}$$

Algorithm (15.43)–(15.46) is computationally more demanding (by one order of magnitude) than diffusion RLS since step (15.45) requires $P_{k,i}$, which is recovered by inverting the matrix $P_{k,i}^{-1}$ that is evaluated in the first step (15.43). The above form was motivated in [267] by using consensus arguments; reference [208] provides more details on the connections and differences between the diffusion strategy (15.40) and the above consensus strategy.

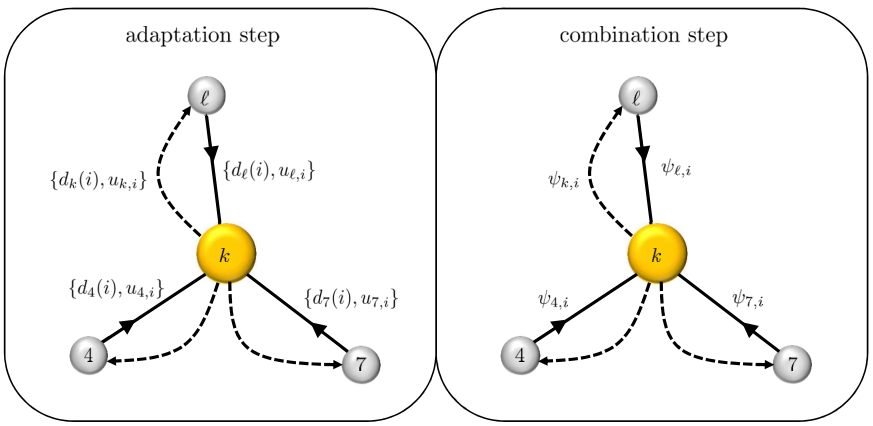

Figure 15.6: During the adaptation step 2 in the diffusion RLS implementation (15.40), agents exchange their data measurements $\{d_\ell(i), u_{\ell,i}\}$ (left). During the consultation step 3, agents exchange their intermediate iterates $\{\psi_{\ell,i}\}$ (right).

Returning to (15.40), we observe that the second step involves updating a Riccati-type variable, $P_{k,i}$, which is supposed to remain positive-definite over time. In order to avoid numerical difficulties that may destroy this critical property, it is often preferred to implement such update schemes in array form [133, 206], where a Cholesky factor of $P_{k,i}$ is updated rather than $P_{k,i}$ itself. Following arguments similar to those developed in [206, Ch. 35], the following array form for diffusion RLS can be motivated. Let

$$P_{k,i} \triangleq P_{k,i}^{1/2} \left(P_{k,i}^{1/2}\right)^* \tag{15.47}$$

denote the Cholesky factorization of $P_{k,i}$, where $P_{k,i}^{1/2}$ is lower-triangular with positive entries on its diagonal. Introduce further the scalar and vector quantities:

$$\gamma_{\ell k}(i) \triangleq 1/(1 + c_{\ell k} u_{\ell,i} P_{k,i} u_{\ell,i}^*) \tag{15.48}$$

$$g_{\ell k,i} \triangleq c_{\ell k}^{1/2} \gamma_{\ell k}(i) P_{k,i} u_{\ell,i}^* \tag{15.49}$$

Then, the updates in (15.40) can be rewritten as:

$$e_\ell(i) \leftarrow d_\ell(i) - u_{\ell,i}\psi_{k,i} \tag{15.50}$$

$$\psi_{k,i} \leftarrow \psi_{k,i} + c_{\ell k}^{1/2} \left[g_{\ell k,i}\gamma_{\ell k}^{-1/2}(i)\right] \left[\gamma_{\ell k}^{-1/2}(i)\right]^{-1} e_\ell(i) \tag{15.51}$$

$$P_{k,i} \leftarrow P_{k,i} - g_{\ell k,i} g_{\ell k,i}^* / \gamma_{\ell k}(i) \tag{15.52}$$

These updates can be implemented in array form as follows. We form the pre-array matrix:

$$D \triangleq \begin{bmatrix} 1 & 0_{1\times M} \\ c_{\ell k}^{1/2} \left(P_{k,i}^{1/2}\right)^* u_{\ell,i}^* & \left(P_{k,i}^{1/2}\right)^* \end{bmatrix} \tag{15.53}$$

where $P_{k,i}^{1/2}$ is the Cholesky factor of the matrix $P_{k,i}$ appearing on the right-hand side of (15.52). Next, we determine a unitary transformation, $\Theta_{\ell k,i}$, that transforms D into an upper-triangular form with positive entries on the diagonal. Specifically, we perform the QR factoriza-

tion of matrix D:

$$\underbrace{\begin{bmatrix} 1 & 0_{1\times M} \\ c_{\ell k}^{1/2}\left(P_{k,i}^{1/2}\right)^{*} u_{\ell,i}^{*} & \left(P_{k,i}^{1/2}\right)^{*} \end{bmatrix}}_{D} = \underbrace{\Theta_{\ell k,i}\begin{bmatrix} \gamma_{\ell k}^{-1/2}(i) & g_{\ell k,i}^{*}\gamma_{\ell k}^{-1/2}(i) \\ 0 & \left(P_{k,i}^{1/2}\right)^{*} \end{bmatrix}}_{QR} \tag{15.54}$$

where the resulting $P_{k,i}^{1/2}$ on the right-hand side of the above equation now refers to the Cholesky factor of the updated matrix $P_{k,i}$ appearing on the left-hand side of (15.52). The other quantities in the post-array (15.54) correspond to what is needed to perform the update (15.51). In summary, we arrive at the following array form.

Array form of diffusion RLS strategy (ATC)

step 1 (initialization by agent k)

$\psi_{k,i} \leftarrow w_{k,i-1}$

$P_{k,i}^{1/2} \leftarrow \lambda^{-1/2} P_{k,i-1}^{1/2}$

step 2 (adaptation)

Update $\{\psi_{k,i}, P_{k,i}^{1/2}\}$ by iterating over $\ell \in \mathcal{N}_k$:

$$\begin{bmatrix} \gamma_{\ell k}^{-1/2}(i) & g_{\ell k,i}^{*}\gamma_{\ell k}^{-1/2}(i) \\ 0 & \left(P_{k,i}^{1/2}\right)^{*} \end{bmatrix} \leftarrow \mathrm{QR}\left(\begin{bmatrix} 1 & 0_{1\times M} \\ c_{\ell k}^{1/2}\left(P_{k,i}^{1/2}\right)^{*} u_{\ell,i}^{*} & \left(P_{k,i}^{1/2}\right)^{*} \end{bmatrix}\right)$$

$e_{\ell}(i) \leftarrow d_{\ell,i} - u_{\ell,i}\psi_{k,i}$

$\psi_{k,i} \leftarrow \psi_{k,i} + c_{\ell k}^{1/2}\left[g_{\ell k,i}\gamma_{\ell k}^{-1/2}(i)\right]\left[\gamma_{\ell k}^{-1/2}(i)\right]^{-1} e_{\ell}(i)$

end

step 3 (combination)

$$w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\psi_{\ell,i}$$

(15.55)

We illustrate the operation of algorithm (15.55) numerically for the case of the averaging rule (11.148) for A and the Metropolis rule (8.100) for C. Figure 15.7 shows the connected network topology with $N = 20$ agents used for this simulation. Figure 15.8 plots the evolution of the ensemble-average learning curves, $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$, for the ATC LMS diffusion strategy (7.23) with uniform step-size $\mu_k = 0.005$ and for the array form of the RLS diffusion strategy (15.55) with $\delta = 1 \times 10^{-6}$

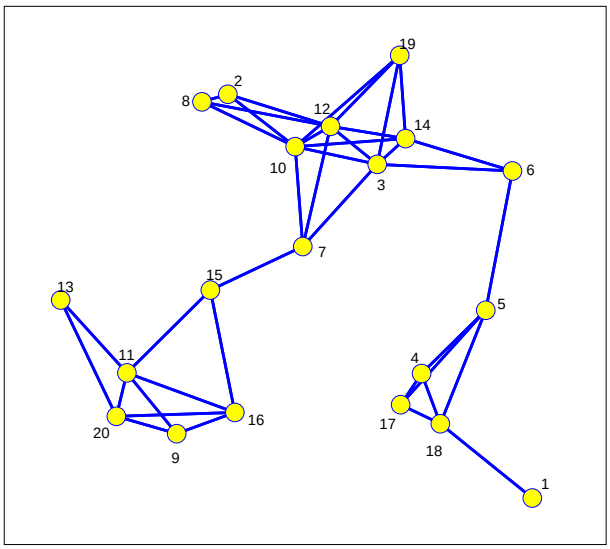

Figure 15.7: A connected network topology consisting of $N = 20$ agents employing the averaging rule (11.148) for A and the Metropolis rule (8.100) for C in the diffusion RLS implementation (15.55).

and $\lambda = 0.998$. The curves are obtained by averaging the trajectories $\{\frac{1}{N}\|\widetilde{\boldsymbol{w}}_i\|^2\}$ over 100 repeated experiments. The label on the vertical axes in the figures refer to the learning curve $\frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ by writing $\mathrm{MSD}_{\mathrm{dist,av}}(i)$, with an iteration index i. Each experiment involves running the algorithms on real-valued data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ generated according to the model $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i)$, with $M = 5$. The unknown vector w^o is generated randomly and its norm is normalized to one.

15.7 Distributed State-Space Estimation

Distributed strategies can also be applied to the solution of state-space filtering and smoothing problems [53, 54, 59, 61, 88, 112, 142, 181, 182]. Here, we describe briefly a diffusion version of the distributed Kalman filter. Thus, consider a network consisting of N agents observing the state vector, $\boldsymbol{x}_i$, of size $n \times 1$ of a linear state-space model. At every time i, every agent k collects a measurement vector $\boldsymbol{y}_{k,i}$ of size $p \times 1$,

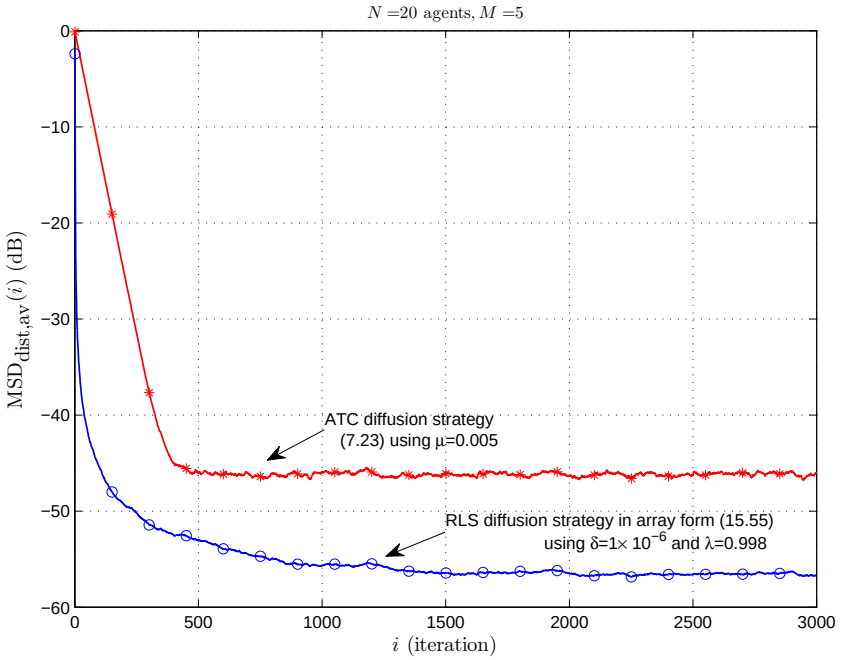

Figure 15.8: Evolution of the learning curves for ATC LMS diffusion (7.23) and RLS diffusion (15.55).

which is related to the state vector as follows:

$$\boldsymbol{x}_{i+1} = F_i \boldsymbol{x}_i + G_i \boldsymbol{n}_i \tag{15.56}$$

$$\boldsymbol{y}_{k,i} = H_{k,i} \boldsymbol{x}_i + \boldsymbol{v}_{k,i}, \quad k = 1, 2, \ldots, N \tag{15.57}$$

The signals $\boldsymbol{n}_i$ and $\boldsymbol{v}_{k,i}$ denote state and measurement noises of sizes $n \times 1$ and $p \times 1$, respectively, and they are assumed to be zero-mean, uncorrelated and white, with covariance matrices denoted by

$$\mathbb{E} \begin{bmatrix} \boldsymbol{n}_i \\ \boldsymbol{v}_{k,i} \end{bmatrix} \begin{bmatrix} \boldsymbol{n}_j \\ \boldsymbol{v}_{k,j} \end{bmatrix}^* \triangleq \begin{bmatrix} Q_i & 0 \\ 0 & R_{k,i} \end{bmatrix} \delta_{ij} \tag{15.58}$$

The initial state vector, $\boldsymbol{x}_o$, is assumed to have zero mean with

$$\mathbb{E}\, \boldsymbol{x}_o \boldsymbol{x}_o^* = \Pi_o > 0 \tag{15.59}$$

and is uncorrelated with $\boldsymbol{n}_i$ and $\boldsymbol{v}_{k,i}$, for all i and k. We further assume that $R_{k,i} > 0$. The parameter matrices $\{F_i, G_i, H_{k,i}, Q_i, R_{k,i}, \Pi_o\}$ are

assumed to be known by node k. Let $\widehat{\boldsymbol{x}}_{k,i|j}$ denote a local estimator for $\boldsymbol{x}_i$ that is computed by agent k at time i based on local observations and on neighborhood data up to time j. The following diffusion strategy was developed in [54, 59, 61] to approximate predicted and filtered versions of these local estimators in a distributed manner for data satisfying model (15.56)–(15.59). For every agent k, we start with $\widehat{\boldsymbol{x}}_{k,0|-1} = 0$ and $P_{k,0|-1} = \Pi_o$, where $P_{k,0|-1}$ is an $M \times M$ matrix. At every time instant i, every agent k performs the calculations listed in (15.60). In this implementation, the combination policy $A = [a_{\ell k}]$ consists of nonnegative scalar coefficients and is left-stochastic. It was argued in Eq. (17) in [54] that, in general, an enhanced fusion of the local estimators $\{\boldsymbol{\psi}_{\ell,i}\}$ can be attained by employing convex-combination coefficients defined in terms of certain inverse matrices, $\{P^{-1}_{\ell,i|i}\}$. This construction, however, would entail added computational cost and require the sharing of additional information regarding the inverses $\{P^{-1}_{\ell,i|i}\}$. The implementation (15.60) shown below from [54] employs scalar combination coefficients $\{a_{\ell k}\}$ in order to reduce the complexity of the resulting algorithm. Reference [117] studies the alternative fusion of the estimators $\{\boldsymbol{\psi}_{\ell,i}\}$ in the diffusion Kalman filter by exploiting information about the inverses $\{P^{-1}_{\ell,i|i}\}$.

Time and measurement-form of diffusion Kalman filter

step 1 (initialization by agent k)

$$
\begin{aligned}
&\boldsymbol{\psi}_{k,i} \leftarrow \widehat{\boldsymbol{x}}_{k,i|i-1} \\
&P_{k,i} \leftarrow P_{k,i|i-1}
\end{aligned}
$$

step 2 (adaptation)

Update $\{\boldsymbol{\psi}_{k,i}, P_{k,i}\}$ by iterating over $\ell \in \mathcal{N}_k$:

$$
\begin{aligned}
&R_e \leftarrow R_{\ell,i} + H_{\ell,i} P_{k,i} H^*_{\ell,i} \\
&\boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k,i} + P_{k,i} H^*_{\ell,i} R_e^{-1} \left(\boldsymbol{y}_{\ell,i} - H_{\ell,i} \boldsymbol{\psi}_{k,i}\right) \\
&P_{k,i} \leftarrow P_{k,i} - P_{k,i} H^*_{\ell,i} R_e^{-1} H_{\ell,i} P_{k,i}
\end{aligned}
\tag{15.60}
$$

end

step 3 (combination)

$$
\begin{aligned}
&\widehat{\boldsymbol{x}}_{k,i|i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i} \\
&P_{k,i|i} = P_{k,i} \\
&\widehat{\boldsymbol{x}}_{k,i+1|i} = F_i \widehat{\boldsymbol{x}}_{k,i|i} \\
&P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*
\end{aligned}
$$

An alternative representation for the above diffusion Kalman filter may be obtained in information form by assuming that $P_{k,i|i-1} > 0$ for all k and i. For every agent k, we start with $\widehat{\boldsymbol{x}}_{k,0|-1} = 0$ and $P^{-1}_{k,0|-1} = \Pi_o^{-1}$. At every time instant i, every agent k then performs the calculations listed in (15.61).

Information form of the diffusion Kalman filter

step 1 (adaptation)

$$
\begin{aligned}
&S_{k,i} = \sum_{\ell\in\mathcal{N}_k} H^*_{\ell,i} R^{-1}_{\ell,i} H_{\ell,i} \\
&\boldsymbol{q}_{k,i} = \sum_{\ell\in\mathcal{N}_k} H^*_{\ell,i} R^{-1}_{\ell,i} \boldsymbol{y}_{\ell,i} \\
&P^{-1}_{k,i|i} = P^{-1}_{k,i|i-1} + S_{k,i} \\
&\boldsymbol{\psi}_{k,i} = \widehat{\boldsymbol{x}}_{k,i|i-1} + P_{k,i|i}\left(\boldsymbol{q}_{k,i} - S_{k,i}\widehat{\boldsymbol{x}}_{k,i|i-1}\right)
\end{aligned}
\tag{15.61}
$$

step 2: (combination)

$$
\begin{aligned}
&\widehat{\boldsymbol{x}}_{k,i|i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell,i} \\
&\widehat{\boldsymbol{x}}_{k,i+1|i} = F_i\widehat{\boldsymbol{x}}_{k,i|i} \\
&P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*
\end{aligned}
$$

Step 1 in (15.61) is similar to the update used in the distributed Kalman filter derived in [181] using consensus-type arguments. One difference is that reference [181] starts from a continuous-time consensus implementation and discretizes it to arrive at the following update relation:

$$
\widehat{\boldsymbol{x}}_{k,i|i} \;=\; \boldsymbol{\psi}_{k,i} + \epsilon \sum_{\ell\in\mathcal{N}_k}(\boldsymbol{\psi}_{\ell,i} - \boldsymbol{\psi}_{k,i}) \tag{15.62}
$$

which, in order to facilitate comparison with (15.61), can be equivalently rewritten as:

$$
\widehat{\boldsymbol{x}}_{k,i|i} \;=\; (1+\epsilon-n_k\epsilon)\,\boldsymbol{\psi}_{k,i} \;+\; \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} \epsilon\,\boldsymbol{\psi}_{\ell,i} \tag{15.63}
$$

where n_k denotes the degree of agent k (i.e., the size of its neighborhood). In comparison, the diffusion step in (15.61) can be written as:

$$
\widehat{\boldsymbol{x}}_{k,i|i} \;=\; a_{kk}\,\boldsymbol{\psi}_{k,i} \;+\; \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \tag{15.64}
$$

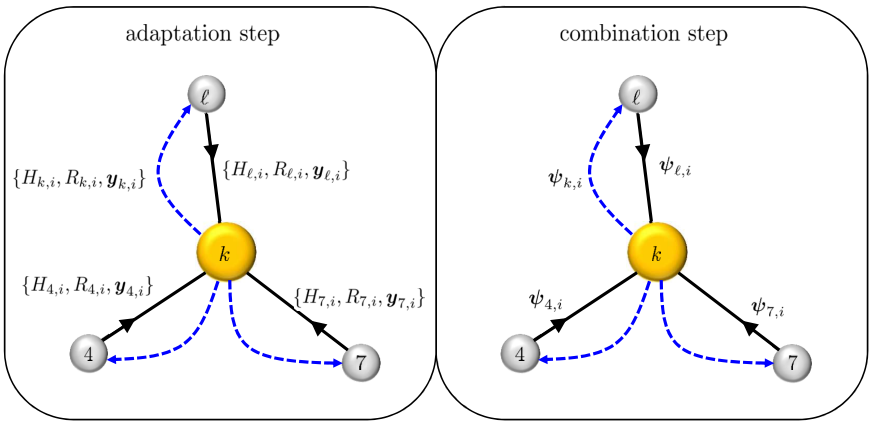

Figure 15.9: During the adaptation step 2 in the diffusion Kalman implementation (15.60), agents exchange data measurements and model measurements $\{H_{\ell,i}, R_{\ell,i}, \boldsymbol{y}_{\ell,i}\}$ (left). During the consultation step 3, agents exchange their intermediate iterates $\{\boldsymbol{\psi}_{\ell,i}\}$ (right).

Observe that the weights used in (15.63) are $(1+\epsilon-n_k\epsilon)$ for the agent's estimator, $\boldsymbol{\psi}_{k,i}$, and ϵ for all other estimators, $\{\boldsymbol{\psi}_{\ell,i}\}$, arriving from the neighbors of agent k. In comparison, the diffusion step (15.64) employs a convex combination of the estimators $\{\boldsymbol{\psi}_{\ell,i}\}$ with generally different weights $\{a_{\ell k}\}$ for different neighbors [53, 54].

Figure 15.9 illustrates the exchange of information that occurs during the adaptation and combination steps in the diffusion Kalman implementations (15.60) or (15.61). During the adaptation step, agents exchange data measurements and model parameters $\{H_{\ell,i}, R_{\ell,i}, \boldsymbol{y}_{\ell,i}\}$ with their neighbors, and during the consultation step agents exchange their intermediate iterates $\{\boldsymbol{\psi}_{\ell,i}\}$.

Example 15.1 (Tracking a projectile). We illustrate the operation of the diffusion and consensus Kalman filters numerically for the network shown in Figure 15.10 with the agents employing the averaging rule (11.148) in the diffusion case. We consider an application where each of the agents in the network is tracking a projectile — see Figure 15.11; each agent has access to noisy measurements of the $(x, y)-$coordinates of the projectile relative to a pre-defined coordinate system. We simulate two scenarios. In one case, the agents run the diffusion Kalman implementation (15.60) and in the second

case, the agents run the consensus implementation that would result form using (15.61) with the combination weights shown in (15.63) with $\epsilon = 0.001$.

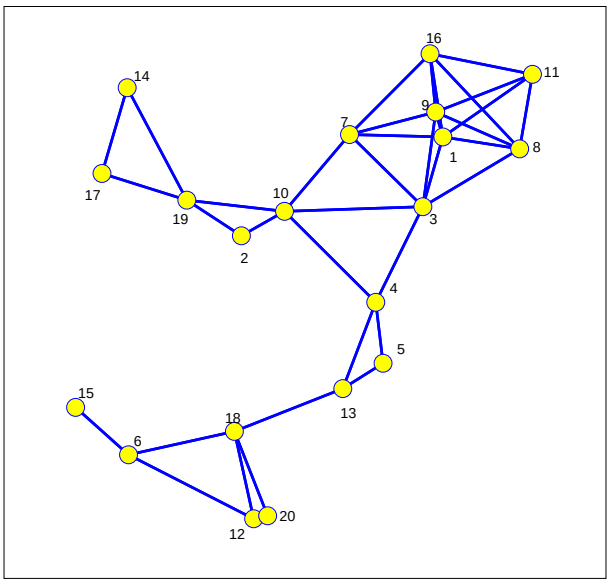

Figure 15.10: A connected network topology consisting of $N = 20$ agents employing the averaging rule (11.148).

We consider a simplified model and assume the target is moving within the plane $z = 0$. Referring to Figure 15.11, the target is launched from location (x_o, y_o) at an angle θ with the horizontal axis at an initial speed s. The initial velocity components along the horizontal and vertical directions are therefore:

$$s_x(0) = s\cos\theta, \quad s_y(0) = s\sin\theta \tag{15.65}$$

The motion of the object is governed by Newton's laws of motion; the acceleration along the vertical direction is downwards and its magnitude is given by $g \approx 10$ m/s^2. The motion along the horizontal direction is uniform (with zero acceleration) so that the horizontal velocity component is constant for all time instants and remains equal to s_x:

$$s_x(t) = s\cos\theta, \quad t \geq 0 \tag{15.66}$$

For the vertical direction, the velocity component satisfies the equation of motion:

$$s_y(t) = s\sin\theta - gt, \quad t \geq 0 \tag{15.67}$$

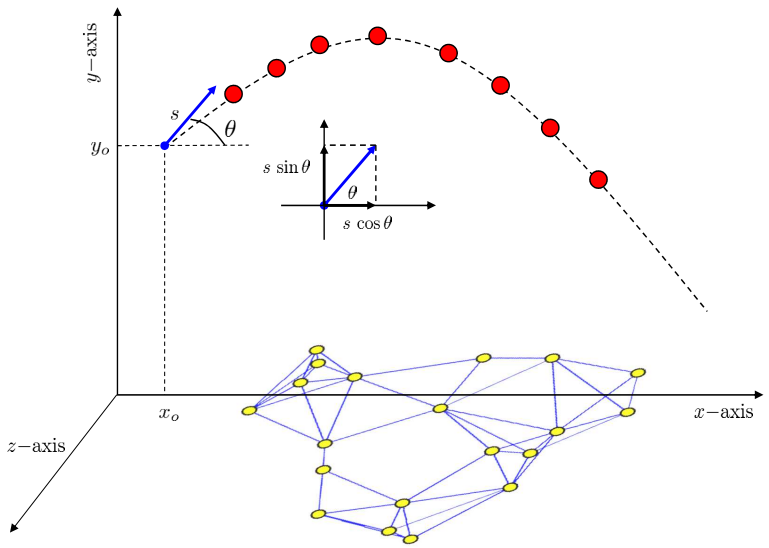

Figure 15.11: The object is launched from location (x_o, y_o) at an angle θ with the horizontal direction. Under idealized conditions, the trajectory is parabolic. Using noisy measurements of the target location $(x(t), y(t))$ by multiple agents, the objective is to estimate the actual trajectory of the object.

We denote the location coordinates of the object at any time t by $(x(t), y(t))$. These coordinates satisfy the differential equations

$$\frac{dx(t)}{dt} = s_x(t), \qquad \frac{dy(t)}{dt} = s_y(t) \tag{15.68}$$

We sample the equations of motion every T units of time and write

$$s_x(i) \triangleq s_x(iT) = s\cos\theta \tag{15.69}$$

$$s_y(i) \triangleq s_y(iT) = s\sin\theta - igT \tag{15.70}$$

$$x(i+1) = x(i) + Ts_x(i) \tag{15.71}$$

$$y(i+1) = y(i) + Ts_y(i) \tag{15.72}$$

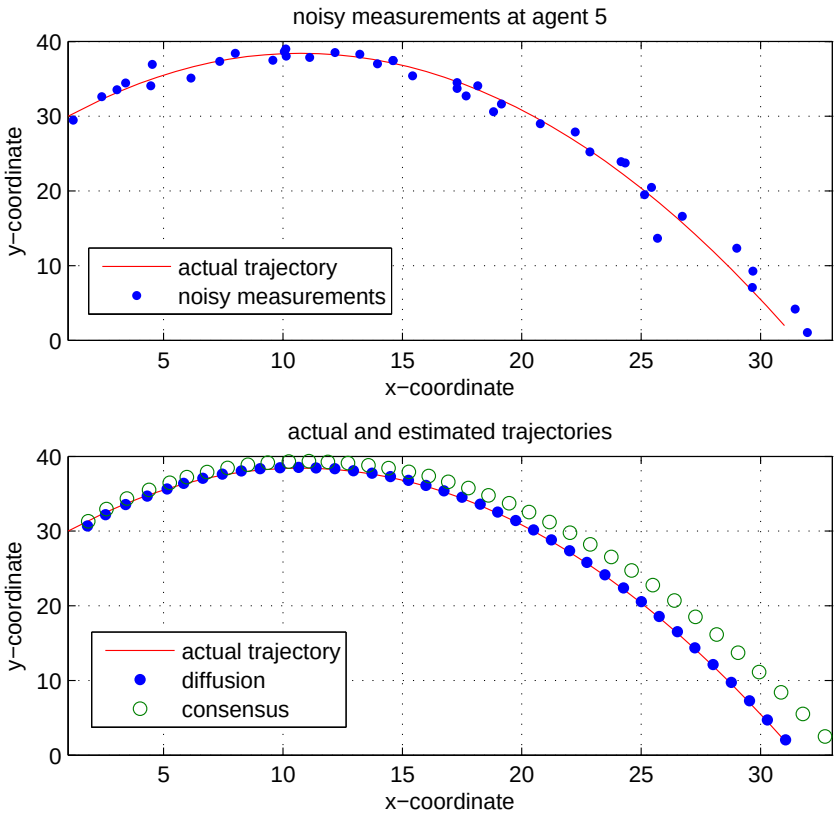

Figure 15.12: Estimated trajectories obtained by the diffusion Kalman implementation (15.60) and by the consensus implementation that results form using (15.61) with the combination weights shown in (15.63) with $\epsilon = 0.001$. The top plot shows the noisy measurements collected by one of the agents.

As such, the dynamics of the moving object can be approximated by the following discretized state-space equation:

$$\underbrace{\begin{bmatrix} x(i+1) \\ y(i+1) \\ s_x(i+1) \\ s_y(i+1) \end{bmatrix}}_{\boldsymbol{x}_{i+1}} = \underbrace{\begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} x(i) \\ y(i) \\ s_x(i) \\ s_y(i) \end{bmatrix}}_{\boldsymbol{x}_i} - \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} gT}_{d_i} \tag{15.73}$$

Note that the state vector $\boldsymbol{x}_i$ in this model involves four entries. Com-

pared with (15.56), we see that the state recursion in this case includes a deterministic driving term, d_i, and does not include process noise ($G_i = 0$); if desired, we may include a process noise term to model disturbances in the state evolution (such as errors arising from the discretization process). The deterministic driving term can be incorporated into the statement of the diffusion and consensus filters by modifying the update relation $\widehat{\boldsymbol{x}}_{k,i+1|i} = F_i\widehat{\boldsymbol{x}}_{k,i|i}$ that appears in the combination steps in the statements (15.60) and (15.61) by

$$\widehat{\boldsymbol{x}}_{k,i+1|i} \;=\; F_i\widehat{\boldsymbol{x}}_{k,i|i} \;+\; d_i \tag{15.74}$$

The tracking problem we are interested in is one that estimates and tracks the state vector $\boldsymbol{x}_i$ based on noisy measurements of the location coordinates of the object by networked agents.

We denote the measurement vector at each agent k at time i by $\boldsymbol{y}_{k,i}$ and it satisfies:

$$\boldsymbol{y}_{k,i} \;=\; \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{H} \begin{bmatrix} x(i) \\ y(i) \\ s_x(i) \\ s_y(i) \end{bmatrix} \;+\; \boldsymbol{v}_{k,i} \tag{15.75}$$

where $\boldsymbol{v}_{k,i}$ denotes a 2×1 zero-mean white noise process with covariance matrix assumed to be of the form $R_{k,i} = \sigma_{v,k}^2 I_2$. The variances $\{\sigma_{v,k}^2\}$ are selected randomly from within the interval $[0, 0.5]$. It is seen that the entries of the vector $\boldsymbol{y}_{k,i}$ are noisy measurements of the x and $y-$coordinates of the location of the moving object. We use the following values in the simulation:

$$\Pi_o = I_4, \;\; (x_o, y_o) = (1, 30), \;\; s = 15, \;\; T = 0.01, \;\; \theta = 60^o \tag{15.76}$$

Figure 15.12 plots the actual trajectory, the noisy measurements sampled by one of the agents, and the averaged recovered trajectory (averaged over all agents and over 100 experiments) by the diffusion network and by the consensus network.

■

Acknowledgements

This work was supported in part by the National Science Foundation under grants CCF-0942936 and CCF-1011918. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

A condensed overview article covering select material from this extended treatment appears in [207]. The author is grateful to IEEE for allowing reproduction of material from [207] in this work. The simulation figures in this work were generated using the MATLAB software, which is a registered trademark of MathWorks Inc., 24 Prime Park Way, Natick, MA 01760-1500.

The author is grateful to several of his current and former graduate students Jianshu Chen, Xiaochuan Zhao, Sheng-Yuan Tu, Zaid Towfic, Cassio G. Lopes, Federico S. Cattivelli, Bicheng Ying, Stefan Vlaski, Chung-Kai Yu, Ricardo Merched, and Vitor H. Nascimento for their insightful contributions and thoughtful feedback on earlier material and drafts for this manuscript.

The author is also grateful to students from his graduate level course at UCLA on *Inference over Networks* for their feedback on an earlier draft of these lecture notes. A list of assignment problems that complements these notes can be downloaded from the author's research group website at http://www.ee.ucla.edu/asl.

Appendices

A

Complex Gradient Vectors

Let $g(z)$ denote a scalar real or complex-valued function of a complex variable, z. The function $g(z)$ need not be holomorphic in the variable z and, therefore, it need not be differentiable in the traditional complex differentiation sense (cf. definition (A.3) further ahead). In many instances though, we are only interested in determining the locations of the stationary points of $g(z)$. For these cases, it is sufficient to rely on a different notion of differentiation, which we proceed to motivate following [3, 47, 107, 111, 116, 197, 206, 218, 251]. We start by defining complex gradient vectors in this appendix, followed by complex Hessian matrices in Appendix B. We also explain how the evaluation of gradient vectors and Hessian matrices gets simplified when the independent variable z happens to be real-valued. In the treatment that follows, we examine both situations when the variables $\{z, z^*\}$ are either scalar-valued or vector-valued.

A.1 Cauchy-Riemann Conditions

To motivate the alternative differentiation concept, we first review briefly the traditional definition of complex differentiation. Thus, as-

sume z is a scalar and let us express it in terms of its real and imaginary parts, denoted by x and y, respectively:

$$z \triangleq x + jy, \quad j \triangleq \sqrt{-1} \tag{A.1}$$

We can then interpret $g(z)$ as a two-dimensional function of the real variables $\{x, y\}$ and represent its real and imaginary parts as functions of these same variables, say, as $u(x, y)$ and $v(x, y)$:

$$g(z) \triangleq u(x, y) + jv(x, y) \tag{A.2}$$

We denote the traditional complex derivative of $g(z)$ with respect to z by $g'(z)$ and define it as the limit:

$$g'(z) \triangleq \lim_{\Delta z \to 0} \frac{g(z + \Delta z) - g(z)}{\Delta z} \tag{A.3}$$

or, more explicitly,

$$g'(z) = \lim_{\Delta z \to 0} \frac{g(x + \Delta x, y + \Delta y) - g(x, y)}{\Delta x + j\Delta y} \tag{A.4}$$

where we are writing $\Delta z = \Delta x + j\Delta y$. For $g(z)$ to be differentiable at location z, in which case it is also said to be *holomorphic* at z, then the above limit needs to exist regardless of the direction from which $z + \Delta z$ approaches z. In particular, if we set $\Delta y = 0$ and let $\Delta x \to 0$, then the above definition gives that $g'(z)$ should be equal to

$$g'(z) = \frac{\partial u(x, y)}{\partial x} + j\frac{\partial v(x, y)}{\partial x} \tag{A.5}$$

On the other hand, if we set $\Delta x = 0$ and let $\Delta y \to 0$ so that $\Delta z = j\Delta y$, then the definition gives that the same $g'(z)$ should be equal to

$$g'(z) = \frac{\partial v(x, y)}{\partial y} - j\frac{\partial u(x, y)}{\partial y} \tag{A.6}$$

Expressions (A.5) and (A.6) must coincide, which means that the real and imaginary parts of $g(z)$ should satisfy the conditions:

$$\begin{cases} \dfrac{\partial u(x, y)}{\partial x} = \dfrac{\partial v(x, y)}{\partial y} \\ \dfrac{\partial u(x, y)}{\partial y} = -\dfrac{\partial v(x, y)}{\partial x} \end{cases} \tag{A.7}$$

These are known as the *Cauchy-Riemann* conditions [5, 197]. It can be shown that these conditions are not only necessary for a complex function $g(z)$ to be differentiable at location z, but if the partial derivatives of $u(x, y)$ and $v(x, y)$ are continuous, then they are also sufficient.

Example A.1 (Real-valued functions). Consider the quadratic function $g(z) = |z|^2$. It is straightforward to verify that $g(x, y) = x^2 + y^2$ so that

$$u(x, y) = x^2 + y^2, \qquad v(x, y) = 0 \tag{A.8}$$

Therefore, the Cauchy-Riemann conditions (A.7) are not satisfied in this case (except at the point $x = y = 0$). More generally, it is straightforward to verify that any other (nonconstant) real-valued function, $g(z)$, cannot satisfy (A.7) except possibly at some locations. It turns out though that real-valued cost functions of this form are commonplace in problems involving estimation, adaptation, and learning. Fortunately, in these applications, we are rarely interested in evaluating the traditional complex derivative of $g(z)$. Instead, we are more interested in determining the location of the stationary points of $g(z)$. To do so, it is sufficient to rely on a different notion of differentiation based on what is sometimes known as the Wirtinger calculus [47, 251, 264], which we describe next.

■

A.2 Scalar Arguments

We continue with the case in which $z \in \mathbb{C}$ is a scalar and allow $g(z)$ to be real or complex-valued so that $g(z) \in \mathbb{C}$. We again express z in terms of its real and imaginary parts as in (A.1), and similarly express $g(z)$ as a function of both x and y, i.e., as $g(x, y)$. The (Wirtinger) partial derivatives of $g(z)$ with respect to the complex arguments z and z^*, which we shall also refer to as the complex gradients of $g(z)$, are defined in terms of the partial derivatives of $g(x, y)$ with respect to the real arguments x and y as follows:

$$\begin{cases} \dfrac{\partial g(z)}{\partial z} & \triangleq & \dfrac{1}{2}\left\{\dfrac{\partial g(x,y)}{\partial x} - j\,\dfrac{\partial g(x,y)}{\partial y}\right\} \\ \dfrac{\partial g(z)}{\partial z^*} & \triangleq & \dfrac{1}{2}\left\{\dfrac{\partial g(x,y)}{\partial x} + j\,\dfrac{\partial g(x,y)}{\partial y}\right\} \end{cases} \tag{A.9}$$

The above expressions can be grouped together in vector form as:

$$\begin{bmatrix} \partial g(z)/\partial z \\ \\ \partial g(z)/\partial z^* \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix}\begin{bmatrix} \partial g(x,y)/\partial x \\ \\ \partial g(x,y)/\partial y \end{bmatrix} \tag{A.10}$$

so that, by inversion, it also holds that

$$\begin{bmatrix} \partial g(x,y)/\partial x \\ \\ \partial g(x,y)/\partial y \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix}\begin{bmatrix} \partial g(z)/\partial z \\ \\ \partial g(z)/\partial z^* \end{bmatrix} \tag{A.11}$$

The reason why the partial derivatives (A.9) are useful can be readily seen when $g(z)$ is real-valued, namely, $g(z) \in \mathbb{R}$. In that case, and by definition, a point $z^o = x^o + jy^o$ is said to be a stationary point of $g(z)$ if, and only if, (x^o, y^o) is a stationary point of $g(x, y)$. The latter condition is equivalent to requiring

$$\frac{\partial g(x,y)}{\partial x}\bigg|_{x=x^o, y=y^o} = 0 \tag{A.12}$$

and

$$\frac{\partial g(x,y)}{\partial y}\bigg|_{x=x^o, y=y^o} = 0 \tag{A.13}$$

These two conditions combined are turn is equivalent to the following single condition in terms of the complex gradient vector:

$$\frac{\partial g(z)}{\partial z}\bigg|_{z=z^o} = 0 \tag{A.14}$$

In this way, either of the partial derivatives defined by (A.9) enable us to locate stationary points of the real-valued function $g(z)$. Note that we are using the superscript notation "^{o}", as in z^o, to refer to stationary points.

Example A.2 (Wirtinger complex differentiation)**.** We illustrate the definition of the partial derivatives (A.9) by considering a few examples. We will observe from the results in these examples that (Wirtinger) complex differentiation with respect to z treats z^* as a constant and, similarly, complex differentiation with respect to z^* treats z as a constant:

(1) Let $g(z) = z^2$. Then, $g(x, y) = (x^2 - y^2) + j2xy$ so that from (A.9):

$$\frac{\partial g(z)}{\partial z} = \frac{1}{2}(4x + j4y) = 2z, \qquad \frac{\partial g(z)}{\partial z^*} = 0 \tag{A.15}$$

(2) Let $g(z) = |z|^2$. Then, $g(x, y) = x^2 + y^2$ and

$$\frac{\partial g(z)}{\partial z} = (x - jy) = z^*, \quad \frac{\partial g(z)}{\partial z^*} = (x + jy) = z \tag{A.16}$$

(3) Let $g(z) = \kappa + \alpha z + \beta z^* + \gamma |z|^2$, where $(\kappa, \alpha, \beta, \gamma)$ are scalar constants. Then,

$$\frac{\partial g(z)}{\partial z} = \alpha + \gamma z^*, \quad \frac{\partial g(z)}{\partial z^*} = \beta + \gamma z \tag{A.17}$$

■

A.3 Vector Arguments

We consider next the case in which z is a *column* vector argument, say, of size $M \times 1$, and whose individual entries are denoted by $\{z_m\}$, i.e.,

$$z = \mathrm{col}\{z_1, z_2, \ldots, z_M\} \in \mathbb{C}^M \tag{A.18}$$

We continue to allow $g(z)$ to be real or complex-valued so that $g(z) \in \mathbb{C}$. The (Wirtinger) partial derivative of $g(z)$ with respect to z is again denoted by $\partial\, g(z)/\partial z$ and is defined as the *row* vector:

$$\frac{\partial g(z)}{\partial z} \triangleq \left[\begin{array}{cccc} \dfrac{\partial g}{\partial z_1} & \dfrac{\partial g}{\partial z_2} & \cdots & \dfrac{\partial g}{\partial z_M} \end{array} \right], \quad \left\{ \begin{array}{l} z \text{ is a column} \\ \partial g/\partial z \text{ is a row} \end{array} \right. \tag{A.19}$$

in terms of the individual (Wirtinger) partial derivatives $\{\partial g/\partial z_m\}$. Expression (A.19) for $\partial\, g(z)/\partial z$ is also known as the *Jacobian* of $g(z)$. We shall refer to (A.19) as the complex gradient of $g(z)$ with respect to z and denote it more frequently by the alternative notation $\nabla_z\, g(z)$, i.e.,

$$\nabla_z\, g(z) \triangleq \left[\begin{array}{cccc} \dfrac{\partial g}{\partial z_1} & \dfrac{\partial g}{\partial z_2} & \cdots & \dfrac{\partial g}{\partial z_M} \end{array} \right], \quad \left\{ \begin{array}{l} z \text{ is a column} \\ \nabla_z\, g(z) \text{ is a row} \end{array} \right. \tag{A.20}$$

There is not a clear convention in the literature on whether the gradient vector relative to z should be defined as a row vector (as in (A.20)) or as a column vector; both choices are common and both choices are useful. We prefer to use the *row* convention (A.20) because it leads to differentiation results that are consistent with what we are familiar with from the rules of traditional differentiation in the real domain — see Example A.3 below. This is largely a matter of convenience.

Likewise, along with (A.20), we define the complex gradient of $g(z)$ with respect to z^* to be the *column* vector:

$$\nabla_{z^*} g(z) \triangleq \begin{bmatrix} \partial g/\partial z_1^* \\ \partial g/\partial z_2^* \\ \vdots \\ \partial g/\partial z_M^* \end{bmatrix} \equiv \frac{\partial g(z)}{\partial z^*}, \qquad \begin{cases} z^* \text{ is a row} \\ \nabla_{z^*} g(z) \text{ is a column} \end{cases} \tag{A.21}$$

Observe again the useful conclusion that when $g(z)$ is real-valued, then a vector $z^o = x^o + jy^o$ is a stationary point of $g(z)$ if, and only if,

$$\nabla_z g(z)|_{z=z^o} = 0 \tag{A.22}$$

Example A.3 (Complex gradients). Let us again consider a few examples:

(1) Let $g(z) = a^*z$, where $\{a, z\}$ are column vectors. Then,

$$\nabla_z g(z) = a^*, \quad \nabla_{z^*} g(z) = 0 \tag{A.23}$$

(2) Let $g(z) = \|z\|^2 = z^*z$, where z is a column vector. Then,

$$\nabla_z g(z) = z^*, \quad \nabla_{z^*} g(z) = z \tag{A.24}$$

(3) Let $g(z) = \kappa + a^*z + z^*b + z^*Cz$, where κ is a scalar, $\{a, b\}$ are column vectors, and C is a matrix. Then,

$$\nabla_z g(z) = a^* + z^*C, \quad \nabla_{z^*} g(z) = b + Cz \tag{A.25}$$

■

A.4 Real Arguments

When $z \in \mathbb{R}^M$ is real-valued and the function $g(z) \in \mathbb{R}$ is real-valued as well, the gradient vector is still defined as the row vector:

$$\nabla_z g(z) \triangleq \left[\begin{array}{cccc} \dfrac{\partial g}{\partial z_1} & \dfrac{\partial g}{\partial z_2} & \cdots & \dfrac{\partial g}{\partial z_M} \end{array} \right], \quad \left\{ \begin{array}{l} z \text{ is a column} \\ \nabla_z g(z) \text{ is a row} \end{array} \right. \tag{A.26}$$

in terms of the traditional partial derivatives of $g(z)$ with respect to the real scalar arguments $\{z_m\}$. Likewise, and in a manner that is consistent with (A.21), we define the gradient vector of $g(z)$ with respect to z^{T} to be the following *column* vector:

$$\nabla_{z^{\mathsf{T}}} g(z) \triangleq \left[\begin{array}{c} \partial g/\partial z_1 \\ \partial g/\partial z_2 \\ \vdots \\ \partial g/\partial z_M \end{array} \right], \quad \left\{ \begin{array}{l} z^{\mathsf{T}} \text{ is a row} \\ \nabla_{z^{\mathsf{T}}} g(z) \text{ is a column} \end{array} \right. \tag{A.27}$$

In particular, note the useful relation

$$\nabla_{z^{\mathsf{T}}} g(z) = [\nabla_z g(z)]^{\mathsf{T}} \tag{A.28}$$

This relation holds for both cases when z itself is real-valued or complex-valued.

Example A.4 (Quadratic cost functions I). Consider the quadratic function

$$g(z) = \kappa + a^{\mathsf{T}} z + z^{\mathsf{T}} b + z^{\mathsf{T}} C z \tag{A.29}$$

where κ is a scalar, $\{a, b\}$ are column vectors of dimension $M \times 1$ each, and C is an $M \times M$ *symmetric* matrix (all of them are real-valued in this case). Then, it can be easily verified that

$$\nabla_z g(z) = a^{\mathsf{T}} + b^{\mathsf{T}} + 2z^{\mathsf{T}} C \tag{A.30}$$

The reason for the additional factor of two in the rightmost term can be justified by carrying out the calculation of the gradient vector explicitly. Indeed, if we denote the individual entries of $\{a, b, z, C\}$ by $\{a_m, b_m, z_m, C_{mn}\}$, then

$$g(z) = \kappa + \sum_{m=1}^{M} (a_m + b_m) z_m + \sum_{m=1}^{M} \sum_{n=1}^{M} z_m C_{mn} z_n \tag{A.31}$$

so that

$$\begin{aligned}\frac{\partial g(z)}{\partial z_m} &= (a_m + b_m) + 2C_{mm}z_m + \sum_{n\neq m}^{M}(C_{mn} + C_{nm})z_n \\ &= (a_m + b_m) + 2\sum_{n=1}^{M} C_{nm}z_n \end{aligned} \tag{A.32}$$

where we used the fact that C is symmetric and, hence, $C_{mn} = C_{nm}$. Collecting all the partial derivatives into the gradient vector defined by (A.26) we arrive at (A.30).

Observe that while in the complex case, the arguments z and z^* are treated independently of each other during differentiation, this is not the case for the arguments z and z^{T} in the real case. In particular, since we can express the inner product $z^{\mathsf{T}}b$ as $b^{\mathsf{T}}z$, then the derivative of $z^{\mathsf{T}}b$ with respect to z is equal to the derivative of $b^{\mathsf{T}}z$ with respect to z (which explains the appearance of the term b^{T} in (A.30)).

■

Example A.5 (Quadratic cost functions II)**.** Consider the same quadratic function (A.29) with the only difference being that C is now arbitrary and *not* necessarily symmetric. Then, the same argument from Example A.4 will show that:

$$\nabla_z g(z) = a^{\mathsf{T}} + b^{\mathsf{T}} + z^{\mathsf{T}}(C + C^{\mathsf{T}}) \tag{A.33}$$

where $2C$ in (A.30) is replaced by $C + C^{\mathsf{T}}$.

■

B

Complex Hessian Matrices

Hessian matrices involve second-order partial derivatives, which we shall assume to be continuous functions of their arguments whenever necessary. Some effort is needed to define Hessian matrices for functions of complex variables. For this reason, we consider first the case of real arguments to help motivate the extension to complex arguments. In this appendix we only consider *real-valued* functions $g(z) \in \mathbb{R}$, which corresponds to the situation of most interest to us since utility or cost functions in adaptation and learning are generally real-valued.

B.1 Hessian Matrices for Real Arguments

We continue to denote the individual entries of the column vector $z \in \mathbb{R}^M$ by $z = \mathrm{col}\{z_1, z_2, \ldots, z_M\}$. The Hessian matrix of $g(z) \in \mathbb{R}$ is an $M \times M$ *symmetric* matrix function of z, denoted by $H(z)$, and whose $(m,n)-$th entry is constructed as follows:

$$[H(z)]_{m,n} \triangleq \frac{\partial^2 g(z)}{\partial z_m \partial z_n} = \frac{\partial}{\partial z_m}\left[\frac{\partial g(z)}{\partial z_n}\right] = \frac{\partial}{\partial z_n}\left[\frac{\partial g(z)}{\partial z_m}\right] \tag{B.1}$$

in terms of the partial derivatives of $g(z)$ with respect to the real scalar arguments $\{z_m, z_n\}$. For example, for a two-dimensional argument z

(i.e., $M = 2$), the four entries of the 2×2 Hessian matrix would be given by:

$$H(z) \;=\; \begin{bmatrix} \dfrac{\partial^2 g(z)}{\partial z_1^2} & \dfrac{\partial^2 g(z)}{\partial z_1 \partial z_2} \\ \dfrac{\partial^2 g(z)}{\partial z_2 \partial z_1} & \dfrac{\partial^2 g(z)}{\partial z_2^2} \end{bmatrix} \tag{B.2}$$

It is straightforward to recognize that the Hessian matrix $H(z)$ defined by (B.1) can be obtained as the result of two successive gradient vector calculations with respect to z and z^{T} in the following manner (where the order of the differentiation does not matter):

$$H(z) \;\triangleq\; \nabla_{z^{\mathsf{T}}} \left[\nabla_z \, g(z)\right] \;=\; \nabla_z \left[\nabla_{z^{\mathsf{T}}} \, g(z)\right] \qquad (M \times M) \tag{B.3}$$

For instance, using the first expression, the gradient operation $\nabla_z \, g(z)$ generates a $1 \times M$ (row) vector function and the subsequent differentiation with respect to z^{T} leads to the $M \times M$ Hessian matrix, $H(z)$. It is clear from (B.3) that the Hessian matrix is indeed symmetric so that

$$H(z) \;=\; H^{\mathsf{T}}(z) \tag{B.4}$$

A useful property of Hessian matrices is that they help characterize the nature of stationary points of functions $g(z)$ that are twice continuously differentiable. Specifically, if z^o is a stationary point of $g(z)$ (i.e., a point where $\nabla_z \, g(z) = 0$), then the following facts hold (see, e.g., [36, 93]):

(a) z^o is a local minimum of $g(z)$ if $H(z^o) > 0$, i.e., if all eigenvalues of $H(z^o)$ are positive.

(b) z^o is a local maximum of $g(z)$ if $H(z^o) < 0$, i.e., if all eigenvalues of $H(z^o)$ are negative.

Example B.1 (Quadratic cost functions). Consider the quadratic function

$$g(z) = \kappa + a^{\mathsf{T}} z + z^{\mathsf{T}} b + z^{\mathsf{T}} C z \tag{B.5}$$

where κ is a scalar, $\{a, b\}$ are column vectors of dimension $M \times 1$ each, and C is an $M \times M$ *symmetric* matrix (all of them are real-valued in this case).

We know from (A.22) and (A.30) that any stationary point, z^o, of $g(z)$ should satisfy the linear system of equations

$$Cz^o = \frac{1}{2}(a+b) \tag{B.6}$$

It follows that z^o is unique if, and only if, C is nonsingular. Moreover, in this case, the Hessian matrix is given by

$$H = 2C \tag{B.7}$$

which is independent of z. It follows that the quadratic function $g(z)$ will have a *unique* global minimum if, and only if, $C > 0$.

■

B.2 Hessian Matrices for Complex Arguments

We now extend the definition of Hessian matrices to functions $g(z) \in \mathbb{R}$ that are still *real-valued* but their argument, $z \in \mathbb{C}^M$, is complex-valued. This case is of great interest in adaptation, learning, and estimation problems since cost functions are generally real-valued while their arguments can be complex-valued. The Hessian matrix of $g(z)$ can now be defined in two equivalent forms by working either with the complex variables $\{z, z^*\}$ directly or with the real and imaginary parts $\{x, y\}$ of z. In contrast to the case of real arguments studied above in (B.3), where the Hessian matrix had dimensions $M \times M$, the Hessian matrix for complex arguments will be twice as large and will have dimensions $2M \times 2M$ for the reasons explained below.

We start by expressing each entry z_m of z in terms of its real and imaginary components as

$$z_m = x_m + jy_m, \quad m = 1, 2, \ldots, M \tag{B.8}$$

We subsequently collect the real and imaginary factors $\{x_m\}$ and $\{y_m\}$ into two real vectors:

$$x \triangleq \mathrm{col}\{x_1, x_2, \ldots, x_M\} \tag{B.9}$$

$$y \triangleq \mathrm{col}\{y_1, y_2, \ldots, y_M\} \tag{B.10}$$

so that

$$z = x + jy \tag{B.11}$$

Then, we can equivalently express $g(z)$ as a function of $2M$ *real* variables as $g(z) = g(x, y)$. We now proceed to define the Hessian matrix of $g(z)$ in two equivalent ways by working with either the complex variables $\{z, z^*\}$ or the real variables $\{x, y\}$. We consider the latter case first since we can then call upon the earlier definition (B.3) for real arguments.

B.2.1 First Possibility: Real Hessian Matrix

Since $g(x, y) \in \mathbb{R}$ is a function of the real arguments $\{x, y\}$, we can invoke definition (B.3) to associate with $g(x, y)$ a *real* Hessian matrix $H(x, y)$; its dimensions will be $2M \times 2M$. This Hessian matrix will involve second-order partial derivatives relative to x and y. For example, when $z = x + jy$ is a *scalar*, then $H(x, y)$ will be 2×2 and given by:

$$H(x,y) = \begin{bmatrix} \dfrac{\partial^2 g(x,y)}{\partial x^2} & \dfrac{\partial^2 g(x,y)}{\partial x \partial y} \\ \dfrac{\partial^2 g(x,y)}{\partial y \partial x} & \dfrac{\partial^2 g(x,y)}{\partial y^2} \end{bmatrix}, \quad z = x + jy \tag{B.12}$$

Likewise, when z is two-dimensional (i.e., $M = 2$) with entries $z_1 = x_1 + jy_1$ and $z_2 = x_2 + jy_2$, then the Hessian matrix of $g(z)$ will be 4×4 and given by:

$$H(x,y) = \left[\begin{array}{cc|cc} \dfrac{\partial^2 g(z)}{\partial x_1^2} & \dfrac{\partial^2 g(z)}{\partial x_1 \partial x_2} & \dfrac{\partial^2 g(z)}{\partial x_1 \partial y_1} & \dfrac{\partial^2 g(z)}{\partial x_1 \partial y_2} \\ \dfrac{\partial^2 g(z)}{\partial x_2 \partial x_1} & \dfrac{\partial^2 g(z)}{\partial x_2^2} & \dfrac{\partial^2 g(z)}{\partial x_2 \partial y_1} & \dfrac{\partial^2 g(z)}{\partial x_2 \partial y_2} \\ \hline \dfrac{\partial^2 g(z)}{\partial y_1 \partial x_1} & \dfrac{\partial^2 g(z)}{\partial y_1 \partial x_2} & \dfrac{\partial^2 g(z)}{\partial y_1^2} & \dfrac{\partial^2 g(z)}{\partial y_1 \partial y_2} \\ \dfrac{\partial^2 g(z)}{\partial y_2 \partial x_1} & \dfrac{\partial^2 g(z)}{\partial y_2 \partial x_2} & \dfrac{\partial^2 g(z)}{\partial y_2 \partial y_1} & \dfrac{\partial^2 g(z)}{\partial y_2^2} \end{array}\right] \tag{B.13}$$

More generally, for arguments $z = x+jy$ of arbitrary dimensions $M\times 1$, the real Hessian matrix of $g(z)$ can be expressed in partitioned form in terms of 4 sub-matrices of size $M \times M$ each:

$$\begin{aligned} H(x,y) &= \left[\begin{array}{c|c} \nabla_{x^{\mathsf{T}}}[\nabla_x\, g(x,y)] & \nabla_{x^{\mathsf{T}}}[\nabla_y\, g(x,y)] \\ \hline \nabla_{y^{\mathsf{T}}}[\nabla_x\, g(x,y)] & \nabla_{y^{\mathsf{T}}}[\nabla_y\, g(x,y)] \end{array}\right] \\ &\triangleq \left[\begin{array}{c|c} H_{x^{\mathsf{T}}x} & \left(H_{y^{\mathsf{T}}x}\right)^{\mathsf{T}} \\ \hline H_{y^{\mathsf{T}}x} & H_{y^{\mathsf{T}}y} \end{array}\right] \end{aligned} \tag{B.14}$$

where we introduced the compact notation $\{H_{x^{\mathsf{T}}x}, H_{y^{\mathsf{T}}y}, H_{y^{\mathsf{T}}x}\}$ to denote the following second-order differentiation operations relative to the variables x and y:

$$\left\{\begin{array}{lcl} H_{x^{\mathsf{T}}x} & \triangleq & \nabla_{x^{\mathsf{T}}}[\nabla_x\, g(x,y)] \\ H_{y^{\mathsf{T}}y} & \triangleq & \nabla_{y^{\mathsf{T}}}[\nabla_y\, g(x,y)] \\ H_{y^{\mathsf{T}}x} & \triangleq & \nabla_{y^{\mathsf{T}}}[\nabla_x\, g(x,y)] \end{array}\right. \tag{B.15}$$

We can express result (B.14) more compactly by working with the $2M \times 1$ extended vector v that is obtained by stacking x and y into a single vector:

$$v \triangleq \mathrm{col}\{x, y\} \tag{B.16}$$

Then, the function $g(z)$ can also be regarded as a function of v, namely, $g(v)$. It is straightforward to verify that the same Hessian matrix $H(x,y)$ given by (B.14) can be expressed in terms of differentiation of $g(v)$ with respect to v as follows (compare with (B.3)):

$$H(v) \triangleq \nabla_{v^{\mathsf{T}}}\left[\nabla_v\, g(v)\right] = \nabla_v\left[\nabla_{v^{\mathsf{T}}}\, g(v)\right] = H(x,y) \qquad (2M\times 2M) \tag{B.17}$$

We shall use the alternative representation $H(v)$ more frequently than $H(x,y)$ and refer to it as the *real* Hessian matrix. It is clear from expressions (B.14) or (B.17) that the Hessian matrix so defined is symmetric so that

$$H(v) = H^{\mathsf{T}}(v) \tag{B.18}$$

Again, a useful property of the Hessian matrix is that it can be used to characterize the nature of stationary points of functions $g(z)$ that are twice continuously differentiable. Specifically, if $z^o = x^o + jy^o$ is a stationary point of $g(z)$ (i.e., a point where $\nabla_z g(z) = 0$), then the following facts hold for $v^o = \text{col}\{x^o, y^o\}$:

(a) z^o is a local minimum of $g(z)$ if $H(v^o) > 0$, i.e., all eigenvalues of $H(v^o)$ are positive.

(b) z^o is a local maximum of $g(z)$ if $H(v^o) < 0$, i.e., all eigenvalues of $H(v^o)$ are negative.

B.2.2 Second Possibility: Complex Hessian Matrix

Besides $H(v)$, we can associate a second Hessian matrix representation with $g(z)$ by working directly with the complex variables z and z^* rather than their real and imaginary parts, x and y (or v). We refer to this second representation as the *complex* Hessian matrix and we denote it by $H_c(z)$, with the subscript "c" used to distinguish it from the real Hessian matrix, $H(v)$, defined by (B.17). The complex Hessian, $H_c(z)$, is still $2M \times 2M$ and its four block partitions are now defined in terms of (Wirtinger) complex gradient operations relative to the variables z and z^* as follows (compare with (B.14)):

$$H_c(z) \triangleq \left[\begin{array}{c|c} H_{z^*z} & (H_{z^\mathsf{T}z})^* \\ \hline H_{z^\mathsf{T}z} & (H_{z^*z})^\mathsf{T} \end{array}\right] \qquad (2M \times 2M) \tag{B.19}$$

where the $M \times M$ block matrices $\{H_{z^*z}, H_{z^\mathsf{T}z}\}$ correspond to the operations:

$$\left\{\begin{array}{lcl} H_{z^*z} & \triangleq & \nabla_{z^*}[\nabla_z\, g(z)] \\ H_{z^\mathsf{T}z} & \triangleq & \nabla_{z^\mathsf{T}}[\nabla_z\, g(z)] \end{array}\right. \tag{B.20}$$

It is clear from definition (B.19) that the complex Hessian matrix is now Hermitian so that

$$H_c(z) \;=\; [H_c(z)]^* \tag{B.21}$$

For example, for the same case (B.12) when z is a scalar, definition (B.19) leads to:

$$H_c(z) = \begin{bmatrix} \dfrac{\partial^2 g(z)}{\partial z^* \partial z} & \dfrac{\partial^2 g(z)}{\partial z^{*2}} \\ \dfrac{\partial^2 g(z)}{\partial z^2} & \dfrac{\partial^2 g(z)}{\partial z \partial z^*} \end{bmatrix} \tag{B.22}$$

Likewise, for the two-dimensional case (B.13), the complex Hessian matrix is given by:

$$H_c(z) = \left[\begin{array}{cc|cc} \dfrac{\partial^2 g(z)}{\partial z_1^* \partial z_1} & \dfrac{\partial^2 g(z)}{\partial z_1^* \partial z_2} & \dfrac{\partial^2 g(z)}{\partial z_1^{*2}} & \dfrac{\partial^2 g(z)}{\partial z_1^* \partial z_2^*} \\ \dfrac{\partial^2 g(z)}{\partial z_2^* \partial z_1} & \dfrac{\partial^2 g(z)}{\partial z_2^* \partial z_2} & \dfrac{\partial^2 g(z)}{\partial z_2^* \partial z_1^*} & \dfrac{\partial^2 g(z)}{\partial z_2^{*2}} \\ \hline \dfrac{\partial^2 g(z)}{\partial z_1^2} & \dfrac{\partial^2 g(z)}{\partial z_1 \partial z_2} & \dfrac{\partial^2 g(z)}{\partial z_1 \partial z_1^*} & \dfrac{\partial^2 g(z)}{\partial z_1 \partial z_2^*} \\ \dfrac{\partial^2 g(z)}{\partial z_2 \partial z_1} & \dfrac{\partial^2 g(z)}{\partial z_2^2} & \dfrac{\partial^2 g(z)}{\partial z_2 \partial z_1^*} & \dfrac{\partial^2 g(z)}{\partial z_2 \partial z_2^*} \end{array}\right] \tag{B.23}$$

Observe further that if we introduce the $2M \times 1$ extended vector:

$$u \triangleq \operatorname{col}\left\{ z, (z^*)^{\mathsf{T}} \right\} \tag{B.24}$$

then we can express $H_c(z)$ in the following equivalent form in terms of the variable u (compare with (B.17)):

$$H_c(u) \triangleq \nabla_{u^*} \left[\nabla_u \, g(u) \right] = \nabla_u \left[\nabla_{u^*} \, g(u) \right] = H_c(z) \quad (2M \times 2M) \tag{B.25}$$

B.2.3 Relation Between Both Representations

The two Hessian forms, $H(v)$ and $H_c(u)$, defined by (B.17) and (B.25) are closely related to each other. Indeed, using (A.10), it can be verified that [218, 251]:

$$\begin{cases} H_c(u) &= \frac{1}{4} D H(v) D^* \\ H(v) &= D^* H_c(u) D \end{cases} \tag{B.26}$$

where D is the following $2M \times 2M$ block matrix:

$$D \triangleq \begin{bmatrix} I_M & jI_M \\ I_M & -jI_M \end{bmatrix} \tag{B.27}$$

where I_M denotes the identity matrix of size M. It is straightforward to verify that

$$DD^* = 2\,I_{2M} \tag{B.28}$$

so that D is almost unitary (apart from scaling by $1/\sqrt{2}$).

It follows from (B.26) and (B.28) that the matrices $H_c(u)$ and $\frac{1}{2}H(v)$ are similar to each other and, hence, the eigenvalues of $H_c(u)$ coincide with the eigenvalues of $\frac{1}{2}H(v)$ [104, 113]. We conclude that the *complex* Hessian matrix, $H_c(u)$, can also be used to characterize the nature of stationary points of $g(z)$, just like it was the case with the *real* Hessian matrix, $H(v)$. Specifically, if $z^o = x^o + jy^o$ is a stationary point of $g(z)$ (i.e., a point where $\nabla_z\, g(z) = 0$), then the following facts hold:

(a) z^o is a local minimum of $g(z)$ if $H_c(u^o) > 0$, i.e., all eigenvalues of $H_c(u^o)$ are positive.

(b) z^o is a local maximum of $g(z)$ if $H_c(u^o) < 0$, i.e., all eigenvalues of $H_c(u^o)$ are negative.

where $u^o = \mathrm{col}\left\{z^o, (z^{o*})^{\mathsf{T}}\right\}$.

For ease of reference, Table B.1 summarizes the various definitions of Hessian matrices for *real-valued* functions $g(z) \in \mathbb{R}$ for both cases when z is real or complex-valued. In the latter case, there are two equivalent representations for the Hessian matrix: one representation is in terms of the real components $\{x, y\}$ and the second representation is in terms of the complex components $\{z, z^*\}$. The Hessian matrix has dimensions $M \times M$ when z is real and $2M \times 2M$ when z is complex. It is customary to use the compact notation $\nabla_z^2\, g(z)$ to refer to the Hessian matrix whether z is real or complex and by that notation we mean the following:

$$\nabla_z^2\, g(z) \triangleq \begin{cases} \nabla_{z^{\mathsf{T}}}[\nabla_z\, g(z)], & \text{when } z \text{ is real } \ (M \times M) \\ \nabla_{u^*}[\nabla_u\, g(u)], & \text{when } z \text{ is complex } \ (2M \times 2M) \end{cases} \tag{B.29}$$

Table B.1: Definition of Hessian matrices for real-valued functions $g(z) \in \mathbb{R}$ for both cases when z is real-valued or complex-valued.

	Hessian matrix	**variables**	**dimensions**
z **real**	$H(z) = \nabla_{z^{\mathsf{T}}}[\nabla_z\, g(z)]$		$M \times M$
z **complex** $z = x + jy$	$H(v) = \nabla_{v^{\mathsf{T}}}[\nabla_v\, g(v)]$	$v = \begin{bmatrix} x \\ y \end{bmatrix}$	$2M \times 2M$
	$H_c(u) = \nabla_{u^*}[\nabla_u\, g(u)]$	$u = \begin{bmatrix} z \\ (z^*)^{\mathsf{T}} \end{bmatrix}$	

Example B.2 (Hessian matrix calculations)**.** Let us illustrate the above definitions by considering a couple of examples.

(1) Let $g(z) = |z|^2 = x^2 + y^2$, where z is a scalar. Then,

$$H(v) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \equiv H, \qquad H_c(u) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \equiv H_c \tag{B.30}$$

In this case, the Hessian matrices turn out to be constant and independent of v and u.

(2) Consider now

$$g(z) = |z_1|^2 + 2\,\mathrm{Re}(z_1^* z_2) \;=\; x_1^2 + y_1^2 + 2x_1x_2 + 2y_1y_2 \tag{B.31}$$

where $z = \mathrm{col}\{z_1, z_2\}$ is 2×1. Then, the Hessian matrices are again independent of v and u:

$$H(v) = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix} \equiv H, \qquad H_c(u) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \equiv H_c \tag{B.32}$$

(3) Consider the real-valued quadratic function:

$$g(z) = \kappa + a^* z + z^* a + z^* C z \tag{B.33}$$

where κ is a real scalar, a is a column vector, and C is a Hermitian matrix. Then,

$$H_{z^*z} = \nabla_{z^*}\left[\nabla_z\, g(z)\right] = C \tag{B.34}$$

$$H_{z^\mathsf{T}z} = \nabla_{z^\mathsf{T}}\left[\nabla_z\, g(z)\right] = 0 \tag{B.35}$$

so that

$$H_c(u) = \begin{bmatrix} C & 0 \\ 0 & C^\mathsf{T} \end{bmatrix} \equiv H_c \tag{B.36}$$

$$H(v) = \begin{bmatrix} C + C^\mathsf{T} & j(C - C^\mathsf{T}) \\ j(C^\mathsf{T} - C) & C + C^\mathsf{T} \end{bmatrix} \equiv H \tag{B.37}$$

It follows from the expression for $H_c(u)$ that it is sufficient to examine the inertia of C to determine the nature of the stationary point(s) of $g(z)$.

■

Example B.3 (Block diagonal Hessian matrix)**.** Observe from definition (B.19) that the complex Hessian matrix becomes *block diagonal* whenever $H_{z^\mathsf{T}z} = 0$ in which case

$$H_c(z) = \begin{bmatrix} H_{z^*z} & 0 \\ 0 & (H_{z^*z})^\mathsf{T} \end{bmatrix} \qquad (2M \times 2M) \tag{B.38}$$

For example, as shown in the calculation leading to (B.36), block diagonal Hessian matrices, $H_c(z)$ or $H_c(u)$, arise when $g(z)$ is quadratic in z. Such quadratic functions are common in design problems involving mean-square-error criteria in adaptation and learning — see, e.g., expression (2.63) in the body of the text.

■

C

Convex Functions

Let $g(z) \in \mathbb{R}$ denote a *real-valued* function of a possibly vector argument, $z \in \mathbb{C}^M$. It is sufficient for our purposes to assume that $g(z)$ is differentiable whenever necessary (although we shall also comment on the situation in which $g(z)$ may not be differentiable at some points). By differentiability here we mean that the (Wirtinger) complex gradient vector, $\nabla_z g(z)$, and the Hessian matrix, $\nabla_z^2 g(z)$, both exist in the manner defined in Appendices A and B. In particular, if we express z in terms of its real and imaginary arguments, $z = x + jy$, then we are assuming that the following partial derivatives exist whenever necessary:

$$\frac{\partial g(x,y)}{\partial x_m}, \quad \frac{\partial g(x,y)}{\partial y_n}, \quad \frac{\partial^2 g(x,y)}{\partial x_m^2}, \quad \frac{\partial^2 g(x,y)}{\partial y_n}, \quad \frac{\partial^2 g(x,y)}{\partial x_m \partial y_n} \tag{C.1}$$

for $n, m = 1, 2, \ldots, M$, and where $\{x_m, y_n\}$ denote the individual entries of the vectors $x, y \in \mathbb{R}^M$.

In the sequel, we define convexity for both cases when $z \in \mathbb{R}^M$ is real-valued and when $z \in \mathbb{C}^M$ is complex-valued. We start with the former case of real z, which is the situation most commonly studied

in the literature [29, 45, 177, 190]. Subsequently, we explain how the definitions and results extend to functions of complex arguments, z; these extensions are necessary to deal with situations that arise in the context of adaptation and learning in signal processing and communications problems.

C.1 Convexity in the Real Domain

We assume initially that the argument $z \in \mathbb{R}^M$ is real-valued where, as already stated earlier, the function $g(z) \in \mathbb{R}$ is also real-valued. We discuss three forms of convexity: the standard definition of convexity followed by strict convexity and then strong convexity.

C.1.1 Convex Sets

We first introduce the notion of convex sets. A set $\mathcal{S} \subset \mathbb{R}^M$ is said to be convex if for any pair of points $z_1, z_2 \in \mathcal{S}$, all points that lie on the line segment connecting z_1 and z_2 also belong to $\mathcal{S}$. Specifically,

$$\forall z_1, z_2 \in \mathcal{S} \;\text{ and }\; 0 \leq \alpha \leq 1 \;\;\Longrightarrow\;\; \alpha z_1 + (1-\alpha) z_2 \in \mathcal{S}. \tag{C.2}$$

Figure C.1 illustrates this definition by showing two convex sets and one non-convex set. In the latter case, a segment is drawn between two points inside the set and it is seen that some of the points on the segment lie outside the set.

C.1.2 Convexity

The function $g(z)$ is said to be convex if its domain, written as $dom(g)$, is a convex set and if for any points $z_1, z_2 \in dom(g)$ and for any $0 \leq \alpha \leq 1$, it holds that

$$g(\alpha z_1 + (1-\alpha) z_2) \;\leq\; \alpha g(z_1) \;+\; (1-\alpha) g(z_2) \tag{C.3}$$

In other words, all points belonging to the line segment connecting $g(z_1)$ to $g(z_2)$ lie on or above the graph of $g(z)$ — see the plot on the left side of Figure C.2. An equivalent characterization of convexity is

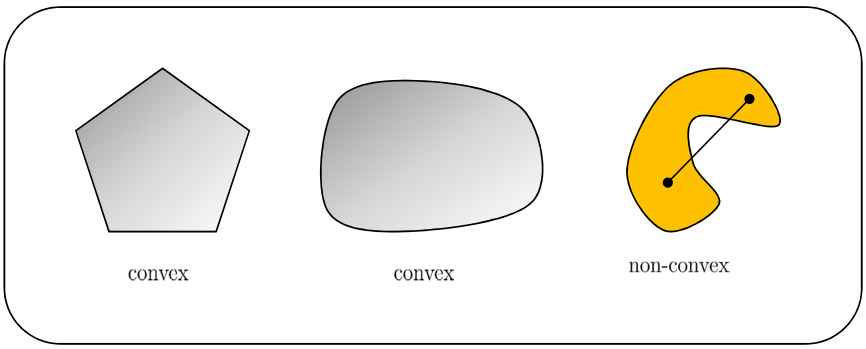

Figure C.1: The two sets on the left are examples of convex sets, while the set on the right is a non-convex set.

that for any z_o and z:

$$g(z) \;\geq\; g(z_o) \;+\; [\nabla_z\, g(z_o)]\,(z - z_o) \tag{C.4}$$

in terms of the inner product between the gradient vector at z_o and the vector difference $(z - z_o)$. This condition means that the tangent plane at z_o lies beneath the graph of the function — see the plot on the right side of Figure C.2.

A useful property of every convex function is that, when a minimum exists, it can only be a global minimum; there can be multiple global minima but no local minima. That is, any stationary point at which the gradient vector of $g(z)$ is annihilated can only correspond to a global minimum of the function; the function cannot have local maxima, minima, or saddle points. A second useful property of convex functions, and which follows from (C.4), is that for any z_1 and z_2:

$$g(z) \text{ convex} \implies [\nabla_z\, g(z_2) - \nabla_z\, g(z_1)]\,(z_2 - z_1) \geq 0 \tag{C.5}$$

in terms of the inner product between two differences: the difference in the gradient vectors and the difference in the vectors themselves. The above result means that these difference vectors are aligned (i.e., have a nonnegative inner product). Result (C.5) follows by using (C.4) to

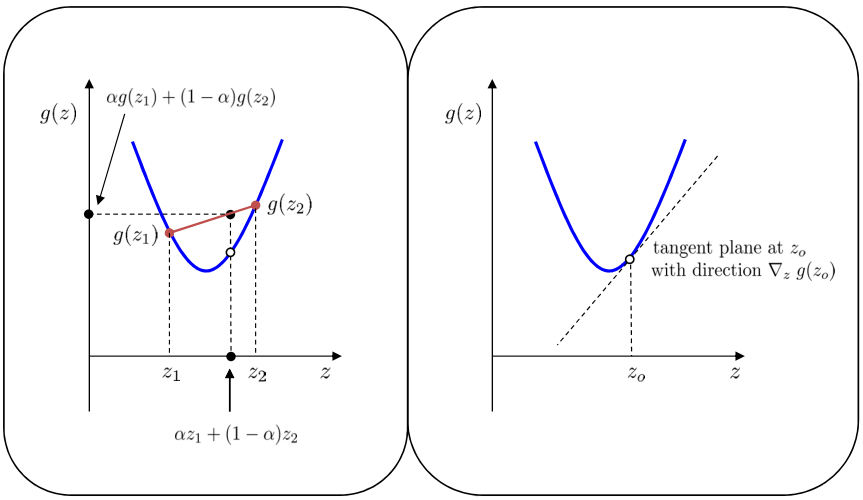

Figure C.2: Two equivalent characterizations of convexity for *differentiable* functions $g(z)$ as defined by (C.3) and (C.4).

write

$$g(z_2) \;\geq\; g(z_1) \;+\; [\nabla_z\, g(z_1)]\,(z_2 - z_1) \tag{C.6}$$

$$g(z_1) \;\geq\; g(z_2) \;+\; [\nabla_z\, g(z_2)]\,(z_1 - z_2) \tag{C.7}$$

so that upon substitution of the second inequality into the right-hand side of the first inequality we obtain

$$g(z_2) \;\geq\; g(z_2) \;+\; [\nabla_z\, g(z_2)]\,(z_1 - z_2) \;+\; [\nabla_z\, g(z_1)]\,(z_2 - z_1) \tag{C.8}$$

from which we obtain (C.5).

Example C.1 (Convexity and sub-gradients)**.** Property (C.4) is stated in terms of the gradient vector of $g(z)$ evaluated at location z_o. This gradient vector exists because we assumed the function $g(z)$ to be differentiable. There exist, however, cases where the function $g(z)$ need not be differentiable at all points. For example, for scalar arguments z, the function $g(z) = |z|$ is convex but is not differentiable at $z = 0$. For such non-differentiable convex functions, the characterization (C.4) can be replaced by the statement that the function $g(z)$ is convex if, and only if, for every z_o, a row vector $y \in \partial g(z_o)$ can be found such that

$$g(z) \;\geq\; g(z_o) \;+\; y(z - z_o) \tag{C.9}$$

in terms of the inner product between y and the vector difference $(z - z_o)$. The vector y is called a sub-gradient and the notation $\partial g(z_o)$ denotes the set of all possible sub-gradients, also called the sub-differential of $g(z)$ at $z = z_o$; this situation is illustrated in Figure C.3. When $g(z)$ is differentiable at z_o, then there is a unique sub-gradient vector and it coincides with $\nabla_z g(z_o)$. In that case, statement (C.9) reduces to (C.4). We continue our presentation by focusing on differentiable functions $g(z)$.

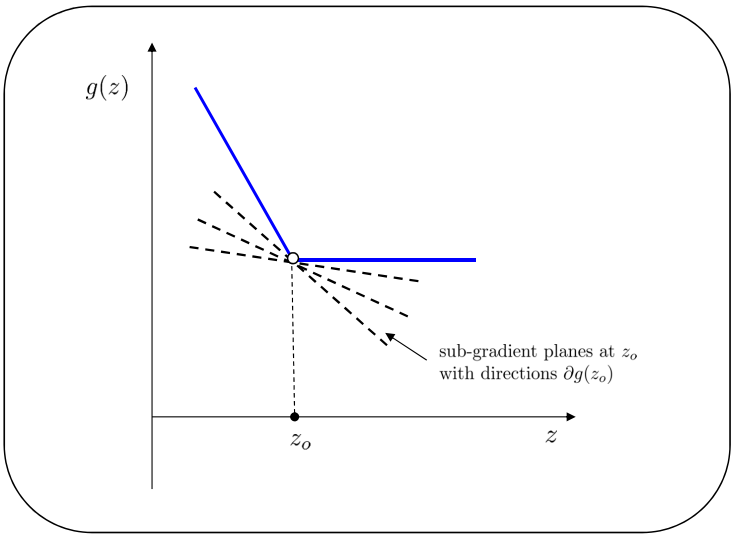

Figure C.3: A non-differentiable convex function with a multitude of sub-gradient directions at the point of non-differentiability.

■

Example C.2 (Some useful operations that preserve convexity)**.** It is straightforward to verify from the definition (C.3) that the following operations preserve convexity:

(1) if $g(z)$ is convex then $h(z) = g(Az + b)$ is also convex for any constant matrix A and vector b. That is, affine transformations of z do not destroy convexity.

(2) If $g_1(z)$ and $g_2(z)$ are convex functions, then $h(z) = \max\{g_1(z), g_2(z)\}$ is convex. That is, pointwise maximization does not destroy convexity.

(3) If $g_1(z)$ and $g_2(z)$ are convex functions, then $h(z) = a_1 g_1(z) + a_2 g_2(z)$ is also convex for any nonnegative coefficients a_1 and a_2.

■

C.1.3 Strict Convexity

The function $g(z)$ is said to be *strictly* convex if the inequalities in (C.3) or (C.4) are replaced by *strict* inequalities. More specifically, for any $z_1 \neq z_2$ and $0 < \alpha < 1$, a strictly convex function should satisfy:

$$g(\alpha z_1 + (1-\alpha)z_2) \;<\; \alpha g(z_1) \;+\; (1-\alpha)g(z_2) \tag{C.10}$$

A useful property of every strictly convex function is that, when a minimum exists, then it is both *unique* and also the global minimum of the function. A second useful property replaces (C.5) by the following statement with a strict inequality for any $z_1 \neq z_2$:

$$g(z) \text{ strictly convex} \implies [\nabla_z\, g(z_2) - \nabla_z\, g(z_1)]\,(z_2 - z_1) > 0 \tag{C.11}$$

C.1.4 Strong Convexity

The function $g(z)$ is said to be *strongly* convex (or, more specifically, $\nu-$strongly convex) if it satisfies the following stronger condition for any $0 \leq \alpha \leq 1$:

$$g(\alpha z_1 + (1-\alpha)z_2) \;\leq\; \alpha g(z_1) \;+\; (1-\alpha)g(z_2) \;-\; \frac{\nu}{2}\alpha(1-\alpha)\|z_1 - z_2\|^2 \tag{C.12}$$

for some scalar $\nu > 0$, and where the notation $\|\cdot\|$ denotes the Euclidean norm of its vector argument; although strong convexity can also be defined relative to other vector norms, the Euclidean norm is sufficient for our purposes. Comparing (C.12) with (C.10) we conclude that strong convexity implies strict convexity. Therefore, every strongly convex function has a unique global minimum as well. Nevertheless, strong convexity is a stronger condition than strict convexity so that functions exist that are strictly convex but not necessarily strongly convex. For example, for scalar arguments z, the function $g(z) = z^4$ is strictly convex but not strongly convex. On the other hand, the function $g(z) = z^2$ is strongly convex — see Figure C.4. In summary, it holds that:

$$\text{strong convexity} \implies \text{strict convexity} \implies \text{convexity} \tag{C.13}$$

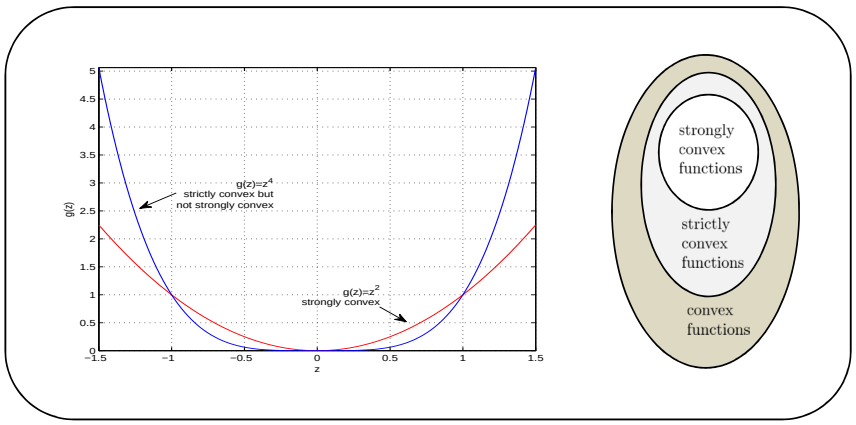

Figure C.4: The function $g(z) = z^4$ is strictly convex but not strongly convex, while the function $g(z) = z^2$ is strongly convex. Observe how $g(z) = z^4$ is more flat around its global minimizer and moves away from it more slowly than in the quadratic case.

A useful property of strongly convex functions is that they grow faster than a linear function in z since an equivalent characterization of strong convexity is that for any z_o and z:

$$g(z) \;\geq\; g(z_o) \;+\; [\nabla_z\, g(z_o)]\,(z - z_o) \;+\; \frac{\nu}{2}\|z - z_o\|^2 \tag{C.14}$$

This means that the graph of $g(z)$ is strictly above the tangent plane at location z_o and moreover, for any z, the distance between the graph and the corresponding point on the tangent plane is at least as large as the quadratic term $\frac{\nu}{2}\|z - z_o\|^2$. In particular, if we specialize (C.14) to the case in which z_o is selected to correspond to the global minimizer of $g(z)$, i.e., as

$$z_o = z^o, \quad \text{where} \quad \nabla_z\, g(z^o) = 0 \tag{C.15}$$

then we conclude that every strongly convex function satisfies the following useful property for every z:

$$g(z) \;-\; g(z^o) \geq \frac{\nu}{2}\|z - z^o\|^2, \quad (z^o \text{ is global minimizer}) \tag{C.16}$$

This property is illustrated in Figure C.5. Another useful property that

follows from (C.14) is that for any z_1, z_2:

$$g(z) \text{ strongly convex} \Longrightarrow [\nabla_z\, g(z_2) - \nabla_z\, g(z_1)]\,(z_2 - z_1) \geq \nu\|z_2 - z_1\|^2 \tag{C.17}$$

This fact, along with the earlier conclusions (C.5) and (C.11) are important properties of convex functions. We summarize them in Table C.1 for ease of reference.

Table C.1: Useful properties implied by the convexity, strict convexity, or strong convexity of a real-valued function $g(z) \in \mathbb{R}$ of a *real* argument $z \in \mathbb{R}^M$.

$g(z)$ convex $\Longrightarrow$	$[\nabla_z\, g(z_2) - \nabla_z\, g(z_1)]\,(z_2 - z_1) \geq 0$
$g(z)$ strictly convex $\Longrightarrow$	$[\nabla_z\, g(z_2) - \nabla_z\, g(z_1)]\,(z_2 - z_1) > 0$
$g(z)$ $\nu-$strongly convex $\Longrightarrow$	$[\nabla_z\, g(z_2) - \nabla_z\, g(z_1)]\,(z_2 - z_1) \geq \nu\|z_2 - z_1\|^2$

C.1.5 Hessian Matrix Conditions

We indicated earlier that it is sufficient for our treatment to assume that the real-valued function $g(z)$ is differentiable whenever necessary. In particular, when it is twice continuously differentiable, then the properties of convexity, strict convexity, and strong convexity can be inferred from the Hessian matrix of $g(z)$ as follows (see, e.g., [177, 190]):

$$\begin{cases} \text{(a)} \;\; \nabla_z^2\, g(z) \geq 0 \text{ for all } z & \Longleftrightarrow \;\; g(z) \text{ is convex.} \\ \text{(b)} \;\; \nabla_z^2\, g(z) > 0 \text{ for all } z & \Longrightarrow \;\; g(z) \text{ is strictly convex.} \\ \text{(c)} \;\; \nabla_z^2\, g(z) \geq \nu I_M > 0 \text{ for all } z & \Longleftrightarrow \;\; g(z) \text{ is } \nu-\text{strongly convex.} \end{cases} \tag{C.18}$$

Since $g(z)$ is real-valued and z is also real-valued in this section, then the Hessian matrix in this case is $M \times M$ and given by the expression shown in the first row of Table B.1 and by equation (B.29), namely,

$$\nabla_z^2\, g(z) \;\triangleq\; \nabla_{z^{\mathsf{T}}} [\nabla_z\, g(z)] \tag{C.19}$$

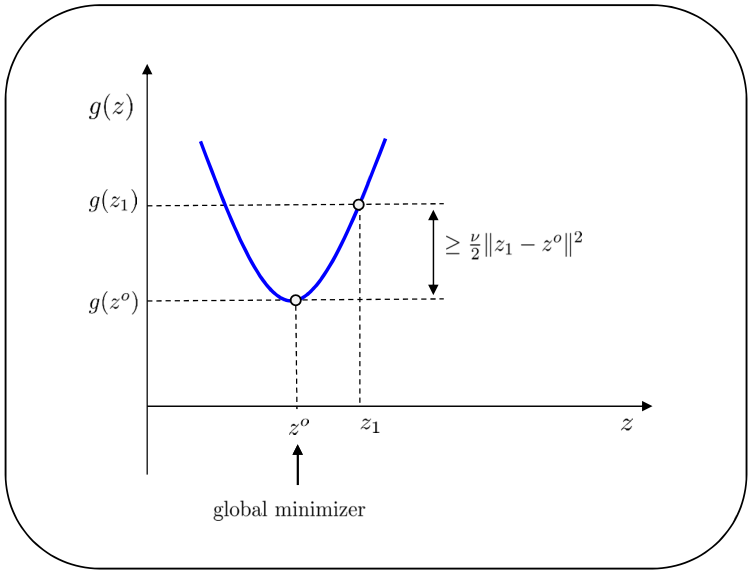

Figure C.5: For $\nu-$strongly convex functions, the increment $g(z_1) - g(z^o)$ grows at least as fast as the quadratic term $\frac{\nu}{2}\|z_1 - z^o\|^2$, as indicated by (C.16) and where z^o is the global minimizer of $g(z)$.

Observe from (C.18) that the positive definiteness of the Hessian matrix is only a sufficient condition for strict convexity (for example, the function $g(z) = z^4$ is strictly convex even though its second-order derivative is not strictly positive for all z). One of the main advantages of working with strongly convex functions is that their Hessian matrices are sufficiently bounded away from zero.

Example C.3 (Strongly-convex functions)**.** The following is a list of useful strongly-convex functions that appear in applications involving adaptation, learning, and estimation:

(1) Consider the quadratic function

$$g(z) \;=\; \kappa + a^{\mathsf{T}}z + z^{\mathsf{T}}a + z^{\mathsf{T}}Cz \tag{C.20}$$

with a symmetric positive-definite matrix C. The Hessian matrix is $\nabla_z^2\, g(z) =$

$2C$, which is sufficiently bounded away from zero for all z since

$$\nabla_z^2 g(z) \;\geq\; 2\lambda_{\min}(C)\, I_M > 0 \tag{C.21}$$

in terms of the smallest eigenvalue of C. Therefore, such quadratic functions are strongly convex.

(2) The regularized logistic (or log-)loss function

$$g(z) \;=\; \ln\left(1 + e^{-\gamma h^{\mathsf{T}} z}\right) \;+\; \frac{\rho}{2}\|z\|^2 \tag{C.22}$$

with a scalar γ, column vector h, and $\rho > 0$ is also strongly convex. This is because the Hessian matrix is given by

$$\nabla_z^2 g(z) \;=\; \rho\, I_M \;+\; hh^{\mathsf{T}}\left(\frac{e^{-\gamma h^{\mathsf{T}} z}}{(1 + e^{-\gamma h^{\mathsf{T}} z})^2}\right) \;\geq\; \rho\, I_M \;>\; 0 \tag{C.23}$$

(3) The regularized hinge loss function

$$g(z) \;=\; \max\left\{0, 1 - \gamma h^{\mathsf{T}} z\right\} \;+\; \frac{\rho}{2}\|z\|^2 \tag{C.24}$$

with a scalar γ, column vector h, and $\rho > 0$ is also strongly convex, although non-differentiable. This result can be verified by noting that the function $\max\left\{0, 1 - \gamma h^{\mathsf{T}} z\right\}$ is convex in z while the regularization term $\frac{\rho}{2}\|z\|^2$ is $\rho-$strongly convex in z.

■

C.2 Convexity in the Complex Domain

We now extend the previous definitions and results to the case in which $z \in \mathbb{C}^M$ is complex-valued, while $g(z) \in \mathbb{R}$ continues to be real-valued. One way to extend the concepts of convexity, strict convexity, and strong convexity to the case of complex arguments is to view $g(z)$ as a function of the extended real variable $v = \mathrm{col}\{x, y\} \in \mathbb{R}^{2M}$, i.e., to work with $g(v)$ instead of $g(z)$, where v is defined in terms of the real and imaginary parts of z, namely, $z = x + jy$. Observe in particular that the complex variables z and z^* can be recovered from knowledge of v as follows:

$$\underbrace{\begin{bmatrix} I_M & jI_M \\ I_M & -jI_M \end{bmatrix}}_{=D}\underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{=v} \;=\; \begin{bmatrix} z \\ (z^*)^{\mathsf{T}} \end{bmatrix} \tag{C.25}$$

where the matrix D was introduced earlier in (B.27).

C.2.1 Convexity

The function $g(z)$ is said to be convex in z if the corresponding function $g(v)$ is convex in v, i.e., if $dom(g(v))$ is a convex set and for any $v_1, v_2 \in dom(g(v))$ and any $0 \leq \alpha \leq 1$, it holds that:

$$g(\alpha v_1 + (1-\alpha)v_2) \;\leq\; \alpha g(v_1) \;+\; (1-\alpha)g(v_2) \tag{C.26}$$

Since $g(z)$ is real-valued, the above condition can be restated in terms of the original complex variables $z_1, z_2 \in \mathbb{C}^M$ as follows:

$$g(\alpha z_1 + (1-\alpha)z_2) \;\leq\; \alpha g(z_1) \;+\; (1-\alpha)g(z_2) \tag{C.27}$$

An equivalent characterization of the convexity condition (C.26) is that for any v_o,

$$g(v) \;\geq\; g(v_o) \;+\; [\nabla_v\, g(v_o)]\,(v - v_o) \tag{C.28}$$

This condition can again be restated in terms of the original complex variables $\{z, z_o\}$. To do so, we first need to find the relation between the gradient vector $\nabla_v\, g(v)$ evaluated in the $v-$domain and the gradient vector $\nabla_z\, g(z)$ evaluated in the $z-$domain. Thus, recall that v is a column vector obtained by stacking x and y on top of each other. Therefore, by referring to definition (A.26), we have that

$$\nabla_v\, g(v) \;=\; \begin{bmatrix} \nabla_x\, g(x,y) & \nabla_y\, g(x,y) \end{bmatrix} \tag{C.29}$$

Multiplying from the right by the matrix D^* from (B.27) we obtain

$$\nabla_v\, g(v) \cdot \frac{1}{2} D^* = \frac{1}{2} \begin{bmatrix} \nabla_x\, g(x,y) & \nabla_y\, g(x,y) \end{bmatrix} \begin{bmatrix} I_M & I_M \\ -jI_M & jI_M \end{bmatrix} \tag{C.30}$$

Now consider the following complex gradient vectors, which correspond to the extension of the earlier definition (A.9) to the vector case for real-valued functions $g(z)$:

$$\begin{cases} \nabla_z\, g(z) & \triangleq & \frac{1}{2}\left[\nabla_x\, g(x,y) \;-\; j\nabla_y\, g(x,y)\right] \\ \nabla_{z^*}\, g(z) & \triangleq & \frac{1}{2}\left[\nabla_{x^\mathsf{T}}\, g(x,y) \;+\; j\nabla_{y^\mathsf{T}}\, g(x,y)\right] \end{cases} \tag{C.31}$$

Substituting into the right-hand side of (C.30) we conclude that

$$\frac{1}{2}\left[\nabla_v\, g(v)\right] D^* \;=\; \begin{bmatrix} \nabla_z\, g(z) & (\nabla_{z^*}\, g(z))^\mathsf{T} \end{bmatrix} \tag{C.32}$$

which is the desired relation between the gradient vectors $\nabla_v g(v)$ and $\nabla_z g(z)$. Using (C.25) and (C.32), and noting that $g(z) = g(v)$, we can now rewrite (C.28) in terms of the original complex variables $\{z, z_o\}$ as follows:

$$g(z) \;\geq\; g(z_o) \;+\; 2\mathrm{Re}\,\{\,[\nabla_z g(z_o)]\,(z - z_o)\,\} \tag{C.33}$$

in terms of the real part of the inner product that appears on the right-hand side. A useful property that follows from (C.33) is that for any z_1 and z_2 :

$$g(z) \text{ convex} \Longrightarrow \mathrm{Re}\,\{\,[\nabla_z\, g(z_2) - \nabla_z\, g(z_1)]\,(z_2 - z_1)\,\} \geq 0 \tag{C.34}$$

C.2.2 Strict Convexity

The function $g(z)$ is said to be *strictly* convex if the inequalities in (C.27) or (C.33) are replaced by strict inequalities. For example, for any $z_1 \neq z_2$ and $0 < \alpha < 1$, a strictly convex function $g(z)$ should satisfy:

$$g(\alpha z_1 + (1-\alpha)z_2) \;<\; \alpha g(z_1) \;+\; (1-\alpha)g(z_2) \tag{C.35}$$

Again, a useful property of every strictly convex function is that, when a minimum exists, then it is both unique and the global minimum of the function. Another useful property is that for any $z_1 \neq z_2$:

$$g(z) \text{ strictly convex} \Longrightarrow \mathrm{Re}\,\{[\nabla_z\, g(z_2) - \nabla_z\, g(z_1)]\,(z_2 - z_1)\} > 0 \tag{C.36}$$

C.2.3 Strong Convexity

The function $g(z)$ is said to be *strongly* convex (or, more specifically, $\nu-$strongly convex) in z if $g(v)$ is $\nu-$strongly convex in v, i.e., if $g(v)$ satisfies the following condition for any $0 \leq \alpha \leq 1$:

$$g(\alpha v_1 + (1-\alpha)v_2) \;\leq\; \alpha g(v_1) \;+\; (1-\alpha)g(v_2) \;-\; \frac{\nu}{2}\alpha(1-\alpha)\|v_1 - v_2\|^2 \tag{C.37}$$

for some $\nu > 0$. Using the fact that

$$\|v_1 - v_2\|^2 \;=\; \|z_1 - z_2\|^2 \tag{C.38}$$

the above condition can be restated in terms of the original complex variables as follows:

$$g(\alpha z_1+(1-\alpha)z_2) \leq \alpha g(z_1)+(1-\alpha)g(z_2)-\frac{\nu}{2}\alpha(1-\alpha)\|z_1-z_2\|^2 \quad \text{(C.39)}$$

An equivalent characterization of strong convexity is that for any z_o,

$$g(z) \;\geq\; g(z_o) \;+\; 2\mathrm{Re}\left\{[\nabla_z\, g(z_o)]\,(z-z_o)\right\} \;+\; \frac{\nu}{2}\|z-z_o\|^2 \quad \text{(C.40)}$$

In particular, if we select z_o to correspond to the global minimizer of $g(z)$, i.e.,

$$z_o = z^o \quad \text{where} \quad \nabla_z\, g(z^o) = 0 \quad \text{(C.41)}$$

then strongly convex functions satisfy the following useful property:

$$g(z) \;-\; g(z^o) \geq \frac{\nu}{2}\|z-z^o\|^2, \quad (z^o \text{ is global minimizer}) \quad \text{(C.42)}$$

Another useful property that follows from (C.40) is that for any z_1, z_2:

$$g(z) \text{ strongly convex} \Longrightarrow$$

$$\mathrm{Re}\left\{[\nabla_z\, g(z_2)-\nabla_z\, g(z_1)]\,(z_2-z_1)\right\} \geq \frac{\nu}{2}\|z_2-z_1\|^2 \quad \text{(C.43)}$$

This fact, along with the earlier conclusions (C.34) and (C.36) are important properties of convex functions. We summarize them in Table C.2 for ease of reference.

Table C.2: Useful properties implied by the convexity, strict convexity, or strong convexity of a real-valued function $g(z) \in \mathbb{R}$ of a *complex* argument $z \in \mathbb{C}^M$.

$g(z)$ convex $\Longrightarrow$	$\mathrm{Re}\left\{\,[\nabla_z\, g(z_2)-\nabla_z\, g(z_1)]\,(z_2-z_1)\,\right\} \geq 0$
$g(z)$ strictly convex $\Longrightarrow$	$\mathrm{Re}\left\{[\nabla_z\, g(z_2)-\nabla_z\, g(z_1)]\,(z_2-z_1)\right\} > 0$
$g(z)$ $\nu-$strongly convex $\Longrightarrow$	$\mathrm{Re}\left\{[\nabla_z\, g(z_2)-\nabla_z\, g(z_1)]\,(z_2-z_1)\right\} \geq \frac{\nu}{2}\|z_2-z_1\|^2$

C.2.4 Hessian Matrix Conditions

Since $g(z)$ is real-valued and z is now complex-valued, then the Hessian matrix of $g(z)$ is $2M \times 2M$ and given by the expression shown in the last row of Table B.1 — see (B.29). As before, the properties of convexity, strict convexity, and strong convexity can be inferred from the Hessian matrix of $g(z)$ as follows:

$$\left\{\begin{array}{lll} \text{(a)} \;\; \nabla_z^2\, g(z) \geq 0 \text{ for all } z & \Longleftrightarrow & g(z) \text{ is convex.} \\ \text{(b)} \;\; \nabla_z^2\, g(z) > 0 \text{ for all } z & \Longrightarrow & g(z) \text{ is strictly convex.} \\ \text{(c)} \;\; \nabla_z^2\, g(z) \geq \frac{\nu}{2} I_{2M} > 0 \text{ for all } z & \Longleftrightarrow & g(z) \text{ is strongly convex.} \end{array}\right. \tag{C.44}$$

Observe again that the positive definiteness of the Hessian matrix is only a sufficient condition for strict convexity. Moreover, the condition in part (c), with a factor of $\frac{1}{2}$ multiplying ν, follows from the following sequence of arguments:

$$\begin{aligned} g(z) \text{ is } \nu\text{−strongly convex} &\Longleftrightarrow g(v) \text{ is } \nu\text{−strongly convex} \\ &\overset{\text{(C.18)}}{\Longleftrightarrow} H(v) \geq \nu I_{2M} > 0, \quad \text{for all } v \\ &\Longleftrightarrow \frac{1}{4} DH(v)D^* \geq \frac{\nu}{4} DD^* \overset{\text{(B.28)}}{=} \frac{\nu}{2} I > 0 \\ &\overset{\text{(B.26)}}{\Longleftrightarrow} H_c(u) \geq \frac{\nu}{2} I_{2M} > 0 \end{aligned} \tag{C.45}$$

where we used the notation $H(v)$ and $H_c(u)$ to refer to the real and complex forms of the Hessian matrix of $g(z)$ — recall (B.17) and (B.25).

Example C.4 (Quadratic cost functions). Consider the quadratic function

$$g(z) = \kappa + a^* z + z^* a + z^* C z \tag{C.46}$$

with a Hermitian positive-definite matrix $C > 0$. The complex Hessian matrix is given by

$$H_c(u) \;=\; \left[\begin{array}{cc} C & 0 \\ 0 & C^{\mathsf{T}} \end{array}\right] \tag{C.47}$$

which is sufficiently bounded away from zero from below since

$$H_c(u) \;\geq\; \lambda_{\min}(C)\, I_{2M} > 0 \tag{C.48}$$

Therefore, such quadratic functions are strongly convex.

■

D

Mean-Value Theorems

Let $g(z) \in \mathbb{R}$ denote a *real-valued* function of a possibly vector argument z. We assume that $g(z)$ is differentiable whenever necessary. In this appendix, we review useful integral equalities that involve increments of the function $g(z)$ and increments of its gradient vector; the equalities correspond to extensions of the classical mean-value theorem from single-variable real calculus to the case of functions of several and possibly complex variables. We shall use the results of this appendix to establish useful bounds on the increments of strongly convex functions later in Appendix E. We again treat both cases of real and complex arguments.

D.1 Increment Formulae for Real Arguments

Consider first the case in which the argument $z \in \mathbb{R}^M$ is real-valued. We pick any $M-$dimensional vectors z_o and Δz and introduce the following real-valued and differentiable function of the scalar variable $t \in [0, 1]$:

$$f(t) \triangleq g(z_o + t\,\Delta z) \tag{D.1}$$

Then, it holds that

$$f(0) = g(z_o), \qquad f(1) = g(z_o + \Delta z) \tag{D.2}$$

Using the fundamental theorem of calculus (e.g., [36, 150]) we have:

$$f(1) - f(0) = \int_0^1 \frac{df(t)}{dt} dt \tag{D.3}$$

It further follows from definition (D.1) that

$$\frac{df(t)}{dt} \;=\; \frac{d}{dt}\left[g(z_o + t\,\Delta z)\right] \;=\; \left[\nabla_z\, g(z_o + t\,\Delta z)\right]\Delta z \tag{D.4}$$

in terms of the inner product computation on the far right, where $\nabla_z\, g(z)$ denotes the (row) gradient vector of $g(z)$ with respect to z. Substituting (D.4) into (D.3) we arrive at the first desired mean-value theorem result (see, e.g., [190]):

$$g(z_o + \Delta z) \;-\; g(z_o) \;=\; \left(\int_0^1 \nabla_z\, g(z_o + t\Delta z) dt\right)\Delta z \tag{D.5}$$

This result is a useful equality and it holds for any differentiable (*not necessarily convex*) real-valued function $g(z)$. The expression on the right-hand side is an inner product between the column vector Δz and the result of the integration, which is a row vector. Expression (D.5) tells us how the increment of the function $g(z)$ in moving from $z = z_o$ to $z = z_o + \Delta z$ is related to the integral of the gradient vector of $g(z)$ over the segment $z_o + t\,\Delta z$ as t varies over the interval $t \in [0, 1]$.

We can derive a similar relation for increments of the gradient vector itself. To do so, we introduce the column vector function $h(z) = \nabla_{z^{\mathsf{T}}}\, g(z)$ and apply (D.5) to its individual entries to conclude that

$$h(z_o + \Delta z) \;-\; h(z_o) \;=\; \left(\int_0^1 \nabla_z\, h(z_o + r\,\Delta z) dr\right)\Delta z \tag{D.6}$$

Replacing $h(z)$ by its definition, and transposing both sides of the above equality, we arrive at another useful mean-value theorem result:

$$\nabla_z\, g(z_o + \Delta z) \;-\; \nabla_z\, g(z_o) \;=\; \Delta z^{\mathsf{T}} \left(\int_0^1 \nabla_z^2\, g(z_o + r\,\Delta z) dr\right) \tag{D.7}$$

This expression tells us how increments in the gradient vector in moving from $z = z_o$ to $z = z_o + \Delta z$ are related to the integral of the Hessian

matrix of $g(z)$ over the segment $z_o + r\,\Delta z$ and r varies over the interval $r \in [0,1]$. In summary, we arrive at the following statement.

Lemma D.1 (Mean-value theorem: Real arguments)**.** Consider a real-valued and twice-differentiable function $g(z) \in \mathbb{R}$, where $z \in \mathbb{R}^M$ is real-valued. Then, for any $M-$dimensional vectors z_o and Δz, the following increment equalities hold:

$$g(z_o + \Delta z) \;-\; g(z_o) \;=\; \left(\int_0^1 \nabla_z\, g(z_o + t\,\Delta z)dt\right)\Delta z \tag{D.8}$$

$$\nabla_z\, g(z_o + \Delta z) \;-\; \nabla_z\, g(z_o) \;=\; (\Delta z)^{\mathsf{T}}\left(\int_0^1 \nabla_z^2\, g(z_o + r\,\Delta z)dr\right) \tag{D.9}$$

D.2 Increment Formulae for Complex Arguments

We now extend results (D.8) and (D.9) to the case when $z \in \mathbb{C}^M$ is complex valued. The extension can be achieved by replacing $z = x + jy$ by its real and imaginary parts $\{x, y\}$, applying results (D.8) and (D.9) to the resulting function $g(v)$ of the $2M \times 1$ extended real variable

$$v = \mathrm{col}\{x, y\} \tag{D.10}$$

and then transforming back to the complex domain. Indeed, as remarked earlier in (C.25), it is straightforward to verify that the vector v is related to the vector

$$u \;\triangleq\; \mathrm{col}\{z, (z^*)^{\mathsf{T}}\} \tag{D.11}$$

as follows:

$$\begin{cases} \underbrace{\begin{bmatrix} z \\ (z^*)^{\mathsf{T}} \end{bmatrix}}_{\triangleq\, u} \;=\; \underbrace{\begin{bmatrix} I_M & jI_M \\ I_M & -jI_M \end{bmatrix}}_{\triangleq\, D} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\triangleq\, v} \\ \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{v} \;=\; \underbrace{\frac{1}{2}\begin{bmatrix} I_M & I_M \\ -jI_M & jI_M \end{bmatrix}}_{=\frac{1}{2}D^*} \underbrace{\begin{bmatrix} z \\ (z^*)^{\mathsf{T}} \end{bmatrix}}_{u} \end{cases} \tag{D.12}$$

or, more compactly,

$$u = Dv \quad \text{and} \quad v = \frac{1}{2}D^{*}u \tag{D.13}$$

where we used the fact from (B.28) that $DD^{*} = 2I_{2M}$. We can now apply (D.8) to $g(v)$ to get

$$g(v_o + \Delta v) \;-\; g(v_o) \;=\; \left(\int_0^1 \nabla_v \, g(v_o + t\Delta v)dt\right)\Delta v \tag{D.14}$$

where $\nabla_v \, g(v)$ denotes the gradient vector of $g(v)$. We can rewrite (D.14) in terms of the original complex variables $\{z_o, \Delta z\}$. To do so, we call upon relation (C.32) and the equality $g(z) = g(v)$ to rewrite (D.14) as

$$\begin{aligned} & g(z_o + \Delta z) - g(z_o) = \\ & \overset{(D.13)}{=} \frac{1}{2}\left(\int_0^1 \nabla_v \, g(v_o + t\,\Delta v)dt\right) D^{*} \underbrace{D\Delta v}_{\triangleq \Delta u} \\ & \overset{(C.32)}{=} \left(\int_0^1 \left[\begin{array}{cc} \nabla_z \, g(z_o + t\,\Delta z) & (\nabla_{z^*} \, g(z_o + t\,\Delta z))^{\mathsf{T}} \end{array}\right] dt\right) \left[\begin{array}{c} \Delta z \\ (\Delta z^*)^{\mathsf{T}} \end{array}\right] \end{aligned} \tag{D.15}$$

We then arrive at the desired mean-value theorem result in the complex case:

$$g(z_o + \Delta z) \;-\; g(z_o) \;=\; 2\mathrm{Re}\left\{\left(\int_0^1 \nabla_z \, g(z_o + t\Delta z)dt\right)\Delta z\right\} \tag{D.16}$$

where we used the fact that for real-valued functions $g(z)$ it holds that

$$\nabla_{z^*} \, g(z) \;=\; [\nabla_z \, g(z)]^{*} \tag{D.17}$$

Expression (D.16) is the extension of (D.8) to the complex case.

Similarly, applying (D.6) to $h(v) = \nabla_{v^{\mathsf{T}}} \, g(v)$ we obtain that for any v_o and Δv:

$$\nabla_{v^{\mathsf{T}}} \, g(v_o + \Delta v) \;-\; \nabla_{v^{\mathsf{T}}} \, g(v_o) \;=\; \left(\int_0^1 \nabla_v^2 \, g(v_o + r\,\Delta v)dr\right)\Delta v \tag{D.18}$$

Multiplying from the left by $\frac{1}{2}D$ and using (C.30)–(C.31), as well as the fact that $\frac{1}{4}DH_v(v)D^{*} = H_c(u)$ (recall (B.26)), we find that relation (D.18) defined in terms of $\{v_o, \Delta v\}$ can be transformed into the

mean-value theorem relation (D.20) in terms of the variables $\{z_o, \Delta z\}$. Expression (D.20) is the extension of (D.9) to the complex case. Observe how both gradient vectors relative to z^* and z^{T} now appear in the relation. We show below in Example D.1 how the relation can be simplified in the special case when the Hessian matrix turns out to be block diagonal. In summary, we arrive at the following result.

Lemma D.2 (Mean-value theorem: Complex arguments)**.** Consider a real-valued and twice-differentiable function $g(z) \in \mathbb{R}$, where $z \in \mathbb{C}^M$ is complex-valued. Then, for any $M-$dimensional vectors z_o and Δz, the following increment equalities hold:

$$g(z_o + \Delta z) \;-\; g(z_o) = 2\mathrm{Re}\left\{\left(\int_0^1 \nabla_z\, g(z_o + t\,\Delta z)dt\right)\Delta z\right\} \tag{D.19}$$

$$\begin{bmatrix} \nabla_{z^*}\, g(z_o+\Delta z) \\ \nabla_{z^{\mathsf{T}}}\, g(z_o+\Delta z) \end{bmatrix} - \begin{bmatrix} \nabla_{z^*}\, g(z_o) \\ \nabla_{z^{\mathsf{T}}}\, g(z_o) \end{bmatrix} = \left(\int_0^1 \nabla_z^2\, g(z_o + r\Delta z)dr\right)\begin{bmatrix} \Delta z \\ (\Delta z^*)^{\mathsf{T}} \end{bmatrix} \tag{D.20}$$

Example D.1 (Block diagonal Hessian matrix)**.** Consider the real-valued quadratic function

$$g(z) \;=\; \kappa + a^*z + z^*a + z^*Cz \tag{D.21}$$

where κ is a real scalar, a is a column vector, and C is a Hermitian matrix. Then, the Hessian matrix of $g(z)$ is block diagonal and given by

$$\nabla_z^2\, g(z) \equiv H_c(u) \;=\; \begin{bmatrix} C & 0 \\ 0 & C^{\mathsf{T}} \end{bmatrix} \tag{D.22}$$

In this case, expression (D.20) decouples into two separate and equivalent relations. Keeping one of the relations we get

$$\nabla_z\, g(z_o + \Delta z) \;=\; \nabla_z\, g(z_o) \;+\; (\Delta z)^*\, C \tag{D.23}$$

Obviously, in this case, this relation could have been deduced directly from expression (D.21) by using the fact that

$$\nabla_z\, g(z) = a^* + z^*C \tag{D.24}$$

■

E

Lipschitz Conditions

Let $g(z) \in \mathbb{R}$ denote a *real-valued* $\nu-$strongly convex function of a possibly vector argument z. We assume that $g(z)$ is differentiable whenever necessary. In this appendix, we use the mean-value theorems from Appendix D to derive some useful bounds on the increments of strongly convex functions. These bounds will assist in analyzing the mean-square-error stability and performance of distributed algorithms. We treat both cases of real and complex arguments.

E.1 Perturbation Bounds in the Real Domain

Consider first the case in which the argument $z \in \mathbb{R}^M$ is real-valued. Let z^o denote the location of the unique global minimizer of $g(z)$ so that $\nabla_z g(z^o) = 0$. Combining the mean-value theorem results (D.8) and (D.9) we get

$$g(z^o + \Delta z) - g(z^o) = (\Delta z)^{\mathsf{T}} \left[\int_0^1 \int_0^1 t \, \nabla_z^2 \, g(z^o + tr \, \Delta z) dr dt \right] \Delta z \quad \text{(E.1)}$$

Now assume the Hessian matrix of $g(z)$ is uniformly bounded from above, i.e.,

$$\nabla_z^2 \, g(z) \; \leq \delta \, I_M, \quad \text{for all } z \qquad \text{(E.2)}$$

and for some $\delta > 0$. It follows from (E.1) that

$$g(z^o + \Delta z) \;-\; g(z^o) \;\leq\; \frac{\delta}{2}\,\|\Delta z\|^2 \tag{E.3}$$

which leads to the following useful statement for strongly-convex functions.

Lemma E.1 (Perturbation bound: Real arguments)**.** Consider a $\nu-$strongly convex and twice-differentiable function $g(z) \in \mathbb{R}$ and let $z^o \in \mathbb{R}^M$ denote its global minimizer. Assume that its $M \times M$ Hessian matrix (defined according to the first row in Table B.1 or equation (B.29)) is uniformly bounded from above by $\nabla_z^2\, g(z) \leq \delta I_M$, for all z and for some $\delta > 0$. We already know from item (c) in (C.18) that the same Hessian matrix is uniformly bounded from below by νI_M, i.e.,

$$\nu I_M \;\leq\; \nabla_z^2\, g(z) \;\leq\; \delta I_M, \quad \text{for all } z \tag{E.4}$$

Under condition (E.4), it follows from (C.16) and (E.3) that, for any Δz, the function increments are bounded by the squared Euclidean norm of Δz as follows:

$$\frac{\nu}{2}\,\|\Delta z\|^2 \;\leq\; g(z^o + \Delta z) \;-\; g(z^o) \;\leq\; \frac{\delta}{2}\,\|\Delta z\|^2 \tag{E.5}$$

One useful conclusion that follows from (E.5) is that under condition (E.4), every strongly convex function $g(z)$ can be sandwiched between two quadratic functions, namely,

$$g(z^o) + \frac{\nu}{2}\,\|z - z^o\|^2 \;\leq\; g(z) \;\leq\; g(z^o) + \frac{\delta}{2}\,\|z - z^o\|^2 \tag{E.6}$$

A second useful conclusion can be deduced from (E.1) when the size of Δz is small and when the Hessian matrix of $g(z)$ is smooth enough in a small neighborhood around $z = z^o$. Specifically, assume the Hessian matrix function is locally Lipschitz continuous in a small neighborhood around $z = z^o$, namely,

$$\left\|\nabla_z^2\, g(z^o + \Delta z) - \nabla_z^2\, g(z^o)\right\| \;\leq\; \kappa\,\|\Delta z\| \tag{E.7}$$

for sufficiently small values $\|\Delta z\| \leq \epsilon$ and for some $\kappa > 0$. This condition implies that we can write

$$\nabla_z^2\, g(z^o + \Delta z) \;=\; \nabla_z^2\, g(z^o) \;+\; O(\|\Delta z\|) \tag{E.8}$$

It then follows from equality (E.1) that, for sufficiently small Δz:

$$\begin{aligned} g(z^o + \Delta z) \;-\; g(z^o) \;&=\; (\Delta z)^{\mathsf{T}} \left[\frac{1}{2}\nabla_z^2\, g(z^o)\right] \Delta z \;+\; O(\|\Delta z\|^3) \\ &\approx\; (\Delta z)^{\mathsf{T}} \left[\frac{1}{2}\nabla_z^2\, g(z^o)\right] \Delta z \\ &=\; \|\Delta z\|^2_{\frac{1}{2}\nabla_z^2\, g(z^o)} \end{aligned} \tag{E.9}$$

where the symbol $\approx$ in the second line is used to indicate that higher-order powers in $\|\Delta z\|$ are being ignored. Moreover, for any Hermitian positive-definite weighting matrix $W > 0$, the notation $\|x\|^2_W$ refers to the weighted square Euclidean norm $x^* W x$.

We conclude from (E.9) that the increment in the value of the function in a small neighborhood around $z = z^o$ can be well approximated by means of a weighted square Euclidean norm; the weighting matrix in this case is equal to the Hessian matrix of $g(z)$ evaluated at $z = z^o$ and scaled by $1/2$. The error in this approximate evaluation is in the order of $\|\Delta z\|^3$.

Lemma E.2 (Perturbation approximation: Real arguments)**.** Consider the same setting of Lemma E.1 and assume additionally that the Hessian matrix function is locally Lipschitz continuous in a small neighborhood around $z = z^o$ as defined by (E.7). It then follows that the increment in the value of the function $g(z)$ for sufficiently small variations around $z = z^o$ can be well approximated by

$$g(z^o + \Delta z) \;-\; g(z^o) \;\approx\; \Delta z^{\mathsf{T}} \left[\frac{1}{2}\nabla_z^2\, g(z^o)\right] \Delta z \tag{E.10}$$

where the approximation error is in the order of $O(\|\Delta z\|^3)$.

Example E.1 (Quadratic cost functions with real arguments)**.** Consider a quadratic function of the form

$$g(z) \;=\; \kappa - a^{\mathsf{T}} z - z^{\mathsf{T}} a + z^{\mathsf{T}} C z \tag{E.11}$$

where κ is a scalar, a is a column vector, and C is a symmetric positive-definite matrix. It is straightforward to verify, by expanding the right-hand side in the expression below, that $g(z)$ can also be written as

$$g(z) \;=\; \kappa - a^{\mathsf{T}} C^{-1} a \;+\; (z - C^{-1}a)^{\mathsf{T}} C (z - C^{-1}a) \tag{E.12}$$

The Hessian matrix is $\nabla_z^2\, g(z) = 2C$ and it is clear that

$$2\lambda_{\min}(C)\, I_M \;\leq\; \nabla_z^2\, g(z) \;\leq\; 2\lambda_{\max}(C)\, I_M \tag{E.13}$$

in terms of the smallest and largest eigenvalues of C, which are both positive. Therefore, condition (E.4) is automatically satisfied with

$$\nu \;=\; 2\lambda_{\min}(C), \qquad \delta \;=\; 2\lambda_{\max}(C) \tag{E.14}$$

Likewise, condition (E.7) is obviously satisfied since the Hessian matrix in this case is constant and independent of z. The function $g(z)$ has a unique global minimizer and it occurs at the point $z = z^o$ where $\nabla_z\, g(z^o) = 0$. We know from the expression for $g(z)$ that

$$\nabla_z\, g(z) \;=\; -2a^{\mathsf{T}} + 2z^{\mathsf{T}}C \tag{E.15}$$

so that $z^o = C^{-1}a$ and

$$g(z^o) = \kappa - a^{\mathsf{T}}C^{-1}a \tag{E.16}$$

Therefore, applying (E.6) we conclude that

$$g(z^o) \;+\; \lambda_{\min}(C)\,\|z - C^{-1}a\|^2 \;\leq\; g(z) \;\leq\; g(z^o) + \lambda_{\max}(C)\,\|z - C^{-1}a\|^2 \tag{E.17}$$

Note that we could have arrived at this result directly from (E.12) as well.

Moreover, from result (E.10) we would estimate that, for sufficiently small $\|\Delta z\|$,

$$g(z^o + \Delta z) \;-\; g(z^o) \;\approx\; \|\Delta z\|_C^2 \tag{E.18}$$

Actually, in this case, exact equality holds in (E.18) for *any* Δz due to the quadratic nature of the function $g(z)$. Indeed, note from (E.12) that

$$g(z) = g(z^o) \;+\; \|z - z^o\|_C^2 \tag{E.19}$$

so that if we set $z = z^o + \Delta z$, for *any* Δz, the above relation gives

$$g(z^o + \Delta z) \;-\; g(z^o) \;=\; \|\Delta z\|_C^2, \quad \text{for any } \Delta z \tag{E.20}$$

which is a stronger result than (E.18); note in particular that Δz does not need to be infinitesimally small any more, as was the case with (E.10); this latter relation is useful for more general choices of $g(z)$ that are not necessarily quadratic in z.

■

E.2 Lipschitz Conditions in the Real Domain

The statement of Lemma E.1 requires the Hessian matrix to be upper bounded as in (E.2), i.e., $\nabla_z^2\, g(z) \leq \delta I_M$ for all z. For differentiable convex functions, this condition is equivalent to requiring the gradient vector to be Lipschitz continuous, i.e., to satisfy

$$\|\nabla_z\, g(z_2) - \nabla_z\, g(z_1)\| \;\leq\; \delta\,\|z_2 - z_1\| \tag{E.21}$$

for all z_1 and z_2. Since it is customary in the literature to rely more frequently on Lipschitz conditions, the following statement establishes the equivalence of conditions (E.2) and (E.21) for general convex functions (that are not necessarily strongly-convex). One advantage of using condition (E.21) instead of (E.2) is that the function $g(z)$ would not need to be twice-differentiable since condition (E.21) only involves the gradient vector of the function.

Lemma E.3 (Lipschitz and bounded Hessian matrix)**.** Consider a real-valued and twice-differentiable convex function $g(z) \in \mathbb{R}$. Then, the following two conditions are equivalent:

$$\nabla_z^2 g(z) \leq \delta\, I_M, \text{ for all } z \iff \|\nabla_z g(z_2) - \nabla_z g(z_1)\| \leq \delta\,\|z_2 - z_1\|, \text{ for all } z_1, z_2 \tag{E.22}$$

Proof. Assume first that the Hessian matrix, $\nabla_z^2\, g(z)$, is uniformly upper bounded by $\delta\, I_M$ for all z; we know that it is nonnegative definite since $g(z)$ is convex and, therefore, $\nabla_z^2\, g(z)$ is lower bounded by zero. We pick any z_1 and z_2 and introduce the column vector function $h(z) = \nabla_{z^{\mathsf{T}}}\, g(z)$. Applying (D.8) to $h(z)$ gives

$$h(z_2) \;-\; h(z_1) \;=\; \left(\int_0^1 \nabla_z\, h(z_1 + t(z_2 - z_1))dt\right)(z_2 - z_1) \tag{E.23}$$

so that using $0 \leq \nabla_z^2\, g(z) \leq \delta\, I_M$, we get

$$\|\nabla_{z^{\mathsf{T}}}\, g(z_2) \;-\; \nabla_{z^{\mathsf{T}}}\, g(z_1)\| \;\leq\; \left(\int_0^1 \delta dt\right)\|z_2 - z_1\| \tag{E.24}$$

and we arrive at the Lipschitz condition on the right-hand side of (E.22) since $\nabla_{z^{\mathsf{T}}}\, g(z) = [\nabla_z\, g(z)]^{\mathsf{T}}$.

Conversely, assume the Lipschitz condition on the right-hand side of (E.22) holds, and introduce the column vector function $f(t) = \nabla_{z^\mathsf{T}}\, g(z+t\Delta z)$ defined in terms of a scalar real parameter t. Then,

$$\frac{df(t)}{dt} \;=\; \left[\nabla_z^2\, g(z+t\Delta z)\right]\Delta z \tag{E.25}$$

Now, for any Δt and in view of the Lipschitz condition, it holds that

$$\begin{aligned}\|f(t+\Delta t) \;-\; f(t)\| \;&=\; \|\nabla_{z^\mathsf{T}}\, g(z+(t+\Delta t)\Delta z) \;-\; \nabla_{z^\mathsf{T}}\, g(z+t\Delta z)\| \\ &\leq\; \delta\,|\Delta t|\,\|\Delta z\|\end{aligned} \tag{E.26}$$

so that

$$\underbrace{\lim_{\Delta t\to 0}\;\frac{\|f(t+\Delta t)\;-\;f(t)\|}{|\Delta t|}}_{=\|df(t)/dt\|} \;\leq\; \delta\,\|\Delta z\| \tag{E.27}$$

Using (E.25) we conclude that

$$\left\|\left[\nabla_z^2\, g(z+t\Delta z)\right]\,\Delta z\right\| \;\leq\; \delta\,\|\Delta z\|, \quad \text{for any } t, z \text{ and } \Delta z \tag{E.28}$$

Setting $t=0$, squaring both sides, and recalling that the Hessian matrix is symmetric, we obtain

$$(\Delta z)^\mathsf{T}\left[\nabla_z^2\, g(z)\right]^2\Delta z \;\leq\; \delta^2\|\Delta z\|^2, \quad \text{for any } z, \Delta z \tag{E.29}$$

from which we conclude that $\nabla_z^2\, g(z) \leq \delta\, I_M$ for all z, as desired.

□

We can additionally verify that the local Lipschitz condition (E.7) used in Lemma E.2 is actually equivalent to a global Lipschitz property on the Hessian matrix under condition (E.4).

Lemma E.4 (Global Lipschitz condition)**.** Consider a real-valued and twice-differentiable $\nu-$strongly convex function $g(z) \in \mathbb{R}$ and assume it satisfies conditions (E.4) and (E.7). It then follows that the Hessian matrix of $g(z)$ is globally Lipschitz relative to z^o, namely, it satisfies

$$\|\nabla_z^2\, g(z) - \nabla_z^2\, g(z^o)\| \;\leq\; \kappa'\|z-z^o\|, \quad \text{for all } \; z \tag{E.30}$$

where the positive scalar κ' is defined in terms of the parameters $\{\kappa, \delta, \nu, \epsilon\}$ as

$$\kappa' \;=\; \max\left\{\kappa, \frac{\delta-\nu}{\epsilon}\right\} \tag{E.31}$$

Proof. Following [277], for any vector x, it holds that

$$\begin{aligned} x^{\mathsf{T}}\left[\nabla_z^2 g(z) - \nabla_z^2 g(z^o)\right] x &= x^{\mathsf{T}}\nabla_z^2 g(z)x \;-\; x^{\mathsf{T}}\nabla_z^2 g(z^o)x \\ &\overset{(E.4)}{\leq} \delta\|x\|^2 - \nu\|x\|^2 \\ &= (\delta-\nu)\,\|x\|^2 \end{aligned} \tag{E.32}$$

And since the Hessian matrix difference is symmetric, we conclude that $\nabla_z^2 g(z) - \nabla_z^2 g(z^o) \leq (\delta-\nu)I_M$ so that, in terms of the 2−induced norm:

$$\|\nabla_z^2 g(z) - \nabla_z^2 g(z^o)\| \;\leq\; \delta - \nu \tag{E.33}$$

Now, consider any vector z such that $\|z - z^o\| \leq \epsilon$. Then,

$$\|\nabla_z^2 g(z) - \nabla_z^2 g(z^o)\| \overset{(E.7)}{\leq} \kappa\,\|z - z^o\| \overset{(E.31)}{\leq} \kappa'\|z - z^o\| \tag{E.34}$$

On the other hand, for any vector z such that $\|z - z^o\| > \epsilon$, we have

$$\|\nabla_z^2 g(z) - \nabla_z^2 g(z^o)\| \overset{(E.33)}{\leq} \left(\frac{\delta-\nu}{\epsilon}\right)\epsilon \overset{(E.31)}{\leq} \kappa'\|z - z^o\| \tag{E.35}$$

□

E.3 Perturbation Bounds in the Complex Domain

The statement below extends the result of Lemma E.1 to the case of complex arguments, $z \in \mathbb{C}^M$. Comparing the bounds in (E.37) with the earlier result (E.5), we observe that the relations are identical. The only difference in the complex case relative to the real case is that the upper and lower bounds on the complex Hessian matrix in (E.36) are scaled by $1/2$ relative to the bounds in (E.4).

Lemma E.5 (Perturbation bound: Complex arguments)**.** Consider a ν−strongly convex and twice-differentiable function $g(z) \in \mathbb{R}$ and let $z^o \in \mathbb{C}^M$ denote its global minimizer. The function $g(z)$ is real-valued but z is now complex-valued. Assume that the $2M \times 2M$ complex Hessian matrix of $g(z)$ (defined according to the last row of Table B.1 and (B.29)) is uniformly bounded from above by $\nabla_z^2 g(z) \leq \frac{\delta}{2} I_{2M}$, for all z and for some $\delta > 0$. We already know from item (c) in (C.44) that the same Hessian matrix is uniformly bounded from below by $\frac{\nu}{2} I_{2M}$, i.e.,

$$\frac{\nu}{2}\,I_{2M} \;\leq\; \nabla_z^2 g(z) \;\leq\; \frac{\delta}{2}\,I_{2M}, \quad \text{for all } z \tag{E.36}$$

Under condition (E.36) it holds that, for any Δz, the function increments are bounded by the squared Euclidean norm of Δz as follows:

$$\frac{\nu}{2}\,\|\Delta z\|^2 \;\le\; g(z^o+\Delta z) \;-\; g(z^o) \;\le\; \frac{\delta}{2}\,\|\Delta z\|^2 \tag{E.37}$$

Proof. The argument is based on expressing z in terms of its real and imaginary parts, $z = x + jy$, transforming $g(z)$ into a function of the $2M \times 1$ extended real variable $v = \mathrm{col}\{x, y\}$, and then applying the result of Lemma E.1 to $g(v)$.

To begin with, recall that the $2M \times 2M$ Hessian matrix of $g(v)$ is denoted by $H(v)$ and is constructed according to the second row of Table B.1. This real Hessian matrix is related by (B.26) to the complex Hessian matrix, $H_c(u)$, of $g(z)$ and which we are denoting by $\nabla_z^2\, g(z)$ in the statement of the lemma. Therefore, the upper bound on $\nabla_z^2\, g(z)$ in (E.36) can be transformed into an upper bound on $H(v)$ by noting that

$$H(v) \overset{(B.26)}{=} D^*\left[\nabla_z^2\, g(z)\right] D \;\;\le\;\; \frac{\delta}{2}\, D^*D \;=\; \delta\, I_{2M} \tag{E.38}$$

since $D^*D = 2I_{2M}$ and, hence, $H(v) \le \delta I_{2M}$. Combining this result with (C.45) we conclude that the Hessian matrix $H(v)$ is bounded as follows:

$$\nu\, I_{2M} \;\le\; H(v) \;\le\; \delta\, I_{2M} \tag{E.39}$$

Consequently, if we apply the result of Lemma E.1 to the function $g(v)$, whose argument v is real, we find that

$$\frac{\nu}{2}\,\|\Delta v\|^2 \;\le\; g(v^o+\Delta v) \;-\; g(v^o) \;\le\; \frac{\delta}{2}\,\|\Delta v\|^2 \tag{E.40}$$

which is equivalent to the desired relation (E.37) in terms of the original variables $\{z^o, \Delta z\}$ since, for any z, $g(z) = g(v)$ and $\|z\| = \|v\|$.

□

One useful conclusion that follows from (E.37) is that under condition (E.36), the strongly convex function $g(z)$ can be sandwiched between two quadratic functions, namely,

$$g(z^o) + \frac{\nu}{2}\,\|z-z^o\|^2 \;\le\; g(z) \;\le\; g(z^o) + \frac{\delta}{2}\,\|z-z^o\|^2 \tag{E.41}$$

A second useful conclusion is an extension of (E.10) to the case of complex arguments z. Introduce the extended vector:

$$\Delta z^e \;\triangleq\; \begin{bmatrix} \Delta z \\ (\Delta z^*)^{\mathsf{T}} \end{bmatrix} \tag{E.42}$$

Lemma E.6 (Perturbation approximation: Complex arguments)**.** Consider the same setting of Lemma E.5 and assume additionally that the Hessian matrix function is locally Lipschitz continuous in a small neighborhood around $z = z^o$, namely,

$$\left\|\nabla_z^2\, g(z^o + \Delta z) - \nabla_z^2\, g(z^o)\right\| \;\leq\; \kappa\,\|\Delta z\| \tag{E.43}$$

for sufficiently small values $\|\Delta z\| \leq \epsilon$ and for some $\kappa > 0$. It then follows that the increment in the value of the function $g(z)$ for small variations around $z = z^o$ can be well approximated by:

$$g(z^o + \Delta z) \;-\; g(z^o) \;\approx\; (\Delta z^e)^*\left[\frac{1}{2}\nabla_z^2\, g(z^o)\right]\Delta z^e \tag{E.44}$$

where the approximation error is in the order of $O(\|\Delta z\|^3)$.

Proof. Result (E.44) can be derived from (E.10) as follows. We again transform $g(z)$ into the function $g(v)$ of the real variable $v = \mathrm{col}\{x, y\}$ and then apply (E.10) to $g(v)$ for sufficiently small Δv, which gives

$$g(v^o + \Delta v) \;-\; g(v^o) \;\approx\; (\Delta v)^{\mathsf{T}}\left[\frac{1}{2}H(v^o)\right]\Delta v, \quad \text{as } \Delta v \to\; 0 \tag{E.45}$$

in terms of the $2M \times 2M$ Hessian matrix of $g(v)$ evaluated at $v = v^o$. This Hessian matrix is related to the complex Hessian matrix $H_c(u^o)$ according to (B.26). Thus, observe that

$$\begin{aligned}
(\Delta v)^{\mathsf{T}}\left[\frac{1}{2}H(v^o)\right]\Delta v \;&=\; \frac{1}{4}\,(\Delta v)^{\mathsf{T}}\,D^*D\left[\frac{1}{2}H(v^o)\right]D^*D\,\Delta v \\
&\overset{(D.13)}{=}\; \frac{1}{2}\,\underbrace{(\Delta v)^{\mathsf{T}}\,D^*}_{(\Delta u)^*}\,\underbrace{\frac{1}{4}DH(v^o)D^*}_{H_c(u^o)}\,\underbrace{D\Delta v}_{\Delta u} \\
&\overset{(B.26)}{=}\; \frac{1}{2}\,(\Delta u)^*\,H_c(u^o)\Delta u \\
&\overset{(B.24)}{=}\; \frac{1}{2}\begin{bmatrix}(\Delta z)^* & \Delta z^{\mathsf{T}}\end{bmatrix}\nabla_z^2\, g(z^o)\begin{bmatrix}\Delta z \\ (\Delta z^*)^{\mathsf{T}}\end{bmatrix} \\
&=\; (\Delta z^e)^*\left[\frac{1}{2}\nabla_z^2\, g(z^o)\right]\Delta z^e
\end{aligned} \tag{E.46}$$

as claimed. □

Example E.2 (Quadratic cost functions with complex arguments)**.** Let us illustrate the above result by considering a quadratic function of the form

$$g(z) \;=\; \kappa - a^*z - z^*a + z^*Cz \tag{E.47}$$

where κ is a scalar, a is a column vector, and C is a Hermitian positive-definite matrix. It is straightforward to verify, by expanding the right-hand side in the expression below, that $g(z)$ can be also written as

$$g(z) \;=\; \kappa - a^*C^{-1}a \;+\; (z - C^{-1}a)^*C(z - C^{-1}a) \tag{E.48}$$

The Hessian matrix in this case is $2M \times 2M$ and given by

$$\nabla_z^2\, g(z) \;=\; \begin{bmatrix} C & 0 \\ 0 & C^{\mathsf{T}} \end{bmatrix} \tag{E.49}$$

It is clear that

$$\lambda_{\min}(C)\, I_{2M} \;\leq\; \nabla_z^2\, g(z) \;\leq\; \lambda_{\max}(C)\, I_{2M} \tag{E.50}$$

in terms of the smallest and largest eigenvalues of C, which are both positive. Therefore, condition (E.36) is automatically satisfied with

$$\nu \;=\; 2\lambda_{\min}(C), \qquad \delta \;=\; 2\lambda_{\max}(C) \tag{E.51}$$

Likewise, condition (E.43) is satisfied since the Hessian matrix is constant and independent of z. The function $g(z)$ has a unique global minimizer and it occurs at the point $z = z^o$ where $\nabla_z\, g(z^o) = 0$. We know from expression (E.48) for $g(z)$ that $z^o = C^{-1}a$ and $g(z^o) = \kappa - a^*C^{-1}a$. Therefore, applying (E.41) we conclude that

$$g(z^o) + \lambda_{\min}(C)\,\|z - C^{-1}a\|^2 \;\leq\; g(z) \;\leq\; g(z^o) + \lambda_{\max}(C)\,\|z - C^{-1}a\|^2 \tag{E.52}$$

Note that we could have arrived at this result directly from (E.48) as well.

Moreover, we would estimate from (E.44) that

$$\begin{aligned} g(z^o + \Delta z) \;-\; g(z^o) \;&\approx\; \frac{1}{2}\begin{bmatrix} (\Delta z)^* & (\Delta z)^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} C & 0 \\ 0 & C^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} \Delta z \\ (\Delta z^*)^{\mathsf{T}} \end{bmatrix} \\ &=\; \|\Delta z\|_C^2 \end{aligned} \tag{E.53}$$

where the notation $\|x\|_C^2$ now denotes the squared Euclidean quantity x^*Cx. Actually, in this case, exact equality holds in (E.53) for *any* Δz due to the quadratic nature of the function $g(z)$. Indeed, note that (E.48) can be rewritten as

$$g(z) = g(z^o) \;+\; \|z - z^o\|_C^2 \tag{E.54}$$

so that if we set $z = z^o + \Delta z$, for *any* Δz, the above relation gives

$$g(z^o + \Delta z) \;-\; g(z^o) \;=\; \|\Delta z\|_C^2, \quad \text{for any } \Delta z \tag{E.55}$$

which is a stronger result than the approximation in (E.53); note in particular that Δz does not need to be infinitesimally small any more, as was the case with (E.44); this latter result is applicable to more general choices of $g(z)$ that are not necessarily quadratic in z.

■

E.4 Lipschitz Conditions in the Complex Domain

The statement of Lemma E.5 requires the Hessian matrix to be upper bounded as in (E.36), i.e., $\nabla_z^2\, g(z) \leq \frac{\delta}{2} I_{2M}$ for all z. As was the case with real arguments in Lemma E.3, we can argue that for general convex functions (that are not necessarily strongly convex), this condition is equivalent to requiring the gradient vector to be Lipschitz continuous.

Lemma E.7 (Lipschitz and bounded Hessian matrix)**.** Consider a real-valued and twice-differentiable convex function $g(z) \in \mathbb{R}$, where $z \in \mathbb{C}^M$ is now complex valued. Then, the following two conditions are equivalent:

$$\nabla_z^2 g(z) \leq \frac{\delta}{2} I_{2M}, \text{ for all } z \iff \|\nabla_z g(z_2) - \nabla_z g(z_1)\| \leq \frac{\delta}{2}\|z_2 - z_1\|, \text{ for all } z_1, z_2 \tag{E.56}$$

Proof. The above result can be derived from (E.22) as follows. We transform $g(z)$ into the function $g(v)$ of the real variable $v = \text{col}\{x, y\}$, where $z = x + jy$, and then apply (E.22) to $g(v)$.

First, recall from the argument that led to (E.39) that the complex Hessian matrix of $g(z)$ is bounded by $\frac{\delta}{2} I_{2M}$ if, and only if, the real Hessian matrix of $g(v)$ is bounded by $\delta\, I_{2M}$. Using this observation and applying (E.22) to $g(v)$ we get

$$\begin{aligned} \nabla_z^2\, g(z) \;\leq\; \frac{\delta}{2}\, I_{2M} \quad &\overset{(E.39)}{\iff} \quad \nabla_v^2\, g(v) \;\leq\; \delta\, I_{2M}, \quad \text{for all } v \\ &\overset{(E.22)}{\iff} \quad \|\nabla_v\, g(v_2) - \nabla_v\, g(v_1)\| \;\leq\; \delta\, \|v_2 - v_1\| \end{aligned} \tag{E.57}$$

for any v_1, v_2. Now we know from (C.32) that

$$\frac{1}{2}\,[\nabla_v\, g(v)]\, D^* \;=\; \begin{bmatrix} \nabla_z\, g(z) & (\nabla_{z^*}\, g(z))^{\mathsf{T}} \end{bmatrix} \tag{E.58}$$

Recalling from (B.28) that the matrix $D^*/\sqrt{2}$ is unitary, we get

$$\begin{aligned}
\|\nabla_v\, g(v_2) - \nabla_v\, g(v_1)\| \;&=\; \qquad\qquad (E.59)\\
&= \left\| [\nabla_v\, g(v_2) - \nabla_v\, g(v_1)] \cdot \frac{D^*}{\sqrt{2}} \right\| \\
&\overset{(E.58)}{=} \sqrt{2} \cdot \left\| \begin{bmatrix} \nabla_z\, g(z_2) - \nabla_z\, g(z_1) & (\nabla_{z^*}\, g(z_2) - \nabla_{z^*}\, g(z_1))^{\mathsf{T}} \end{bmatrix} \right\| \\
&= 2 \cdot \|\nabla_z\, g(z_2) - \nabla_z\, g(z_1)\|
\end{aligned}$$

where we used (D.17). Noting that $\|v_2 - v_1\| = \|z_2 - z_1\|$ and substituting into (E.57) we conclude that

$$\nabla_z^2\, g(z) \;\le\; \frac{\delta}{2} I_{2M} \iff \|\nabla_z\, g(z_2) - \nabla_z\, g(z_1)\| \;\le\; \frac{\delta}{2}\|z_2 - z_1\|, \quad \text{for all } z_1, z_2 \tag{E.60}$$

as claimed.

□

We can again verify that the local Lipschitz condition (E.43) used in Lemma E.6 is equivalent to a global Lipschitz property on the Hessian matrix under the bounds (E.36). The proof of the following result is similar to that of Lemma E.4.

Lemma E.8 (Global Lipschitz condition)**.** Consider a real-valued and twice-differentiable $\nu-$strongly convex function $g(z) \in \mathbb{R}$ and assume it satisfies conditions (E.36) and (E.43). It then follows that the $2M \times 2M$ Hessian matrix of $g(z)$ is globally Lipschitz relative to $z^o \in \mathbb{C}^M$, namely,

$$\|\nabla_z^2\, g(z) - \nabla_z^2\, g(z^o)\| \;\le\; \kappa' \|z - z^o\|, \quad \text{for all } \; z \tag{E.61}$$

where the positive scalar κ' is defined in terms of the parameters $\{\kappa, \delta, \nu, \epsilon\}$ as

$$\kappa' \;=\; \max\left\{\kappa, \frac{\delta - \nu}{2\epsilon}\right\} \tag{E.62}$$

F

Useful Matrix and Convergence Results

We collect in this appendix several useful matrix properties and convergence results that are called upon in the text.

F.1 Kronecker Products

Traditional Kronecker Form

Let $A = [a_{ij}]_{i,j=1}^{n}$ and $B = [b_{ij}]_{i,j=1}^{m}$ be $n \times n$ and $m \times m$ possibly complex-valued matrices, respectively, whose individual $(i,j)-$th entries are denoted by a_{ij} and b_{ij}. Their Kronecker product is denoted by $K = A \otimes B$ and is defined as the $nm \times nm$ matrix whose entries are given by [104, 113]:

$$K \triangleq A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \ldots & a_{1n}B \\ a_{21}B & a_{22}B & \ldots & a_{2n}B \\ \vdots & & \vdots & \\ a_{n1}B & a_{n2}B & \ldots & a_{nn}B \end{bmatrix} \tag{F.1}$$

In other words, each scalar entry a_{ij} of A is replaced by a block quantity that is equal to a scaled multiple of B, namely, $a_{ij}B$.

Let $\{\lambda_i(A), i = 1, \ldots, n\}$ and $\{\lambda_j(B), j = 1, \ldots, m\}$ denote the eigenvalues of A and B, respectively. Then, the eigenvalues of $A \otimes B$

will consist of all nm product combinations $\{\lambda_i(A)\lambda_j(B)\}$. A similar conclusion holds for the singular values of $A \otimes B$ in relation to the singular values of the individual matrices A and B, which we denote by $\{\sigma_i(A), \sigma_j(B)\}$. Table F.1 lists some well-known properties of Kronecker products for matrices $\{A, B, C, D\}$ of compatible dimensions and column vectors $\{x, y\}$. The last three properties involve the trace and vec operations: the trace of a matrix is the sum of its diagonal elements and the vec operation transforms a matrix into a vector by stacking the columns of the matrix on top of each other.

Table F.1: Properties of the traditional Kronecker product definition (F.1).

1.	$(A+B)\otimes C = (A\otimes C) \;+\; (B\otimes C)$
2.	$(A\otimes B)(C\otimes D) = (AC\otimes BD)$
3.	$(A\otimes B)^{\mathsf{T}} = A^{\mathsf{T}}\otimes B^{\mathsf{T}}$
4.	$(A\otimes B)^{*} = A^{*}\otimes B^{*}$
5.	$(A\otimes B)^{-1} = A^{-1}\otimes B^{-1}$
6.	$(A\otimes B)^{\ell} = A^{\ell}\otimes B^{\ell}$
7.	$\{\lambda(A\otimes B)\} = \{\lambda_i(A)\lambda_j(B)\}_{i=1,j=1}^{n,m}$
8.	$\{\sigma(A\otimes B)\} = \{\sigma_i(A)\sigma_j(B)\}_{i=1,j=1}^{n,m}$
9.	$\det(A\otimes B) = (\det A)^m(\det B)^n$
10.	$\mathrm{Tr}(A\otimes B) = \mathrm{Tr}(A)\mathrm{Tr}(B)$
11.	$\mathrm{Tr}(AB) = \big[\mathrm{vec}(B^{\mathsf{T}})\big]^{\mathsf{T}}\,\mathrm{vec}(A) = [\mathrm{vec}(B^{*})]^{*}\,\mathrm{vec}(A)$
12.	$\mathrm{vec}(ACB) = (B^{\mathsf{T}}\otimes A)\mathrm{vec}(C)$
13.	$\mathrm{vec}(xy^{\mathsf{T}}) = y\otimes x$

Block Kronecker Form

Let $\mathcal{A}$ now denote a *block* matrix of size $np\times np$ with each block having size $p\times p$. We denote the $(i,j)-$th sub-matrix of $\mathcal{A}$ by the notation A_{ij}; it is a block of size $p\times p$. Likewise, we let $\mathcal{B}$ denote a second *block* matrix of size $mp\times mp$ with each of its blocks having the same size $p\times p$. We denote the $(i,j)-$th sub-matrix of $\mathcal{B}$ by the notation B_{ij}; it is a block of size $p\times p$. The *block* Kronecker product of these two matrices is denoted by $\mathcal{K} = \mathcal{A}\otimes_b\mathcal{B}$ and is defined as the following block matrix

of dimensions $nmp^2 \times mnp^2$ [145]:

$$\mathcal{K} \triangleq \mathcal{A} \otimes_b \mathcal{B} = \begin{bmatrix} K_{11} & K_{12} & \ldots & K_{1n} \\ K_{21} & K_{22} & \ldots & K_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ K_{n1} & K_{n2} & \ldots & K_{nn} \end{bmatrix} \tag{F.2}$$

where each block K_{ij} is $mp^2 \times mp^2$ and is constructed as follows:

$$K_{ij} = \begin{bmatrix} A_{ij} \otimes B_{11} & A_{ij} \otimes B_{12} & \ldots & A_{ij} \otimes B_{1m} \\ A_{ij} \otimes B_{21} & A_{ij} \otimes B_{22} & \ldots & A_{ij} \otimes B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{ij} \otimes B_{m1} & A_{ij} \otimes B_{m2} & \ldots & A_{ij} \otimes B_{mm} \end{bmatrix} \tag{F.3}$$

Table F.2 lists some useful properties of block Kronecker products for matrices $\{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}\}$ with blocks of size $p \times p$. The last three properties involve the block vectorization operation denoted by bvec: it vectorizes each block entry of the matrix and then stacks the resulting columns on top of each other, i.e.,

$$\begin{aligned} \text{bvec}(\mathcal{A}) \quad \triangleq \quad & \text{col}\,\{\text{vec}(A_{11}), \text{vec}(A_{21}), \ldots, \text{vec}(A_{n1}), \\ & \text{vec}(A_{21}), \text{vec}(A_{22}), \ldots, \text{vec}(A_{n2}), \\ & \qquad \vdots \\ & \text{vec}(A_{1n}), \text{vec}(A_{2n}), \ldots, \text{vec}(A_{nn})\} \end{aligned} \tag{F.4}$$

Table F.2: Properties of the block Kronecker product definition (F.2).

1.	$(\mathcal{A} + \mathcal{B}) \otimes_b \mathcal{C} = (\mathcal{A} \otimes_b \mathcal{C}) \; + \; (\mathcal{B} \otimes_b \mathcal{C})$
2.	$(\mathcal{A} \otimes_b \mathcal{B})(\mathcal{C} \otimes_b \mathcal{D}) = (\mathcal{AC} \otimes_b \mathcal{BD})$
3.	$(A \otimes B) \otimes_b (C \otimes D) = (A \otimes C) \otimes (B \otimes D)$
4.	$(\mathcal{A} \otimes_b \mathcal{B})^{\mathsf{T}} = \mathcal{A}^{\mathsf{T}} \otimes_b \mathcal{B}^{\mathsf{T}}$
5.	$(\mathcal{A} \otimes_b \mathcal{B})^{*} = \mathcal{A}^{*} \otimes_b \mathcal{B}^{*}$
6.	$\{\lambda(\mathcal{A} \otimes_b \mathcal{B})\} = \{\lambda_i(\mathcal{A})\lambda_j(\mathcal{B})\}_{i=1,j=1}^{np,mp}$
7.	$\text{Tr}(\mathcal{AB}) = \left[\text{bvec}(\mathcal{B}^{\mathsf{T}})\right]^{\mathsf{T}} \text{bvec}(\mathcal{A}) = [\text{bvec}(\mathcal{B}^{*})]^{*} \text{bvec}(\mathcal{A})$
8.	$\text{bvec}(\mathcal{ACB}) = (\mathcal{B}^{\mathsf{T}} \otimes_b \mathcal{A})\text{bvec}(\mathcal{C})$
9.	$\text{bvec}(xy^{\mathsf{T}}) = y \otimes_b x$

Figure F.1 illustrates one of the advantages of working with the bvec operation for block matrices [278]. The figure compares the effect of the block vectorization operation to that of the regular vec operation. It is seen that bvec preserves the locality of the blocks from the original matrix: entries arising from the same block appear together followed by entries of the other successive blocks. In contrast, in the vec construction, entries from different blocks are blended together.

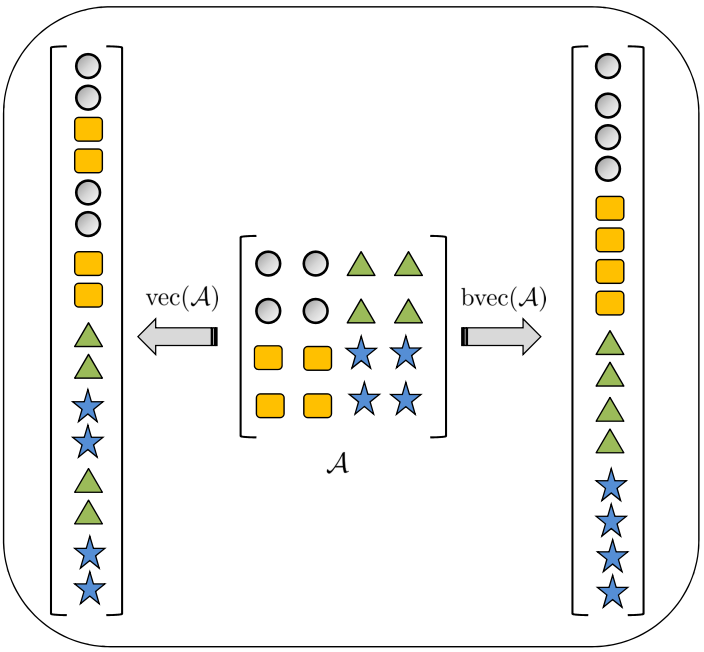

Figure F.1: Schematic comparison of the regular and block vectorization operations. It is seen that the bvec operation preserves the locality of the blocks from the original matrix, while the entries of the blocks get mixed up in the regular vec operation.

F.2 Vector and Matrix Norms

Vector Norms

For any vector x of size $N \times 1$ and entries $\{x_k\}$, any of the definitions

listed in Table F.3 constitutes a valid vector norm.

Table F.3: Useful vector norms, where the $\{x_k\}$ denote the entries of $x \in \mathbb{C}^N$.

$\|x\|_1 \triangleq \sum_{k=1}^{N} \|x_k\|$	(1−norm)
$\|x\|_\infty \triangleq \max_{1\le k\le N} \|x_k\|$	(∞−norm)
$\|x\|_2 \triangleq \left(\sum_{k=1}^{N} \|x_k\|^2\right)^{1/2}$	(Euclidean norm)
$\|x\|_p \triangleq \left(\sum_{k=1}^{N} \|x_k\|^p\right)^{1/p}$	(p−norm, for any real $p \ge 1$)

Matrix Norms

There are similarly many useful matrix norms. For any matrix A of dimensions $N \times N$ and entries $\{a_{\ell k}\}$, any of the definitions listed in Table F.4 constitutes a valid matrix norm. In particular, the 2−induced norm of A is equal to its largest singular value:

$$\|A\|_2 \;=\; \sigma_{\max}(A) \tag{F.5}$$

Table F.4: Useful matrix norms, where the $\{a_{\ell k}\}$ denote the entries of $A \in \mathbb{C}^{N\times N}$.

$\|A\|_1 \triangleq \max_{1\le k\le N} \left(\sum_{\ell=1}^{N} \|a_{\ell k}\|\right)$	(1−norm, or maximum absolute column sum)
$\|A\|_\infty \triangleq \max_{1\le \ell\le N} \left(\sum_{k=1}^{N} \|a_{\ell k}\|\right)$	(∞−norm, or maximum absolute row sum)
$\|A\|_{\mathrm{F}} \triangleq \sqrt{\mathrm{Tr}(A^* A)}$	(Frobenius norm)
$\|A\|_p \triangleq \max_{x\ne 0} \left(\frac{\|Ax\|_p}{\|x\|_p}\right)$	(p−induced norm for any real $p \ge 1$)

A fundamental result in matrix theory is that all matrix norms in finite dimensional spaces are *equivalent*. Specifically, if $\|A\|_a$ and $\|A\|_b$ denote

two generic matrix norms, then there exist positive constants c_ℓ and c_u that bound one norm by the other from above and from below such as [104, 113]:

$$c_\ell \, \|A\|_b \;\le\; \|A\|_a \;\le\; c_u \, \|A\|_b \tag{F.6}$$

The values of $\{c_\ell, c_u\}$ are independent of the matrix entries though they may be dependent on the matrix dimensions. Vector norms are also equivalent to each other.

One Useful Matrix Norm

Let B denote an $N \times N$ matrix with eigenvalues $\{\lambda_k\}$. The spectral radius of B, denoted by $\rho(B)$, is defined as

$$\rho(B) \;\triangleq\; \max_{1 \le k \le N} \; |\lambda_k| \tag{F.7}$$

We introduce the Jordan canonical decomposition of B and write $B = TJT^{-1}$, where T is an invertible transformation and J is a block diagonal matrix, say, with q blocks:

$$J = \text{diag}\{J_1, J_2, \ldots, J_q\} \tag{F.8}$$

Each block J_q has a Jordan structure with an eigenvalue λ_q on its diagonal entries, unit entries on the first sub-diagonal, and zeros everywhere else. For example, for a block of size 4×4:

$$J_q = \begin{bmatrix} \lambda_q & & & \\ 1 & \lambda_q & & \\ & 1 & \lambda_q & \\ & & 1 & \lambda_q \end{bmatrix} \tag{F.9}$$

Let ϵ denote an arbitrary positive scalar that we are free to choose and define the $N \times N$ diagonal scaling matrix:

$$D \;\triangleq\; \text{diag}\left\{\epsilon, \epsilon^2, \ldots, \epsilon^N\right\} \tag{F.10}$$

Following Lemma 5.6.10 from [113] and Problem 14.19 from [133], we can use the quantity T originating from B to define the following matrix norm, denoted by $\|\cdot\|_\rho$, for any matrix A of size $N \times N$:

$$\|A\|_\rho \;\triangleq\; \left\|DT^{-1}ATD^{-1}\right\|_1 \tag{F.11}$$

in terms of the $1-$norm (i.e., maximum absolute column sum) of the matrix product on the right-hand side. It is not difficult to verify that the transformation (F.11) is a valid matrix norm, namely, that it satisfies the following properties, for any matrices A and C of compatible dimensions and for any complex scalar α:

$$\begin{cases} \text{(a) } \|A\|_\rho \geq 0 \text{ with } \|A\|_\rho = 0 \text{ if, and only if, } A = 0 \\ \text{(b) } \|\alpha A\|_\rho = |\alpha| \, \|A\|_\rho \\ \text{(c) } \|A + C\|_\rho \leq \|A\|_\rho \; + \; \|C\|_\rho \text{ (triangular inequality)} \\ \text{(d) } \|AC\|_\rho \leq \|A\|_\rho \, \|C\|_\rho \text{ (sub-multiplicative property)} \end{cases} \tag{F.12}$$

One important property of the $\rho-$norm defined by (F.11) is that when it is applied to the matrix B itself, it will hold that:

$$\rho(B) \; \leq \; \|B\|_\rho \; \leq \; \rho(B) + \epsilon \tag{F.13}$$

That is, the $\rho-$norm of B will be sandwiched between two bounds defined by its spectral radius. It follows that if the matrix B is stable to begin with, so that $\rho(B) < 1$, then we can always select ϵ small enough to ensure $\|B\|_\rho < 1$.

The matrix norm defined by (F.11) is also an induced norm relative to the following vector norm:

$$\|x\|_\rho \; \stackrel{\Delta}{=} \; \|DT^{-1}x\|_1 \tag{F.14}$$

That is, for any matrix A, it holds that

$$\|A\|_\rho \; = \; \max_{x \neq 0} \left(\frac{\|Ax\|_\rho}{\|x\|_\rho} \right) \tag{F.15}$$

Proof. Indeed, using (F.14), we first note that for any vector $x \neq 0$:

$$\begin{aligned} \|Ax\|_\rho &= \|DT^{-1}Ax\|_1 \\ &= \|DT^{-1}A \cdot TD^{-1}DT^{-1} \cdot x\|_1 \\ &\leq \|DT^{-1}ATD^{-1}\|_1 \cdot \|DT^{-1}x\|_1 \\ &= \|A\|_\rho \cdot \|x\|_\rho \end{aligned} \tag{F.16}$$

so that

$$\max_{x \neq 0} \left(\frac{\|Ax\|_\rho}{\|x\|_\rho} \right) \; \leq \; \|A\|_\rho \tag{F.17}$$

To show that equality holds in (F.17), it is sufficient to exhibit one nonzero vector x_o that attains equality. Let k_o denote the index of the column that attains the maximum absolute column sum in the matrix product $DT^{-1}ATD^{-1}$. Let e_{k_o} denote the column basis vector of size $N \times 1$ with one at location k_o and zeros elsewhere. Select

$$x_o \triangleq TD^{-1}e_{k_o} \tag{F.18}$$

Then, it is straightforward to verify that

$$\|x_o\|_\rho \triangleq \|DT^{-1}x_o\|_1 \overset{(F.18)}{=} \|e_{k_o}\|_1 = 1 \tag{F.19}$$

and

$$\begin{aligned} \|Ax_o\|_\rho &\triangleq \|DT^{-1}Ax_o\|_1 \\ &= \|DT^{-1}A \cdot TD^{-1}DT^{-1} \cdot x_o\|_1 \\ &\overset{(F.18)}{=} \|DT^{-1}ATD^{-1}e_{k_o}\|_1 \\ &= \|A\|_\rho \end{aligned} \tag{F.20}$$

so that, for this particular vector, we have

$$\frac{\|Ax_o\|_\rho}{\|x_o\|_\rho} = \|A\|_\rho \tag{F.21}$$

as desired.

□

A Second Useful Matrix Norm

Let $x = \mathrm{col}\{x_1, x_2, \ldots, x_N\}$ now denote an $N \times 1$ *block* column vector whose individual entries are themselves vectors of size $M \times 1$ each. Following [32, 208, 230, 232], the block maximum norm of x is denoted by $\|x\|_{b,\infty}$ and is defined as

$$\|x\|_{b,\infty} \triangleq \max_{1 \le k \le N} \|x_k\| \tag{F.22}$$

That is, $\|x\|_{b,\infty}$ is equal to the largest Euclidean norm of its block components. This vector norm induces a block maximum matrix norm. Let $\mathcal{A}$ denote an arbitrary $N \times N$ block matrix with individual block entries of size $M \times M$ each. Then, the block maximum norm of $\mathcal{A}$ is defined as

$$\|\mathcal{A}\|_{b,\infty} \triangleq \max_{x \neq 0} \left(\frac{\|\mathcal{A}x\|_{b,\infty}}{\|x\|_{b,\infty}} \right) \tag{F.23}$$

The block maximum norm has several useful properties — see [208].

Lemma F.1 (Some useful properties of the block maximum norm)**.** The block maximum norm satisfies the following properties:

(a) Let $\mathcal{U} = \text{diag}\{U_1, U_2, \ldots, U_N\}$ denote an $N \times N$ block diagonal matrix with $M \times M$ unitary blocks $\{U_k\}$. Then, the block maximum norm is unitary-invariant, i.e., $\|\mathcal{U}x\|_{b,\infty} = \|x\|_{b,\infty}$ and $\|\mathcal{U}\mathcal{A}\mathcal{U}^*\|_{b,\infty} = \|\mathcal{A}\|_{b,\infty}$.

(b) Let $\mathcal{D} = \text{diag}\{D_1,\ D_2,\ \ldots, D_N\}$ denote an $N \times N$ block diagonal matrix with $M \times M$ Hermitian blocks $\{D_k\}$. Then, $\rho(\mathcal{D}) = \|\mathcal{D}\|_{b,\infty}$.

(c) Let A be an $N \times N$ matrix and define $\mathcal{A} = A \otimes I_M$ whose blocks are therefore of size $M \times M$ each. If A is left-stochastic (as defined further ahead by (F.46)), then $\|\mathcal{A}^{\mathsf{T}}\|_{b,\infty} = 1$.

(d) Consider a block diagonal matrix $\mathcal{D}$ as in part (b) and any left-stochastic matrices $\mathcal{A}_1$ and $\mathcal{A}_2$ constructed as in part (c). Then, it holds that

$$\rho\left(\mathcal{A}_2^{\mathsf{T}}\,\mathcal{D}\,\mathcal{A}_1^{\mathsf{T}}\right) \;\leq\; \rho(\mathcal{D}) \tag{F.24}$$

Jensen's Inequality

There are several variations and generalizations of Jensen's inequality. One useful form for our purposes is the following. Let $\{w_k\}$ denote a collection of N possibly complex-valued column vectors for $k = 1, 2, \ldots, N$. Let $\{\alpha_k\}$ denote a collection of nonnegative real coefficients that add up to one:

$$\sum_{k=1}^{N} \alpha_k \;= 1, \qquad 0 \leq \alpha_k \leq 1 \tag{F.25}$$

Jensen's inequality states that for any real-valued convex function $f(x) \in \mathbb{R}$, it holds [45, 126, 171]:

$$f\left(\sum_{k=1}^{N} \alpha_k w_k\right) \;\leq\; \sum_{k=1}^{N} \alpha_k f(w_k) \tag{F.26}$$

In particular, let

$$z \;\triangleq\; \sum_{k=1}^{N} \alpha_k w_k \tag{F.27}$$

If we select the function $f(z) = \|z\|^2$ in terms of the squared Euclidean norm of the vector z, then it follows from (F.26) that

$$\left\|\sum_{k=1}^{N} \alpha_k w_k\right\|^2 \leq \sum_{k=1}^{N} \alpha_k \|w_k\|^2 \tag{F.28}$$

There is also a useful stochastic version of Jensen's inequality. If $\boldsymbol{a} \in \mathbb{R}^M$ is a real-valued random variable, then it holds that

$$f(\mathbb{E}\,\boldsymbol{a}) \leq \mathbb{E}\,(f(\boldsymbol{a})) \quad (\text{when } f(x) \in \mathbb{R} \text{ is convex}) \tag{F.29}$$

$$f(\mathbb{E}\,\boldsymbol{a}) \geq \mathbb{E}\,(f(\boldsymbol{a})) \quad (\text{when } f(x) \in \mathbb{R} \text{ is concave}) \tag{F.30}$$

where it is assumed that $\boldsymbol{a}$ and $f(\boldsymbol{a})$ have bounded expectations. We remark that a function $f(x)$ is said to be concave if, and only if, $-f(x)$ is convex.

F.3 Perturbation Bounds on Eigenvalues

We state below two useful results that bound matrix eigenvalues.

Weyl's Theorem

The first result, known as Weyl's Theorem [113, 259], shows how the eigenvalues of a Hermitian matrix are disturbed through additive perturbations to the entries of the matrix. Thus, let $\{A', A, \Delta A\}$ denote arbitrary $N \times N$ Hermitian matrices with ordered eigenvalues $\{\lambda_m(A'), \lambda_m(A), \lambda_m(\Delta A)\}$, i.e.,

$$\lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_N(A) \tag{F.31}$$

and similarly for the eigenvalues of $\{A', \Delta A\}$, with the subscripts 1 and N representing the largest and smallest eigenvalues, respectively. Weyl's Theorem states that if A is perturbed to

$$A' = A + \Delta A \tag{F.32}$$

then the eigenvalues of the new matrix are bounded as follows:

$$\lambda_n(A) + \lambda_N(\Delta A) \leq \lambda_n(A') \leq \lambda_n(A) + \lambda_1(\Delta A) \tag{F.33}$$

for $1 \leq n \leq N$. In particular, it follows that the maximum eigenvalue is perturbed as follows:

$$\lambda_{\max}(A) + \lambda_{\min}(\Delta A) \leq \lambda_{\max}(A') \leq \lambda_{\max}(A) + \lambda_{\max}(\Delta A) \quad \text{(F.34)}$$

In the special case when $\Delta A \geq 0$, we conclude from (F.33) that $\lambda_n(A') \geq \lambda_n(A)$ for all $n = 1, 2, \ldots, N$.

Gershgorin's Theorem

The second result, known as Gershgorin's Theorem [48, 94, 101, 104, 113, 253, 263], specifies circular regions within which the eigenvalues of a matrix are located. Thus, consider an $N \times N$ matrix A with scalar entries $\{a_{\ell k}\}$. With each diagonal entry $a_{\ell\ell}$ we associate a disc in the complex plane centered at $a_{\ell\ell}$ and with

$$r_\ell \triangleq \sum_{k \neq \ell, k=1}^{N} |a_{\ell k}| \quad \text{(F.35)}$$

That is, r_ℓ is equal to the sum of the magnitudes of the non-diagonal entries on the same row as $a_{\ell\ell}$. We denote the disc by D_ℓ; it consists of all points that satisfy

$$D_\ell = \left\{ z \in \mathbb{C}^N \text{ such that } |z - a_{\ell\ell}| \leq r_\ell \right\} \quad \text{(F.36)}$$

The theorem states that the spectrum of A (i.e., the set of all its eigenvalues, denoted by $\lambda(A)$) is contained in the union of all N Gershgorin discs:

$$\lambda(A) \subset \bigcup_{\ell=1}^{N} D_\ell \quad \text{(F.37)}$$

A stronger statement of the Gershgorin theorem covers the situation in which some of the Gershgorin discs happen to be disjoint. Specifically, if the union of L of the discs is disjoint from the union of the remaining $N - L$ discs, then the theorem further asserts that L eigenvalues of A will lie in the first union of L discs and the remaining $N - L$ eigenvalues of A will lie in the second union of $N - L$ discs.

F.4 Lyapunov Equations

In this section, we introduce two particular Lyapunov equations and list some of their properties. We only list results that are used in the text. There are many other insightful results on Lyapunov equations. Interested readers may consult the works [132, 133, 148, 149] and the many references therein for additional information.

Discrete-Time Lyapunov Equations

Given $N \times N$ matrices X, A, and Q, where Q is Hermitian and nonnegative definite, we consider first discrete-time Lyapunov equations, also called Stein equations, of the following form:

$$X - A^* X A = Q \tag{F.38}$$

Let $\lambda_k(A)$ denote any of the eigenvalues of A. In the discrete-time case, a stable matrix A is one whose eigenvalues lie inside the unit disc (i.e., their magnitudes are strictly less than one).

Lemma F.2 (Discrete-time Lyapunov equation)**.** Consider the Lyapunov equation (F.38). The following facts hold:

(a) The solution X is unique if, and only if, $\lambda_k(A)\lambda_\ell^*(A) \neq 1$ for all $k, \ell = 1, 2, \ldots, N$. In this case, the unique solution X is Hermitian.

(b) When A is stable (i.e., all its eigenvalues are inside the unit disc), the solution X is unique, Hermitian, and nonnegative-definite. Moreover, it admits the series representation:

$$X = \sum_{n=0}^{\infty} (A^*)^n Q A^n \tag{F.39}$$

Proof. We call upon property 12 from Table F.1 for Kronecker products and apply the vec operation to both sides of (F.38) to get

$$(I - A^{\mathsf{T}} \otimes A^*)\mathrm{vec}(X) = \mathrm{vec}(Q) \tag{F.40}$$

This linear system of equations has a unique solution, $\mathrm{vec}(X)$, if, and only if, the coefficient matrix, $I - A^{\mathsf{T}} \otimes A^*$, is nonsingular. This latter condition

requires $\lambda_k(A)\lambda_\ell^*(A) \neq 1$ for all $k, \ell = 1, 2, \ldots, N$. When A is stable, all of its eigenvalues lie inside the unit disc and this uniqueness condition is automatically satisfied. If we conjugate both sides of (F.38) we find that X^* satisfies the same Lyapunov equation as X and, hence, by uniqueness, we must have $X = X^*$. Finally, let $F = A^\mathsf{T} \otimes A^*$. When A is stable, the matrix F is also stable since $\rho(F) = [\rho(A)]^2 < 1$. In this case, the matrix inverse $(I - F)^{-1}$ admits the series expansion

$$(I - F)^{-1} = I + F + F^2 + F^3 + \ldots \tag{F.41}$$

so that using (F.40) we have

$$\begin{aligned} \mathrm{vec}(X) &= (I - F)^{-1}\mathrm{vec}(Q) \\ &= \sum_{n=0}^{\infty} F^n \,\mathrm{vec}(Q) \\ &= \sum_{n=0}^{\infty} \left((A^\mathsf{T})^n \otimes (A^*)^n\right) \,\mathrm{vec}(Q) \\ &= \sum_{n=0}^{\infty} \mathrm{vec}\left((A^*)^n Q A^n\right) \end{aligned} \tag{F.42}$$

from which we deduce the series representation (F.39).

□

Continuous-Time Lyapunov Equations

A similar analysis applies to the following continuous-time Lyapunov equation (also called a Sylvester equation):

$$XA^* + AX + Q = 0 \tag{F.43}$$

In the continuous-time case, a stable matrix A is one whose eigenvalues lie in the open left-half plane (i.e., they have strictly negative real parts).

Lemma F.3 (Continuous-time Lyapunov equation)**.** Consider the Lyapunov equation (F.43). The following facts hold:

(a) The solution X is unique if, and only if, $\lambda_k(A) + \lambda_\ell^*(A) \neq 0$ for all $k, \ell = 1, 2, \ldots, N$. In this case, the unique solution X is Hermitian.

(b) When A is stable (i.e., all its eigenvalues lie in the open left-half plane), the solution X is unique, Hermitian, and nonnegative-definite.

Proof. We call again upon property 12 from Table F.1 for Kronecker products and apply the vec operation to both sides of (F.43) to get

$$[(A^* \otimes I) + (I \otimes A)]\,\mathrm{vec}(X) = -\mathrm{vec}(Q) \tag{F.44}$$

This linear system of equations has a unique solution, $\mathrm{vec}(X)$, if, and only if, the coefficient matrix, $(A^* \otimes I) + (I \otimes A)$, is nonsingular. This latter condition requires $\lambda_k(A) + \lambda_\ell^*(A) \neq 0$ for all $k, \ell = 1, 2, \ldots, N$. To see this, let $F = (A^* \otimes I) + (I \otimes A)$ and let us verify that the eigenvalues of F are given by all linear combinations $\lambda_k(A) + \lambda_\ell^*(A)$. Consider the eigenvalue-eigenvector pairs $Ax_k = \lambda_k(A)x_k$ and $A^*y_\ell = \lambda_\ell^*(A)y_\ell$. Then, using property 2 from Table F.1 for Kronecker products we get

$$\begin{aligned} F(y_\ell \otimes x_k) &= [(A^* \otimes I) + (I \otimes A)]\,(y_\ell \otimes x_k) \\ &= (A^*y_\ell \otimes x_k) + (y_\ell \otimes Ax_k) \\ &= \lambda_\ell^*(A)(y_\ell \otimes x_k) + \lambda_k(A)(y_\ell \otimes x_k) \\ &= (\lambda_k(A) + \lambda_\ell^*(A))(y_\ell \otimes x_k) \end{aligned} \tag{F.45}$$

so that the vector $(y_\ell \otimes x_k)$ is an eigenvector for F with eigenvalue $\lambda_k(A) + \lambda_\ell^*(A)$, as claimed. If we now conjugate both sides of (F.43) we find that X^* satisfies the same Lyapunov equation as X and, hence, by uniqueness, we must have $X = X^*$.

□

F.5 Stochastic Matrices

Consider $N \times N$ matrices A with nonnegative entries, $\{a_{\ell k} \geq 0\}$. The matrix $A = [a_{\ell k}]$ is said to be left-stochastic if it satisfies

$$A^{\mathsf{T}}\mathbb{1} = \mathbb{1} \qquad \text{(left-stochastic)} \tag{F.46}$$

where $\mathbb{1}$ denotes the column vector whose entries are all equal to one. It follows that the entries on each column of A add up to one. The matrix A is said to be doubly-stochastic if the entries on each of its columns and on each of its rows add up to one, i.e., if

$$A\mathbb{1} = \mathbb{1}, \quad A^{\mathsf{T}}\mathbb{1} = \mathbb{1} \qquad \text{(doubly-stochastic)} \tag{F.47}$$

Stochastic matrices arise frequently in the study of networks. The following statement lists two properties of stochastic matrices; additional properties can be found in [113, 208].

Lemma F.4 (Properties of stochastic matrices). Let A be an $N \times N$ left or doubly-stochastic matrix:

(a) The spectral radius of A is equal to one, $\rho(A) = 1$. It follows that all eigenvalues of A lie inside the unit disc, i.e., $|\lambda(A)| \leq 1$. The matrix A may have multiple eigenvalues with magnitude equal to one.

(b) If A is additionally a primitive matrix (cf. definition (6.1)), then A will have a single eigenvalue at one (i.e., the eigenvalue at one will have multiplicity one). All other eigenvalues of A will lie strictly inside the unit circle. Moreover, with proper sign scaling, all entries of the right-eigenvector of A corresponding to the single eigenvalue at one will be strictly positive, namely, if we let p denote this right-eigenvector with entries $\{p_k\}$ and normalize the entries to add up to one, then

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0, \quad k = 1, 2, \ldots, N \tag{F.48}$$

We refer to p as the *Perron eigenvector* of A. All other eigenvectors of A associated with the other eigenvalues will have at least one negative or complex entry.

F.6 Convergence of Inequality Recursions

The following are two convergence results involving inequality recursions; proofs appear in [190, pp. 45–50].

Lemma F.5 (Deterministic recursion). Let $u(i) \geq 0$ denote a scalar deterministic (i.e., non-random) sequence that satisfies the inequality recursion:

$$u(i+1) \;\leq\; [1 - a(i)]u(i) \;+\; b(i), \quad i \geq 0 \tag{F.49}$$

(a) When the scalar sequences $\{a(i), b(i)\}$ satisfy the four conditions:

$$0 \leq a(i) < 1, \qquad b(i) \geq 0, \qquad \sum_{i=0}^{\infty} a(i) = \infty, \qquad \lim_{i \to \infty} \frac{b(i)}{a(i)} = 0 \tag{F.50}$$

it holds that $\lim_{i\to\infty} u(i) = 0$.

(b) When the scalar sequences $\{a(i), b(i)\}$ are of the form

$$a(i) = \frac{c}{i+1}, \quad b(i) = \frac{d}{(i+1)^{p+1}}, \;\; c > 0, \;\; d > 0, \;\; p > 0 \tag{F.51}$$

it holds that, for large enough i, the sequence $u(i)$ converges to zero at one of the following rates depending on the value of c:

$$\begin{cases} u(i) \;\leq\; \left(\frac{d}{c-p}\right)\frac{1}{i^p} \;+\; o\left(1/i^p\right), & c > p \\ u(i) \;=\; O\left(\log i/i^p\right), & c = p \\ u(i) \;=\; O\left(1/i^c\right), & c < p \end{cases} \tag{F.52}$$

The fastest convergence rate occurs when $c > p$ and is in the order of $1/i^p$.

Note that part (b) of the above statement uses the big-O and little-o notation. The big-O notation is useful to compare the asymptotic growth rate of two sequences. Thus, writing $a(i) = O(b(i))$ means that $|a(i)| \leq c|b(i)|$ for some constant c and for all large enough $i > I_o$. For example, $a(i) = O(1/i)$ means that the samples of the sequence $a(i)$ decay asymptotically at a rate that is comparable to $1/i$. On the other hand, the little-o notation, $a(i) = o(b(i))$, means that, asymptotically, the sequence $a(i)$ decays faster than the sequence $b(i)$ so that $|a(i)|/|b(i)| \to 0$ as $i \to \infty$. In this case, the notation $a(i) = o(1/i)$ implies that the samples of $a(i)$ decay at a faster rate than $1/i$.

Lemma F.6 (Stochastic recursion)**.** Let $\boldsymbol{u}(i) \geq 0$ denote a scalar sequence of nonnegative random variables satisfying $\mathbb{E}\,\boldsymbol{u}(0) < \infty$ and the stochastic recursion:

$$\mathbb{E}\,[\boldsymbol{u}(i+1)|\,\boldsymbol{u}(0), \boldsymbol{u}(1), \ldots, \boldsymbol{u}(i)\,] \;\leq\; [1 - a(i)]\boldsymbol{u}(i) \;+\; b(i), \quad i \geq 0 \tag{F.53}$$

in terms of the conditional expectation on the left-hand side, and where the scalar and nonnegative deterministic sequences $\{a(i), b(i)\}$ satisfy the five conditions:

$$0 \leq a(i) < 1, \quad b(i) \geq 0, \quad \sum_{i=0}^{\infty} a(i) = \infty, \quad \sum_{i=0}^{\infty} b(i) < \infty, \quad \lim_{i\to\infty} \frac{b(i)}{a(i)} = 0 \tag{F.54}$$

Then, it holds that $\lim_{i\to\infty} \boldsymbol{u}(i) = 0$ almost surely, and $\lim_{i\to\infty} \mathbb{E}\,\boldsymbol{u}(i) = 0$.

G

Logistic Regression

Let $\boldsymbol{\gamma}_k$ denote a binary random variable whose value represents one of two possible classes, $+1$ or -1, depending on whether a feature vector $\boldsymbol{h}_k \in \mathbb{R}^M$ belongs to one class or the other. For example, the entries of $\boldsymbol{h}_k$ could represent measures of a person's weight and height, while the classes ± 1 could correspond to whether the feature $\boldsymbol{h}_k$ represents a male or a female individual. Logistic regression is a useful methodology for dealing with classification problems where one of the variables (the dependent variable) is binary and the second variable (the independent variable) is real-valued; this is in contrast to the more popular linear regression analysis where both variables are real-valued.

G.1 Logistic Function

When $\boldsymbol{\gamma}_k$ is a binary random variable, the relation between its realizations and the corresponding feature vectors $\{\boldsymbol{h}_k\}$ cannot be well represented by a linear regression model. A more suitable model is to represent the conditional probability of $\boldsymbol{\gamma}_k = 1$ given the feature vector $\boldsymbol{h}_k$ as a logistic function of the form [115, 233]:

$$P(\boldsymbol{\gamma}_k = +1 \mid \boldsymbol{h}_k) \;=\; \frac{1}{1 + e^{-\boldsymbol{h}_k^{\mathsf{T}} w^o}} \tag{G.1}$$

for some parameter vector $w^o \in \mathbb{R}^M$. Observe that regardless of the numerical values assumed by the entries of the feature vector $\boldsymbol{h}_k$, the logistic function always returns values between 0 and 1 (as befitting of a true probability measure) — see Figure G.1. Obviously, under the assumed binary model for $\boldsymbol{\gamma}_k$ and since the sum of the probabilities need to add up to one, it holds that

$$P(\boldsymbol{\gamma}_k = -1 \mid \boldsymbol{h}_k) \;=\; \frac{1}{1 + e^{\boldsymbol{h}_k^{\mathsf{T}} w^o}} \tag{G.2}$$

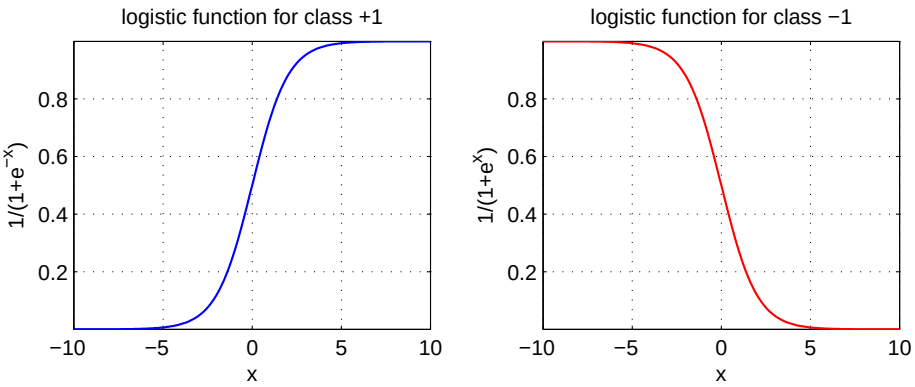

Figure G.1: Typical behavior of logistic functions for two classes. The figure shows plots of the functions $1/(1+e^{-x})$ (left) and $1/(1+e^{x})$ (right) assumed to correspond to classes $+1$ and -1, respectively.

G.2 Odds Function

We can group (G.1) and (G.2) into a single expression for the conditional probability density function (pdf) of $\boldsymbol{\gamma}_k$ and write:

$$p(\boldsymbol{\gamma}_k; w^o \mid \boldsymbol{h}_k) \;=\; \frac{1}{1 + e^{-\boldsymbol{\gamma}_k \boldsymbol{h}_k^{\mathsf{T}} w^o}} \tag{G.3}$$

with $\boldsymbol{\gamma}_k$ appearing in the exponent term on the right-hand side. This pdf is parameterized by w^o. In machine learning or pattern classification applications, one is usually served with a collection of training data $\{\boldsymbol{\gamma}_k, \boldsymbol{h}_k,\ k \geq 1\}$ and the objective is to use the data to estimate the

parameter w^o. Once w^o is recovered, its value can then be used to classify new feature vectors $\{\boldsymbol{h}_\ell\}$ into classes $+1$ or -1. This can be achieved, for example, by computing the odds of the new feature vector belonging to one class or the other. The odds function is defined as:

$$\text{odds} \triangleq \frac{P(\boldsymbol{\gamma}_\ell = +1 \mid \boldsymbol{h}_\ell)}{1 - P(\boldsymbol{\gamma}_\ell = +1 \mid \boldsymbol{h}_\ell)} \tag{G.4}$$

For example, in a scenario where the likelihood that type $+1$ occurs is 0.8 while the likelihood for type -1 is 0.2, we find that the odds of type $+1$ occurring are $4-$to-1, while the odds of type -1 occurring are $1-$to-4. If we compute the log of the odds ratio, we end up with the so-called logit function (or logistic transformation function):

$$\text{logit} \triangleq \ln\left(\frac{P(\boldsymbol{\gamma}_\ell = +1 \mid \boldsymbol{h}_\ell)}{1 - P(\boldsymbol{\gamma}_\ell = +1 \mid \boldsymbol{h}_\ell)}\right) \tag{G.5}$$

There are at least two advantages for the logit representation of the odds function. First, in this representation of the odds, types $+1$ and -1 will always have opposite odds (i.e., one value is the negative of the other). And, more importantly, if we use the assumed model (G.1), then the logit function ends up depending linearly on w^o. Specifically,

$$\text{logit} \;=\; \boldsymbol{h}_\ell^{\mathsf{T}} w^o \tag{G.6}$$

In this way, we can assign feature vectors $\{\boldsymbol{h}_\ell\}$ with nonnegative logit values to one class and feature vectors with negative logit values to another class — see Figure G.2.

G.3 Kullback-Leibler Divergence

To enable the above classification procedure, we still need to determine w^o. One way to estimate w^o is to fit into the training data $\{\boldsymbol{\gamma}_k, \boldsymbol{h}_k, k \geq 1\}$, a probability density function of the form:

$$p(\boldsymbol{\gamma}_k; w \mid \boldsymbol{h}_k) \;=\; \frac{1}{1 + e^{-\boldsymbol{\gamma}_k \boldsymbol{h}_k^{\mathsf{T}} w}} \tag{G.7}$$

for some unknown vector $w \in \mathbb{R}^M$ to be determined. This vector can be selected by minimizing the discrepancy between the above pdf and

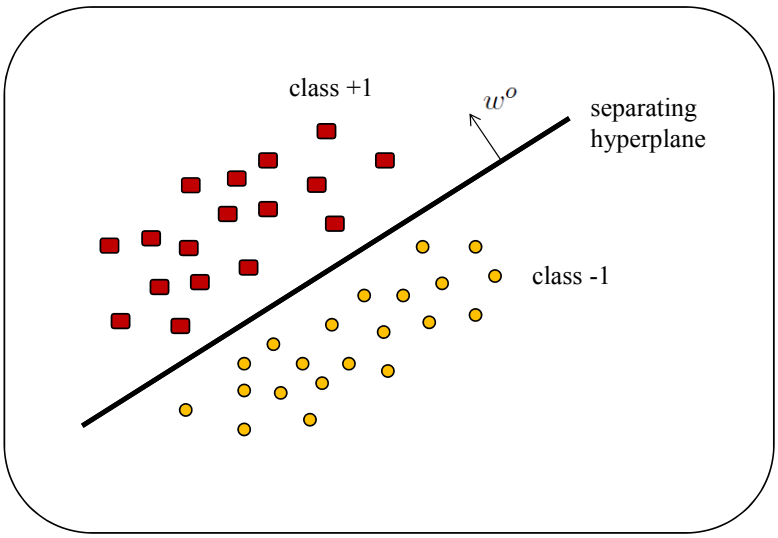

Figure G.2: Classification of feature vectors into two classes: data with non-negative logit values are assigned to one class and data with negative logit values are assigned to another class. The vector w^o defines the direction that is normal to the separating hyperplane.

the actual pdf corresponding to w^o in (G.3). A useful measure of discrepancy between two pdfs is the Kullback-Leibler (KL) divergence measure defined as [81]:

$$D_{\mathrm{KL}} \triangleq \mathbb{E}\left\{\ln\left(\frac{p(\boldsymbol{\gamma}_k; w^o \mid \boldsymbol{h}_k)}{p(\boldsymbol{\gamma}_k; w \mid \boldsymbol{h}_k)}\right)\right\} \tag{G.8}$$

where the expectation is over the distribution of the true pdf. The expression on the right-hand side involves the ratio of two pdfs: one using the true vector w^o and the other using the parameter w. Minimizing over w leads to the optimization problem

$$\min_{w} \; -\mathbb{E}\ln p(\boldsymbol{\gamma}_k; w \mid \boldsymbol{h}_k) \tag{G.9}$$

or, equivalently,

$$\min_{w} \; \mathbb{E}\left\{\ln\left[1 + e^{-\boldsymbol{\gamma}_k \boldsymbol{h}_k^{\mathsf{T}} w}\right]\right\} \tag{G.10}$$

which has the same form as the logistic regression cost function considered in the text — see, e.g., (2.9).

References

[1] R. Abdolee, B. Champagne, and A. H. Sayed. Diffusion LMS strategies for parameter estimation over fading wireless channels. In *Proc. IEEE ICC*, pages 1926–1930. Budapest, Hungary, June 2013.

[2] D. Acemoglu and A. Ozdaglar. Opinion dynamics and learning in social networks. *Dyn. Games Appl.*, 1(1):3–49, Mar. 2011.

[3] T. Adali, P. J. Schreier, and L. L. Scharf. Complex-valued signal processing: The proper way to deal with impropriety. *IEEE Trans. Signal Process.*, 59(11):5101–5125, Nov. 2011.

[4] A. Agarwal and J. Duchi. Distributed delayed stochastic optimization. In *Proc. Neural Information Processing Systems (NIPS)*, pages 873–881. Granada, Spain, Dec. 2011.

[5] L. V. Ahlfors. *Complex Analysis*. McGraw Hill, NY, 3rd edition, 1979.

[6] T. Y. Al-Naffouri and A. H. Sayed. Transient analysis of data-normalized adaptive filters. *IEEE Trans. Signal Process.*, 51(3):639–652, Mar. 2003.

[7] J. Alcock. *Animal Behavior: An Evolutionary Approach*. Sinauer Associates, 9th edition, 2009.

[8] P. Alriksson and A. Rantzer. Distributed Kalman filtering using weighted averaging. In *Proc. Int. Symp. Math. Thy Net. Sys (MTNS)*, pages 1–6. Kyoto, Japan, 2006.

[9] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay. Distributed least-mean-square estimation with partial diffusion. *IEEE Trans. Signal Process.*, 62(2):472–484, Jan. 2014.

[10] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed. Mean-square performance of a convex combination of two adaptive filters. *IEEE Trans. Signal Process.*, 54(3):1078–1090, March 2006.

[11] A. Auslender and M. Teboulle. Interior gradient and proximal methods for convex and conic optimization. *SIAM J. Optim.*, 16(3):697–725, 2006.

[12] A. Avitabile, R. A. Morse, and R. Boch. Swarming honey bees guided by pheromones. *Ann. Entomol. Soc. Am.*, 68:1079–1082, 1975.

[13] T. C. Aysal, M. J. Coates, and M. G. Rabbat. Distributed average consensus with dithered quantization. *IEEE Trans. Signal Process.*, 56(10):4905–4918, October 2008.

[14] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Trans. Signal Process.*, 57(7):2748–2761, July 2009.

[15] A.-L. Barabási. *Linked: How Everything Is Connected to Everything Else and What It Means.* Plume, NY, 2003.

[16] A.-L. Barabási and Z. N. Oltvai. Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101–113, 2004.

[17] R. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 25:21–30, Mar. 2007.

[18] S. Barbarossa and G. Scutari. Bio-inspired sensor network design. *IEEE Signal Processing Magazine*, 24(3):26–35, May 2007.

[19] A. Barrat, M. Barthélemy, and A. Vespignani. *Dynamical Processes on Complex Networks.* Cambridge University Press, 2008.

[20] M. F. Bear, B. W. Connors, and M. A. Paradiso. *Neuroscience: Exploring the Brain.* Lippincott, Williams & Wilkins, 3rd edition, 2006.

[21] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2:183–202, March 2009.

[22] M. Beekman, R. L. Fathke, and T. D. Seeley. How does an informed minority of scouts guide a honey bee swarm as it flies to its new home? *Animal Behavior*, 71:161–171, 2006.

[23] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *Proc. IEEE Int. Symp. Inf. Thy*, pages 1753–1757. Austin, TX, Jun. 2010.

[24] F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli. Order-optimal consensus through randomized path averaging. *IEEE Trans. Inf. Theory*, 56(10):5150–5167, Oct. 2010.

[25] H. Berg. Motile behavior of bacteria. *Physics Today*, 53(1):24–29, 2000.

[26] R. L. Berger. A necessary and sufficient condition for reaching a consensus using DeGroot's method. *J. Amer. Stat. Assoc.*, 76(374):415–418, Jun. 1981.

[27] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. SIAM, PA, 1994.

[28] A. Bertrand, M. Moonen, and A. H. Sayed. Diffusion bias-compensated RLS estimation over adaptive networks. *IEEE Trans. Signal Process.*, 59(11):5212–5224, Nov. 2011.

[29] D. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[30] D. P. Bertsekas. A new class of incremental gradient methods for least squares problems. *SIAM J. Optim.*, 7(4):913–926, 1997.

[31] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.

[32] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Singapore, 1st edition, 1997.

[33] D. P. Bertsekas and J. N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM J. Optim.*, 10(3):627–642, 2000.

[34] P. Bianchi, G. Fort, W. Hachem, and J. Jakubowicz. Convergence of a distributed parameter estimator for sensor networks with local averaging of the estimates. In *Proc. IEEE ICASSP*, pages 3764–3767. Prague, Czech, May 2011.

[35] L. Billera and P. Diaconis. A geometric interpretation of the metropolis-hastings algorithm. *Statist. Sci.*, 16:335–339, 2001.

[36] K. Binmore and J. Davies. *Calculus Concepts and Methods*. Cambridge University Press, 2007.

[37] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[38] D. Blatt, A. O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM J. Optim.*, 18:29–51, 2008.

[39] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proc. IEEE Conf. Dec. Control (CDC)*, pages 2996–3000. Seville, Spain, Dec. 2005.

[40] J. R. Blum. Multidimensional stochastic approximation methods. *Ann. Math. Stat.*, 25:737–744, 1954.

[41] B. Bollobas. *Modern Graph Theory.* Springer, 1998.

[42] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov chain on a graph. *SIAM Review*, 46(4):667–689, Dec. 2004.

[43] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Inf. Theory*, 52(6):2508–2530, Jun. 2006.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning,* NOW Publishers, 3(1):1–122, 2010.

[45] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[46] P. Braca, S. Marano, and V. Matta. Running consensus in wireless sensor networks. In *Proc. 11th International Conference on Information Fusion*, pages 1–6. Cologne, Germany, June 2008.

[47] D. H. Brandwood. A complex gradient operator and its apphcation in adaptive array theory. *IEE Proc.*, 130 parts F and H(1):11–16, 1983.

[48] R. A. Brualdi and S. Mellendorf. Regions in the complex plane containing the eigenvalues of a matrix. *Amer. Math. Monthly*, 101:975–985, 1994.

[49] G. Buzsaki. *Rythms of the Brain.* Oxford University Press, 2011.

[50] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems.* Princeton University Press, 2003.

[51] E. J. Candes, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.*, 14:877–905, 2007.

[52] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Distributed Kalman filtering using consensus strategies. *IEEE J. Sel. Areas Communications*, 26(4):622–633, Sep. 2008.

[53] F. Cattivelli and A. H. Sayed. Diffusion distributed Kalman filtering with adaptive weights. In *Proc. Asilomar Conf. Signals, Syst., Comput.*, pages 908–912. Pacific Grove, CA, Nov. 2009.

[54] F. Cattivelli and A. H. Sayed. Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Trans. Autom. Control*, 55(9):2069–2084, Sep. 2010.

[55] F. Cattivelli and A. H. Sayed. Analysis of spatial and incremental LMS processing for distributed estimation. *IEEE Trans. Signal Process.*, 59(4):1465–1480, April 2011.

[56] F. Cattivelli and A. H. Sayed. Modeling bird flight formations using diffusion adaptation. *IEEE Trans. Signal Process.*, 59(5):2038–2051, May 2011.

[57] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed. A diffusion RLS scheme for distributed estimation over adaptive networks. In *Proc. IEEE Work. Signal Process. Adv. Wireless Comm. (SPAWC)*, pages 1–5. Helsinki, Finland, June 2007.

[58] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed. Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Trans. Signal Process.*, 56(5):1865–1877, May 2008.

[59] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed. Diffusion strategies for distributed Kalman filtering: Formulation and performance analysis. In *Proc. Int. Work. Cogn. Inform. Process. (CIP)*, pages 36–41. Santorini, Greece, June 2008.

[60] F. S. Cattivelli and A. H. Sayed. Diffusion LMS algorithms with information exchange. In *Proc. Asilomar Conf. Signals, Syst., Comput.*, pages 251–255. Pacific Grove, CA, Nov. 2008.

[61] F. S. Cattivelli and A. H. Sayed. Diffusion mechanisms for fixed-point distributed Kalman smoothing. In *Proc. EUSIPCO*, pages 1–4. Lausanne, Switzerland, Aug. 2008.

[62] F. S. Cattivelli and A. H. Sayed. Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process.*, 58(3):1035–1048, Mar. 2010.

[63] R. Cavalcante, I. Yamada, and B. Mulgrew. An adaptive projected subgradient approach to learning in diffusion networks. *IEEE Trans. Signal Process.*, 57(7):2762–2774, July 2009.

[64] C. Chamley, A. Scaglione, and L. Li. Models for the diffusion of beliefs in social networks. *IEEE Signal Processing Magazine*, 30, May 2013.

[65] J. Chen and A. H. Sayed. Bio-inspired cooperative optimization with application to bacteria motility. In *Proc. IEEE ICASSP*, pages 5788–5791. Prague, Czech Republic, May 2011.

[66] J. Chen and A. H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process.*, 60(8):4289–4305, Aug. 2012.

[67] J. Chen and A. H. Sayed. Distributed pareto-optimal solutions via diffusion adaptation. In *Proc. IEEE Work. Stat. Signal Process. (SSP)*, pages 648–651. Ann Arbor, MI, Aug. 2012.

[68] J. Chen and A. H. Sayed. On the limiting behavior of distributed optimization strategies. In *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing*, pages 1535–1542. Monticello, IL, Oct. 2012.

[69] J. Chen and A. H. Sayed. Distributed Pareto optimization via diffusion strategies. *IEEE J. Sel. Topics Signal Process.*, 7(2):205–220, April 2013.

[70] J. Chen and A. H. Sayed. On the learning behavior of adaptive networks — Part I: Transient analysis. *Submitted for publication.* Also available as arXiv:1312.7581 [cs.MA], Dec. 2013.

[71] J. Chen and A. H. Sayed. On the learning behavior of adaptive networks — Part II: Performance analysis. *Submitted for publication.* Also available as arXiv:1312.7580 [cs.MA], Dec. 2013.

[72] J. Chen and A. H. Sayed. Controlling the limit point of left-stochastic policies over adaptive networks. *Submitted for publication*, 2014.

[73] Y. Chen, Y. Gu, and A. O. Hero. Sparse LMS for system identification. In *Proc. IEEE ICASSP*, pages 3125–3128. Taipei, Taiwan, May 2009.

[74] S. Chouvardas, G. Mileounis, N. Kalouptsidis, and S. Theodoridis. A greedy sparsity-promoting LMS for distributed adaptive learning in diffusion networks. In *Proc. IEEE ICASSP*, pages 5415–5419. Vancouver, BC, Canada, 2013.

[75] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis. A sparsity-promoting adaptive algorithm for distributed learning. *IEEE Trans. Signal Process.*, 60(10):5412–5425, Oct. 2012.

[76] S. Chouvardas, K. Slavakis, and S. Theodoridis. Adaptive robust distributed learning in diffusion sensor networks. *IEEE Trans. Signal Process.*, 59(10):4692–4707, Oct. 2011.

[77] N. Christakis and J. Fowler. *Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives.* Little, Brown and Company, 2009.

[78] F. Iutzeler P. Ciblat and W. Hachem. Analysis of sum-weight-like algorithms for averaging in wireless sensor networks. *IEEE Trans. Signal Process.*, 6(11):2802–2814, Jun. 2013.

[79] I. D. Couzin. Collective cognition in animal groups. *Trends in Cognitive Sciences*, 13:36–43, Jan. 2009.

[80] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218:1–11, 2002.

[81] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, NJ, 1991.

[82] D. M. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs: Theory and Applications*. Wiley, NY, 1998.

[83] A. Das and M. Mesbahi. Distributed linear parameter estimation in sensor networks based on laplacian dynamics consensus algorithm. In *Proc. IEEE SECON*, volume 2, pages 440–449. Reston, VA, Sep. 2006.

[84] M. H. DeGroot. Reaching a consensus. *J. Amer. Stat. Assoc.*, 69(345):118–121, 1974.

[85] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction. In *Proc. International Conference on Machine Learning (ICML)*, pages 713–720. Bellevue, WA, Jun. 2011.

[86] P. Di Lorenzo and A. H. Sayed. Sparse distributed learning based on diffusion adaptation. *IEEE Trans. Signal Process.*, 61(6):1419–1433, March 2013.

[87] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, Nov. 2010.

[88] P. M. Djuric and Y. Wang. Distributed Bayesian learning in multiagent systems. *IEEE Signal Processing Magazine*, 29(2):65–76, Mar. 2012.

[89] R. M. Dudley. *Real Analysis and Probability*. Cambridge Univ. Press, 2nd edition, 2003.

[90] L. A. Dugatkin. *Principles of Animal Behavior*. W. W. Norton & Company, 2nd edition, 2009.

[91] R. Durret. *Probability Theory and Examples*. Duxbury Press, 2nd edition, 1996.

[92] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[93] C. H. Edwards Jr. *Advanced Calculus of Several Variables*. Dover Publications, NY, 1995.

[94] D. G. Feingold and R. S. Varga. Block diagonally dominant matrices and generalizations of the gerschgorin circle theorem. *Pacific J. Math.*, 12:1241–1250, 1962.

[95] J. Fernandez-Bes, J. Arenas-Garcia, and A. H. Sayed. Adjustment of combination weights over adaptive diffusion networks. In *Proc. IEEE ICASSP*, pages 1–5. Florence, Italy, May 2014.

[96] A. Feuer and E. Weinstein. Convergence analysis of LMS filters with uncorrelated Gaussian data. *IEEE Trans. Acoust., Speech, Signal Process.*, 33(1):222–230, Feb. 1985.

[97] J. B. Foley and F. M. Boland. A note on the convergence analysis of LMS adaptive filters with Gaussian data. *IEEE Trans. Acoust., Speech, Signal Process.*, 36(7):1087–1089, Jul. 1988.

[98] J. Fowler and N. Christakis. Cooperative behavior cascades in human social networks. *Proc. Nat. Acad. Sciences*, 107(12):5334–5338, 2010.

[99] F. R. Gantmacher. *The Theory of Matrices*. Chelsea Publishing Company, NY, 1959.

[100] W. A. Gardner. Learning characterisitcs of stochastic-gradient-descent algorithms: A general study, analysis, and critique. *Signal Process.*, 6(2):113–133, Apr. 1984.

[101] S. Gerschgorin. Über die abgrenzung der eigenwerte einer matrix. *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk*, 7:749–754, 1931.

[102] O. N. Gharehshiran, V. Krishnamurthy, and G. Yin. Distributed energy-aware diffusion least mean squares: Game-theoretic learning. *IEEE J. Sel. Top. Signal Process.*, 7(5):1–16, Oct. 2013.

[103] B. Golub and M. O. Jackson. Naive learning in social networks and the wisdom of crowds. *American Economic Journal: Microeconomics*, 2:112–149, 2010.

[104] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, 3rd edition, 1996.

[105] W. D. Hamilton. Geometry for the selfish herd. *Journal of Theoretical Biology*, 31:295–311, 1971.

[106] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, Apr. 1970.

[107] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, NJ, 2002.

[108] S. Haykin. *Cognitive Dynamic Systems*. Cambridge University Press, 2012.

[109] E. S. Helou and A. R. De Pierro. Incremental subgradients for constrained convex optimization: A unified framework and new methods. *SIAM J. Optim.*, 20:1547–1572, 2009.

[110] F. H. Heppner. Avian flight formations. *Bird-Banding*, 45(2):160–169, 1974.

[111] A. Hjorungnes. *Complex-Valued Matrix Derivatives*. Cambridge University Press, 2011.

[112] O. Hlinka, O. Sluciak, F. Hlawatsch, and P. M. Djuric. Likelihood consensus and its application to distributed particle filtering. *IEEE Trans. Signal Process.*, 60(8):4334–4349, August 2012.

[113] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2003.

[114] L. Horowitz and K. Senne. Performance advantage of complex LMS for controlling narrow-band adaptive arrays. *IEEE Trans. Acoust., Speech, Signal Process.*, 29(3):722–736, Jun. 1981.

[115] D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. Wiley, NJ, 2nd edition, 2000.

[116] K. Hreutz-Delgado. The complex gradient operator and the cr-calculus. Available online as manuscript arXiv:0906.4835 [math.OC], June 2009.

[117] J. Hu, L. Xie, and C. Zhang. Diffusion Kalman filtering based on covariance intersection. *IEEE Trans. Signal Process.*, 60(2):891–902, Feb. 2012.

[118] S. Hubbard, P. Babak, S. T. Sigurdsson, and K. G. Magnusson. A model of the formation of fish schools and migrations of fish. *Ecological Modeling*, 174:359–374, June 2004.

[119] D. Hummel. Aerodynamic aspects of formation flight in birds. *J. Theor. Biol.*, 104(3):321–347, 1983.

[120] M. D. Intriligator. *Mathematical Optimization and Economic Theory*. Prentice-Hall, NJ, 1971.

[121] M. Jackson. *Social and Economic Networks*. Princeton University Press, Princeton, NJ, 2008.

[122] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control*, 48(6):988–1001, Jun. 2003.

[123] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi. Non-bayesian social learning. *Game. Econ. Behav.*, 76(1):210–225, Sep. 2012.

[124] D. Jakovetic, J. Xavier, and J. M. F. Moura. Cooperative convex optimization in netowrked systems: Augmented lagranian algorithms with directed gossip communication. *IEEE Trans. Signal Process.*, 59(8):3889–3902, Aug. 2011.

[125] S. Janson, M. Middendorf, and M. Beekman. Honeybee swarms: How do scouts guide a swarm of uninformed bees? *Animal Behavior*, 70:349–358, 2005.

[126] J. L. W. V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.

[127] C. Jiang, Y. Chen, and K. J. Ray Liu. Distributed adaptive networks: A graphical evolutionary game-theoretic view. *IEEE Trans. Signal Process.*, 61(22):5675–5688, Nov. 2013.

[128] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *Proc. IEEE Conf. Dec. Control (CDC)*, pages 4185–4190. Cancun, Mexico, December 2008.

[129] B. Johansson, M. Rabi, and M. Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM J. Optim.*, 20:1157–1170, 2009.

[130] S. Jones, R. C. III, and W. Reed. Analysis of error-gradient adaptive linear estimators for a class of stationary dependent processes. *IEEE Trans. Inf. Theory*, 28(2):318–329, Mar. 1982.

[131] B. H. Junker and F. Schreiber. *Analysis of Biological Networks*. Wiley, NJ, 2008.

[132] T. Kailath. *Linear Systems*. Prentice Hall, NJ, 1980.

[133] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, NJ, 2000.

[134] S. Kar and J. M. F. Moura. Sensor networks with random links: Topology design for distributed consensus. *IEEE Trans. Signal Process.*, 56(7):3315–3326, July 2008.

[135] S. Kar and J. M. F. Moura. Distributed consensus algorithms in sensor networks: Link failures and channel noise. *IEEE Trans. Signal Process.*, 57(1):355–369, Jan. 2009.

[136] S. Kar and J. M. F. Moura. Distributed consensus algorithms in sensor netowrks: Quantized data and random link failures. *IEEE Trans. Signal Process.*, 58(3):1383–1400, Mar. 2010.

[137] S. Kar and J. M. F. Moura. Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs. *IEEE J. Sel. Topics Signal Process.*, 5(4):674–690, Aug. 2011.

[138] S. Kar, J. M. F. Moura, and K. Ramanan. Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication. *IEEE Trans. Inf. Theory*, 58(6):3575–3605, Jun. 2012.

[139] R. M. Karp. Reducibility among combinational problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–104. Plenum Press, NY, 1972.

[140] D. Kempe, A Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. Annual IEEE Symp. Found. Computer Sci.*, pages 482–491. Cambridge, MA, Oct. 2003.

[141] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers. Steady state analysis of diffusion LMS adaptive networks with noisy links. *IEEE Trans. Signal Process.*, 60(2):974–979, Feb. 2012.

[142] U. A. Khan and J. M. F. Moura. Distributing the Kalman filter for large-scale systems. *IEEE Trans. Signal Process.*, 56(10):4919–4935, Oct. 2008.

[143] W. Kocay and D. L. Kreher. *Graphs, Algorithms and Optimization*. Chapman & Hall/CRC Press, Boca Raton, 2005.

[144] A. N. Kolmogorov and S. V. Fomin. *Introductory Real Analysis*. Dover Publications, 1975.

[145] R. H. Koning, H. Neudecker, and T. Wansbeek. Block kronecker products and the vecb operator. *Linear Algebra Appl.*, 149:165–184, Apr. 1991.

[146] F. Kopos, editor. *Biological Networks*. World Scientific Publishing Company, 2007.

[147] Y. Kopsinis, K. Slavakis, and S. Theodoridis. Online sparse system identification and signal reconstruction using projections onto weighted balls. *IEEE Trans. Signal Process.*, 59(3):936–952, Mar. 2010.

[148] P. Lancaster and L. Rodman. *Algebraic Riccati Equations*. Oxford University Press, NY, 1995.

[149] P. Lancaster and M. Tismenetsky. *Theory of Matrices with Applications*. Academic Press, NY, 2nd edition, 1985.

[150] R. Larson and B. H. Edwards. *Calculus*. Brooks Cole, 9th edition, 2009.

[151] J-W. Lee, S-E. Kim, W.-J. Song, and A. H. Sayed. Spatio-temporal diffusion mechanisms for adaptation over networks. In *Proc. EUSIPCO*, pages 1040–1044. Barcelona, Spain, Aug.–Sep. 2011.

[152] J.-W. Lee, S.-E. Kim, W.-J. Song, and A. H. Sayed. Spatio-temporal diffusion strategies for estimation and detection over networks. *IEEE Trans. Signal Process.*, 60(8):4017–4034, August 2012.

[153] S. Lee and A. Nedic. Distributed random projection algorithm for convex optimization. *IEEE J. Sel. Topics Signal Process.*, 7(2):221–229, Apr. 2013.

[154] T. G. Lewis. *Network Science: Theory and Applications*. Wiley, NJ, 2009.

[155] J. Li and A. H. Sayed. Modeling bee swarming behavior through diffusion adaptation with asymmetric information sharing. *EURASIP Journal on Advances in Signal Processing*, 2012. 2012:18, doi:10.1186/1687-6180-2012-18.

[156] L. Li, C. G. Lopes, J. Chambers, and A. H. Sayed. Distributed estimation over an adaptive incremental network based on the affine projection algorithm. *IEEE Trans. Signal Process.*, 58(1):151–164, Jan. 2010.

[157] Y. Liu, C. Li, and Z. Zhang. Diffusion sparse least-mean squares over networks. *IEEE Trans. Signal Process.*, 60(8):4480–4485, Aug. 2012.

[158] C. Lopes and A. H. Sayed. Diffusion adaptive networks with changing topologies. In *Proc. IEEE ICASSP*, pages 3285–3288. Las Vegas, April 2008.

[159] C. G. Lopes and A. H. Sayed. Distributed processing over adaptive networks. In *Proc. Adaptive Sensor Array Processing Workshop*, pages 1–5. MIT Lincoln Laboratory, MA, June 2006.

[160] C. G. Lopes and A. H. Sayed. Diffusion least-mean-squares over adaptive networks. In *Proc. IEEE ICASSP*, volume 3, pages 917–920. Honolulu, Hawaii, April 2007.

[161] C. G. Lopes and A. H. Sayed. Incremental adaptive strategies over distributed networks. *IEEE Trans. Signal Process.*, 55(8):4064–4077, Aug. 2007.

[162] C. G. Lopes and A. H. Sayed. Steady-state performance of adaptive diffusion least-mean squares. In *Proc. IEEE Work. Stat. Signal Process. (SSP)*, pages 136–140. Madison, WI, Aug. 2007.

[163] C. G. Lopes and A. H. Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Trans. Signal Process.*, 56(7):3122–3136, July 2008.

[164] O. Macchi. *Adaptive Processing: The Least Mean Squares Approach with Applications in Transmission*. Wiley, NY, 1995.

[165] G. Mateos, Gonzalo, I. D. Schizas, and G. B. Giannakis. Distributed recursive least-squares for consensus-based in-network adaptive estimation. *IEEE Trans. Signal Process.*, 57(11):4583–4599, Nov. 2009.

[166] G. Mateos, I. D. Schizas, and G. B. Giannakis. Performance analysis of the consensus-based distributed LMS algorithm. *EURASIP J. Adv. Signal Process.*, pages 1–19, 2009. 10.1155/2009/981030, Article ID 981030.

[167] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[168] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, PA, 2001.

[169] S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge Univ. Press, 2nd edition, 2009.

[170] H. Milinski and R. Heller. Influence of a predator on the optimal foraging behavior of sticklebacks. *Nature*, 275:642–644, 1978.

[171] D. S. Mitrinović. *Elementary Inequalities*. P. Noordhoff Ltd., Netherlands, 1964.

[172] A. Nedic and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. Optim.*, 12(1):109–138, 2001.

[173] A. Nedic and A. Olshevsky. Distributed optimization over time-varying directed graphs. *Submitted for publication*. Also available as arXiv:1303.2289 [math.OC], Mar. 2014.

[174] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1):48–61, Jan. 2009.

[175] A. Nedic and A. Ozdaglar. Cooperative distributed multi-agent optimization. In Y. Eldar and D. Palomar, editors, *Convex Optimization in Signal Processing and Communications*, pages 340–386. Cambridge University Press, 2010.

[176] Y. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[177] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[178] M. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[179] R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *Proc. IEEE Conf. Dec. Control (CDC)*, pages 8179–8184. Seville, Spain, Dec. 2005.

[180] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control*, 51:401–420, Mar. 2006.

[181] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *Proc. 46th IEEE Conf. Decision Control*, pages 5492–5498. New Orleans, LA, Dec. 2007.

[182] R. Olfati-Saber. Kalman-consensus filter: Optimality, stability, and performance. In *Proc. IEEE Conf. Dec. Control (CDC)*, pages 7036–7042. Shangai, China, 2009.

[183] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, Jan. 2007.

[184] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control*, 49:1520–1533, Sep. 2004.

[185] R. Olfati-Saber and J. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proc. IEEE Conf. Dec. Control (CDC)*, pages 6698–6703. Seville, Spain, Dec. 2005.

[186] A. Papoulis and S. U. Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, NY, 4th edition, 2002.

[187] B. L. Partridge. The structure and function of fish schools. *Scientific American*, 246(6):114–123, June 1982.

[188] K. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(6):52–67, 2002.

[189] S. U. Pillai, T. Suel, and S. Cha. The Perron–Frobenius theorem: Some of its applications. *IEEE Signal Process. Mag.*, 22(2):62–75, Mar. 2005.

[190] B. Poljak. *Introduction to Optimization*. Optimization Software, NY, 1987.

[191] B. T. Poljak and Y. Z. Tsypkin. Pseudogradient adaptation and training algorithms. *Autom. Remote Control*, 12:83–94, 1973.

[192] J. B. Predd, S. B. Kulkarni, and H. V. Poor. Distributed learning in wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):56–69, Jul. 2006.

[193] J. B. Predd, S. R. Kulkarni, and H. V. Poor. A collaborative training algorithm for distributed learning. *IEEE Trans. Inf. Theory*, 55(4):1856–1871, April 2009.

[194] M. G. Rabbat and R. D. Nowak. Quantized incremental algorithms for distributed optimization. *IEEE J. Sel. Areas Commun.*, 23(4):798–808, 2005.

[195] M. G. Rabbat, R. D. Nowak, and J. A. Bucklew. Generalized consensus computation in networked systems with erasure links. In *Proc. IEEE Work. Signal Process. Adv. Wireless Comm. (SPAWC)*, pages 1088–1092. New York, NY, June 2005.

[196] S. S. Ram, A. Nedic, and V. V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *J. Optim. Theory Appl.*, 147(3):516–545, 2010.

[197] R. Remmert. *Theory of Complex Functions*. Springer-Verlag, 1991.

[198] W. Ren and R. W. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control*, 50:655–661, May 2005.

[199] C. W. Reynolds. Flocks, herds, and schools: A distributed behavior model. *ACM Proc. Comput. Graphs Interactive Tech.*, pages 25–34, 1987.

[200] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22:400–407, 1951.

[201] O. L. Rortveit, J. H. Husoy, and A. H. Sayed. Diffusion LMS with communications constraints. In *Proc. Asilomar Conf. Signals, Syst., Comput.*, pages 1645–1649. Pacific Grove, CA, Nov. 2010.

[202] H. L. Royden. *Real Analysis*. Prentice-Hall, NJ, 3rd edition, 1988.

[203] V. Saligrama, M. Alanyali, and O. Savas. Distributed detection in sensor networks with packet losses and finite capacity links. *IEEE Trans. Signal Process.*, 54:4118–4132, 2006.

[204] S. Sardellitti, M. Giona, and S. Barbarossa. Fast distributed average consensus algorithms based on advection-diffusion processes. *IEEE Trans. Signal Process.*, 58(2):826–842, Feb. 2010.

[205] A. H. Sayed. *Fundamentals of Adaptive Filtering*. Wiley, NJ, 2003.

[206] A. H. Sayed. *Adaptive Filters*. Wiley, NJ, 2008.

[207] A. H. Sayed. Adaptive networks. *Proceedings of the IEEE*, 102(4):460–497, April 2014.

[208] A. H. Sayed. Diffusion adaptation over networks. In R. Chellapa and S. Theodoridis, editors, *E-Reference Signal Processing*, vol. 3, pages 323–454. Academic Press, 2014. Also available as arXiv:1205.4220v1 [cs.MA], May 2012.

[209] A. H. Sayed and F. Cattivelli. Distributed adaptive learning mechanisms. In S. Haykin and K. J. Ray Liu, editors, *Handbook on Array Processing and Sensor Networks*, pages 695–722. Wiley, NJ, 2009.

[210] A. H. Sayed and C. Lopes. Distributed recursive least-squares strategies over adaptive networks. In *Proc. Asilomar Conf. Signals, Syst. Comput.*, pages 233–237. Pacific Grove, CA, Oct.-Nov. 2006.

[211] A. H. Sayed and C. G. Lopes. Adaptive processing over distributed networks. *IEICE Trans. Fund. of Electron., Commun. and Comput. Sci.*, E90-A(8):1504–1510, 2007.

[212] A. H. Sayed and F. A. Sayed. Diffusion adaptation over networks of particles subject to brownian fluctuations. In *Proc. Asilomar Conf. Signals, Syst., Comput.*, pages 685–690. Pacific Grove, CA, Nov. 2011.

[213] A. H. Sayed, S-Y. Tu, and J. Chen. Online learning and adaptation over networks: More information is not necessarily better. In *Proc. Information Theory and Applications Workshop (ITA)*, pages 1–8. San Diego, Feb. 2013.

[214] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic. Diffusion strategies for adaptation and learning over networks. *IEEE Signal Processing Magazine*, 30(3):155–171, May 2013.

[215] D. S. Scherber and H. C. Papadopoulos. Locally constructed algorithms for distributed computations in ad-hoc networks. In *Proc. Information Processing in Sensor Networks (IPSN)*, pages 11–19. Berkeley, CA, April 2004.

[216] I. D. Schizas, G. Mateos, and G. B. Giannakis. Distributed LMS for consensus-based in-network adaptive processing. *IEEE Trans. Signal Process.*, 57(6):2365–2382, June 2009.

[217] L. Schmetterer. Stochastic approximation. *Proc. Berkeley Symp. Math. Statist. Probab.*, pages 587–609, 1961.

[218] P. J. Schreier and L. L. Scharf. *Statistical Signal Processing of Complex-Valued Data*. Cambridge University Press, 2010.

[219] T. D. Seeley, R. A. Morse, and P. K. Visscher. The natural history of the flight of honey bee swarms. *Psyche.*, 86:103–114, 1979.

[220] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer, 2nd edition, 2007.

[221] D. Shah. Gossip algorithms. *Found. Trends Netw.*, 3:1–125, 2009.

[222] K. Slavakis, Y. Kopsinis, and S. Theodoridis. Adaptive algorithm for sparse system identification using projections onto weighted ℓ_1 balls. In *Proc. IEEE ICASSP*, pages 3742–3745. Dallas, TX, Mar. 2010.

[223] S. Sonnenburg, V. Franc, E. Yom-Tov, and M. Sebag. Pascal large scale learning challenge. Online site at http://largescale.ml.tu-berlin.de, 2008.

[224] A. Speranzon, C. Fischione, and K. H. Johansson. Distributed and collaborative estimation over wireless sensor networks. In *Proc. IEEE Conf. Dec. Control (CDC)*, pages 1025–1030. San Dieog, USA, Dec. 2006.

[225] O. Sporns. *Networks of the Brain*. MIT Press, 2010.

[226] K. Srivastava and A. Nedic. Distributed asynchronous constrained stochastic optimization. *IEEE J. Sel. Topics. Signal Process.*, 5(4):772–790, Aug. 2011.

[227] S. S. Stankovic, M. S. Stankovic, and D. S. Stipanovic. Decentralized parameter estimation by consensus based stochastic approximation. *IEEE Trans. Autom. Control*, 56(3):531–543, Mar. 2011.

[228] D. J. T. Sumpter and S. C. Pratt. A modeling framework for understanding social insect foraging. *Behavioral Ecology and Sociobiology*, 53:131–144, 2003.

[229] J. Surowiecki. *The Wisdom of the Crowds*. Doubleday, 2004.

[230] N. Takahashi and I. Yamada. Parallel algorithms for variational inequalities over the cartesian product of the intersections of the fixed point sets of nonexpansive mappings. *J. Approx. Theory*, 153(2):139–160, Aug. 2008.

[231] N. Takahashi and I. Yamada. Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks. In *Proc. IEEE ICASSP*, pages 3518–3521. Dallas, TX, Mar. 2010.

[232] N. Takahashi, I. Yamada, and A. H. Sayed. Diffusion least-mean-squares with adaptive combiners: Formulation and performance analysis. *IEEE Trans. Signal Process.*, 58(9):4795–4810, Sep. 2010.

[233] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 4th edition, 2008.

[234] S. Theodoridis, K. Slavakis, and I. Yamada. Adaptive learning in a world of projections: A unifying framework for linear and nonlinear classification and regression tasks. *IEEE Signal Processing Magazine*, 28(1):97–123, Jan. 2011.

[235] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Statistical Society: Series B*, 58:267–288, 1996.

[236] Z. Towfic, J. Chen, and A. H. Sayed. On the generalization ability of distributed online learners. In *Proc. IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. Santander, Spain, Sep. 2012.

[237] Z. Towfic and A. H. Sayed. Adaptive stochastic convex optimization over networks. In *Proc. 51th Annual Allerton Conference on Communication, Control, and Computing*, pages 1–6. Monticello, IL, Oct. 2013.

[238] Z. Towfic and A. H. Sayed. Adaptive penalty-based distributed stochastic convex optimization. *IEEE Trans. Signal Process.*, 62(15):3924–3938, Aug. 2014.

[239] Z. J. Towfic, J. Chen, and A. H. Sayed. Collaborative learning of mixture models using diffusion adaptation. In *Proc. IEEE Workshop Mach. Learn. Signal Process. (MLSP)*, pages 1–6. Beijing, China, Sep. 2011.

[240] K. I. Tsianos, S. Lawlor, and M. G. Rabbat. Push-sum distributed dual averaging for convex optimization. In *Proc. IEEE Conf. Dec. Control (CDC)*, pages 5453–5458. Hawaii, Dec. 2012.

[241] J. Tsitsiklis and M. Athans. Convergence and asymptotic agreement in distributed decision problems. *IEEE Trans. Autom. Control*, 29(1):42–50, Jan. 1984.

[242] J. Tsitsiklis, D. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. Autom. Control*, 31(9):803–812, Sep. 1986.

[243] Y. Z. Tsypkin. *Adaptation and Learning in Automatic Systems*. Academic Press, NY, 1971.

[244] S-Y. Tu and A. H. Sayed. Adaptive networks with noisy links. In *Proc. IEEE Globecom*, pages 1–5. Houston, TX, December 2011.

[245] S-Y. Tu and A. H. Sayed. Cooperative prey herding based on diffusion adaptation. In *Proc. IEEE ICASSP*, pages 3752–3755. Prague, Czech Republic, May 2011.

[246] S.-Y. Tu and A. H. Sayed. Mobile adaptive networks. *IEEE J. Sel. Topics. Signal Process.*, 5(4):649–664, Aug. 2011.

[247] S-Y. Tu and A. H. Sayed. On the effects of topology and node distribution on learning over complex adaptive networks. In *Proc. Asilomar Conf. Signals, Syst., Comput.*, pages 1166–1171. Pacific Grove, CA, Nov. 2011.

[248] S.-Y. Tu and A. H. Sayed. Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Trans. Signal Process.*, 60(12):6217–6234, Dec. 2012.

[249] S-Y. Tu and A. H. Sayed. Effective information flow over mobile adaptive networks. In *Proc. Int. Work. Cogn. Inform. Process. (CIP)*, pages 1–6. Parador de Baiona, Spain, May 2012.

[250] S-Y. Tu and A. H. Sayed. On the influence of informed agents on learning and adaptation over networks. *IEEE Trans. Signal Process.*, 61(6):1339–1356, Mar. 2013.

[251] A. van den Bos. Complex gradient and hessian. *IEE Proc. Vis. Image Signal Process.*, 141(6):380–382, 1994.

[252] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, NY, 2000.

[253] R. S. Varga. *Gersgorin and His Circles*. Springer-Verlag, Berlin, 2004.

[254] T. Vicsek, A. Czirook, E. Ben-Jacob, O. Cohen, and I. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75:1226–1229, Aug. 1995.

[255] R. von Mises and H. Pollaczek-Geiringer. Praktische verfahren der gleichungs-auflösung. *Z. Agnew. Math. Mech.*, 9:152–164, 1929.

[256] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[257] C. Waters and B. Bassler. Quorum sensing: cell-to-cell communication in bacteria. *Annual Review of Cell and Developmental Biology*, 21:319–346, 2005.

[258] G. B. Wetherhill. *Sequential Methods in Statistics*. Methuen, London, 1966.

[259] H. Weyl. Über beschrankte quadratiche formen, deren differenz vollsteig ist. *Rend. Circ. Mat. Palermo*, 27:373–392, 1909.

[260] B. Widrow and M. E. Hoff, Jr. Adaptive switching circuits. *IRE WESCON Conv. Rec.*, Pt. 4:96–104, 1960.

[261] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson Jr. Stationary and nonstationary learning characterisitcs of the LMS adaptive filter. *Proceedings of the IEEE*, 64(8):1151–1162, Aug. 1976.

[262] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice Hall, NJ, 1985.

[263] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, 1965.

[264] W. Wirtinger. Zur formalen theorie der funktionen von mehr komplexen veränderlichen. *Math. Ann.*, 97:357–375, 1927.

[265] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Syst. Control Lett.*, 53(1):65–78, Sep. 2004.

[266] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proc. IPSN, 2005*, pages 63–70. Los Angeles, CA, April 2005.

[267] L. Xiao, S. Boyd, and S. Lall. A space-time diffusion scheme peer-to-peer least-squares-estimation. In *Proc. Information Processing in Sensor Networks (IPSN)*, pages 168–176. Nashville, TN, April 2006.

[268] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi. Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties. *IEEE Trans. Knowledge and Data Engineering*, 25(11):2483–2493, Nov. 2013.

[269] N. R. Yousef and A. H. Sayed. A unified approach to the steady-state and tracking analysis of adaptive filters. *IEEE Trans. Signal Process.*, 49(2):314–324, February 2001.

[270] C.-K. Yu and A. H. Sayed. A strategy for adjusting combination weights over adaptive networks. In *Proc. IEEE ICASSP*, pages 4579–4583. Vancouver, Canada, May 2013.

[271] C.-K. Yu, M. van der Schaar, and A. H. Sayed. Reputation design for adaptive networks with selfish agents. In *Proc. IEEE Work. Signal Process. Adv. Wireless Comm. (SPAWC)*, pages 160–164. Darmstadt, Germany, June 2013.

[272] L. A. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Trans. Autom. Control*, 8:59–60, Jan. 1963.

[273] X. Zhao and A. H. Sayed. Clustering via diffusion adaptation over networks. In *Proc. Int. Work. Cogn. Inform. Process. (CIP)*, pages 1–6. Parador de Baiona, Spain, May 2012.

[274] X. Zhao and A. H. Sayed. Combination weights for diffusion strategies with imperfect information exchange. In *Proc. IEEE ICC*, pages 398–402. Ottawa, Canada, June 2012.

[275] X. Zhao and A. H. Sayed. Learning over social networks via diffusion adaptation. In *Proc. Asilomar Conf. Signals, Syst., Comput.*, pages 709–713. Pacific Grove, CA, Nov. 2012.

[276] X. Zhao and A. H. Sayed. Performance limits for distributed estimation over LMS adaptive networks. *IEEE Trans. Signal Process.*, 60(10):5107–5124, Oct. 2012.

[277] X. Zhao and A. H. Sayed. Asynchronous adaptation and learning over networks — Part I: Modeling and stability analsysis. *Submitted for publication.* Also available as arXiv:1312.5434 [cs.SY], Dec. 2013.

[278] X. Zhao and A. H. Sayed. Asynchronous adaptation and learning over networks – Part II: Performance analysis. *Submitted for publication.* Also available as arXiv:1312.5438 [cs.SY], Dec. 2013.

[279] X. Zhao and A. H. Sayed. Attaining optimal batch performance via distributed processing over networks. In *Proc. IEEE ICASSP*, pages 5214–5218. Vancouver, Canada, May 2013.

[280] X. Zhao, S.-Y. Tu, and A. H. Sayed. Diffusion adaptation over networks under imperfect information exchange and non-stationary data. *IEEE Trans. Signal Process.*, 60(7):3460–3475, July 2012.

Errata

Version: June 2, 2015.

A. H. Sayed, *Adaptation, Learning, and Optimization over Networks*, Foundations and Trends in Machine Learning, volume 7, issue 4-5, NOW Publishers, Boston-Delft, 518pp, 2014.

Remark. *The typos are already marked in red in the manuscript's pdf file.*

1. Expressions (2.55) and (2.56): the running index for the summations and the arguments of $\mu(i)$ and $\mu^2(i)$ should be i' instead of i.
2. Expression (4.148): ρw^o is missing on the right-hand side.
3. Expression (4.149): $-\rho^2 w^o (w^o)^{\mathsf{T}}$ is missing on the right-hand side.
4. Expression (5.102): replace rightmost ∞ by 0 (similar to (3.91)).
5. Figure 7.4: replace $\{\boldsymbol{w}_{4,i-1}, \boldsymbol{w}_{7,i-1}, \boldsymbol{w}_{\ell,i-1}\}$ by $\{\boldsymbol{\psi}_{4,i}, \boldsymbol{\psi}_{7,i}, \boldsymbol{\psi}_{\ell,i}\}$.
6. Expressions (9.307) and (10.116): replace second "$\leq$" by "$=$".
7. Expression (11.20): symbol $\mathbb{E}$ missing before first $\boldsymbol{s}^e_{k,i}$ on first line.
8. Rephrase sentence right after (11.111) as: "Selecting the origin of time at some large time and iterating from there"
9. Expression (11.130), second line, second symbol $\otimes$ should be $\otimes_b$.
10. Expression (11.131), first two lines, second symbol $\otimes$ should be $\otimes_b$.
11. Fourth line below expression (14.3): replace "add up to one" by "add up to zero".
12. Expression (13.47): $(2 + N_I)$ should be $\left(2 + N_I^{-1}\right)$.
13. Expression (14.38): replace $\boldsymbol{w}$ by $\widetilde{\boldsymbol{w}}$ on the right-hand side.
14. Expression (14.39): first equality, replace $\boldsymbol{s}_{e,i}$ by $\boldsymbol{s}^e_{\ell,i}$.
15. Expression (14.47): replace ζ_k by ζ_ℓ.
16. Sentence after (E.32), replace the word "Hermitian" by "symmetric."