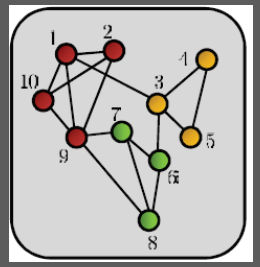
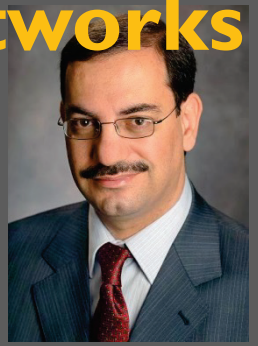


INFERENCE OVER NETWORKS

LECTURE #16: Evolution of Multi-Agent Networks



Professor Ali H. Sayed
UCLA Electrical Engineering

Course EE210B
Spring Quarter 2015

Proc. IEEE, vol. 102, no. 4, pp. 460-497, April 2014.
Foundations and Trends in Machine Learning, vol. 7, no. 4-5, pp. 311-801, July 2014.

Reference

Chapter 8 (Evolution of Multi-Agent Networks, pp. 470-496):

A. H. Sayed, ``Adaptation, learning, and optimization over networks," ***Foundations and Trends in Machine Learning***, vol. 7, issue 4-5, pp. 311-801, NOW Publishers, 2014.

Setting

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 lectures in order to emphasize in each lecture 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.

Setting

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.

Network Error Dynamics (MSE Networks)

ICASSP 2015

Proc. IEEE, vol. 102, no. 4, pp. 460-497, April 2014.
Foundations and Trends in Machine Learning, vol. 7, no. 4-5, pp. 311-801, July 2014.

State Recursion

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].

Example #8.1

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:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^{o} + \boldsymbol{v}_k(i), \qquad k = 1, 2, \ldots, N \tag{8.1}$$

Example #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 $\sigma^2_{v,k}$ 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

$$w^o_k \;=\; w^o \;=\; R^{-1}_{u,k} r_{du,k} \tag{8.3}$$

Example #8.1

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)).

Example #8.1

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	$\begin{cases} \boldsymbol{\psi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^*[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \end{cases}$
CTA diffusion	$\begin{cases} \boldsymbol{\psi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^*[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k,i-1}] \end{cases}$
ATC diffusion	$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^*[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \\ \boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{\psi}_{\ell,i} \end{cases}$

Example #8.1

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 $\{a_{o,\ell k}, a_{1,\ell k}, a_{2,\ell k}\}$:

$$\left\{\begin{array}{rcl} \boldsymbol{\phi}_{k,i-1} & = & \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{\psi}_{k,i} & = & \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\ \boldsymbol{\phi}_{\ell,i-1} \ + \ \mu_k \boldsymbol{u}_{k,i}^{*}\left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\phi}_{k,i-1}\right] \\ \boldsymbol{w}_{k,i} & = & \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\boldsymbol{\psi}_{\ell,i} \end{array}\right. \qquad (8.5)$$

Example #8.1

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:

$$A_o \triangleq [a_{o,\ell k}], \quad A_1 \triangleq [a_{1,\ell k}], \quad A_2 \triangleq [a_{2,\ell k}] \tag{8.6}$$

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 $P = A_1 A_o A_2$:

Example #8.1

$$\text{non-cooperative:} \quad A_1 = A_o = A_2 = I_N \longrightarrow P = I_N \tag{8.7}$$

$$\text{consensus:} \quad A_o = A, \;\; A_1 = I_N = A_2 \longrightarrow P = A \tag{8.8}$$

$$\text{CTA diffusion:} \quad A_1 = A, \;\; A_2 = I_N = A_o \longrightarrow P = A \tag{8.9}$$

$$\text{ATC diffusion:} \quad A_2 = A, \;\; A_1 = I_N = A_o \longrightarrow P = A \tag{8.10}$$

We associate with each agent k the following three errors:

$$\widetilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i} \tag{8.11}$$

$$\widetilde{\boldsymbol{\psi}}_{k,i} \triangleq w^o - \boldsymbol{\psi}_{k,i} \tag{8.12}$$

$$\widetilde{\boldsymbol{\phi}}_{k,i-1} \triangleq w^o - \boldsymbol{\phi}_{k,i-1} \tag{8.13}$$

Example #8.1

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

$$\begin{cases} \widetilde{\boldsymbol{\phi}}_{k,i-1} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\ \widetilde{\boldsymbol{w}}_{\ell,i-1} \\ \widetilde{\boldsymbol{\psi}}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\ \widetilde{\boldsymbol{\phi}}_{\ell,i-1} - \mu_k \boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i} \widetilde{\boldsymbol{\phi}}_{k,i-1} - \mu_k \boldsymbol{u}_{k,i}^* \boldsymbol{v}_k(i) \\ \widetilde{\boldsymbol{w}}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\ \widetilde{\boldsymbol{\psi}}_{\ell,i} \end{cases} \tag{8.14}$$

Example #8.1

In a manner similar to (3.126), the gradient noise process at each agent k is given by

$$\boldsymbol{s}_{k,i}(\boldsymbol{\phi}_{k,i-1}) \;=\; \left(R_{u,k} - \boldsymbol{u}_{k,i}^{*}\boldsymbol{u}_{k,i}\right)\widetilde{\boldsymbol{\phi}}_{k,i-1} \;-\; \boldsymbol{u}_{k,i}^{*}\boldsymbol{v}_{k}(i) \tag{8.15}$$

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):

$$\widetilde{\boldsymbol{w}}_{i} \;\triangleq\; \begin{bmatrix} \widetilde{\boldsymbol{w}}_{1,i} \\ \widetilde{\boldsymbol{w}}_{2,i} \\ \vdots \\ \widetilde{\boldsymbol{w}}_{N,i} \end{bmatrix}, \quad \widetilde{\boldsymbol{\psi}}_{i} \;\triangleq\; \begin{bmatrix} \widetilde{\boldsymbol{\psi}}_{1,i} \\ \widetilde{\boldsymbol{\psi}}_{2,i} \\ \vdots \\ \widetilde{\boldsymbol{\psi}}_{N,i} \end{bmatrix}, \quad \widetilde{\boldsymbol{\phi}}_{i-1} \;\triangleq\; \begin{bmatrix} \widetilde{\boldsymbol{\phi}}_{1,i-1} \\ \widetilde{\boldsymbol{\phi}}_{2,i-1} \\ \vdots \\ \widetilde{\boldsymbol{\phi}}_{N,i-1} \end{bmatrix} \tag{8.16}$$

Example #8.1

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:

$$\boldsymbol{z}_i \triangleq \begin{bmatrix} \boldsymbol{u}_{1,i}^* \boldsymbol{v}_1(i) \\ \boldsymbol{u}_{2,i}^* \boldsymbol{v}_2(i) \\ \vdots \\ \boldsymbol{u}_{N,i}^* \boldsymbol{v}_N(i) \end{bmatrix}, \qquad \boldsymbol{s}_i \triangleq \begin{bmatrix} \boldsymbol{s}_{1,i}(\boldsymbol{\phi}_{1,i-1}) \\ \boldsymbol{s}_{2,i}(\boldsymbol{\phi}_{2,i-1}) \\ \vdots \\ \boldsymbol{s}_{N,i}(\boldsymbol{\phi}_{N,i-1}) \end{bmatrix} \tag{8.17}$$

Example #8.1

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 $\{a_{o,\ell k} I_M, a_{1,\ell k} I_M, \ a_{2,\ell k} I_M\}$.

Example #8.1

We also introduce the following $N \times N$ *block* diagonal matrices, whose individual entries are of size $M \times M$ each:

$$\mathcal{M} \triangleq \text{diag}\{ \mu_1 I_M,\ \mu_2 I_M,\ \ldots,\ \mu_N I_M \} \tag{8.19}$$

$$\mathcal{R}_i \triangleq \text{diag}\{ \boldsymbol{u}_{1,i}^* \boldsymbol{u}_{1,i},\ \boldsymbol{u}_{2,i}^* \boldsymbol{u}_{2,i},\ \ldots,\ \boldsymbol{u}_{N,i}^* \boldsymbol{u}_{N,i} \} \tag{8.20}$$

From (8.14), we can easily conclude that the block network variables (8.16) satisfy the relations:

$$\left\{ \begin{array}{rcl} \widetilde{\boldsymbol{\phi}}_{i-1} & = & \mathcal{A}_1^{\mathsf{T}} \widetilde{\boldsymbol{w}}_{i-1} \\ \widetilde{\boldsymbol{\psi}}_i & = & \left[\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\mathcal{R}_i \right] \widetilde{\boldsymbol{\phi}}_{i-1} - \mathcal{M}\boldsymbol{z}_i \\ \widetilde{\boldsymbol{w}}_i & = & \mathcal{A}_2^{\mathsf{T}} \widetilde{\boldsymbol{\psi}}_i \end{array} \right. \tag{8.21}$$

Example #8.1

so that the network weight error vector, $\widetilde{\boldsymbol{w}}_i$, ends up evolving according to the following *stochastic* state-space recursion:

$$\widetilde{\boldsymbol{w}}_i \;=\; \mathcal{A}_2^{\mathsf{T}}\left(\mathcal{A}_o^{\mathsf{T}} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right)\mathcal{A}_1^{\mathsf{T}}\widetilde{\boldsymbol{w}}_{i-1} \;-\; \mathcal{A}_2^{\mathsf{T}}\mathcal{M}\boldsymbol{z}_i, \;\; i \geq 0 \;\; \text{(distributed)} \qquad (8.22)$$

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:

$$\widetilde{\boldsymbol{w}}_i \;=\; \left(I_{MN} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right)\widetilde{\boldsymbol{w}}_{i-1} \;-\; \mathcal{M}\boldsymbol{z}_i, \;\; i \geq 0 \qquad \text{(non-cooperative)} \qquad (8.23)$$

where the matrices $\{\mathcal{A}_o, \mathcal{A}_1, \mathcal{A}_2\}$ do not appear any longer, and with a block diagonal coefficient matrix $(I_{MN} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)$.

Example #8.1

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

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]:

Example #8.2

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i = \mathcal{B}\,(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}) \quad \text{(distributed)} \tag{8.28}$$

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i = (I_{MN} - \mathcal{M}\mathcal{R})\,(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}) \quad \text{(non-cooperative)} \tag{8.29}$$

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)):

Example #8.2

$$\mathcal{B}_{\text{ncop}} = I_{MN} - \mathcal{M}\mathcal{R} \tag{8.30}$$

$$\mathcal{B}_{\text{cons}} = \mathcal{A}^{\mathsf{T}} - \mathcal{M}\mathcal{R} \tag{8.31}$$

$$\mathcal{B}_{\text{atc}} = \mathcal{A}^{\mathsf{T}}\left(I_{MN} - \mathcal{M}\mathcal{R}\right) \tag{8.32}$$

$$\mathcal{B}_{\text{cta}} = \left(I_{MN} - \mathcal{M}\mathcal{R}\right)\mathcal{A}^{\mathsf{T}} \tag{8.33}$$

where $\mathcal{A} = A \otimes I_M$.

■

Example #8.3

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:

$$\mathcal{M} = \mu I_N \otimes I_M, \qquad \mathcal{R} = I_N \otimes R_u \tag{8.34}$$

Example #8.3

so that expressions (8.30)–(8.33) reduce to

$$\begin{cases} \mathcal{B}_{\text{ncop}} &= I_N \otimes (I_M - \mu R_u) \\ \mathcal{B}_{\text{cons}} &= A^{\mathsf{T}} \otimes I_M - \mu(I_M \otimes R_u) \\ \mathcal{B}_{\text{atc}} &= A^{\mathsf{T}} \otimes (I_M - \mu R_u) \\ \mathcal{B}_{\text{cta}} &= A^{\mathsf{T}} \otimes (I_M - \mu R_u) \end{cases} \tag{8.35}$$

Example #8.3

For example, starting from (8.32) we have

$$\begin{aligned}
\mathcal{B}_{\text{atc}} &= \mathcal{A}^{\mathsf{T}}\,(I_{MN} - \mathcal{M}\mathcal{R}) \\
&= (A \otimes I_M)^{\mathsf{T}}\,[(I_N \otimes I_M) \;-\; (\mu I_N \otimes I_M)(I_N \otimes R_u)] \\
&= (A \otimes I_M)^{\mathsf{T}}\,[(I_N \otimes I_M) \;-\; \mu(I_N \otimes I_M)(I_N \otimes R_u)] \\
&= (A \otimes I_M)^{\mathsf{T}}\,[(I_N \otimes I_M) \;-\; \mu(I_N \otimes R_u)] \\
&= (A^{\mathsf{T}} \otimes I_M)\,[I_N \otimes (I_M - \mu R_u)] \\
&= A^{\mathsf{T}} \otimes (I_M - \mu R_u) \qquad (8.36)
\end{aligned}$$

Example #8.3

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 $m = 1, 2, \ldots, M$:

$$\lambda(\mathcal{B}_{\rm diff}) = \lambda_k(A)\,[1 - \mu\lambda_m(R_u)] \tag{8.37}$$

$$\lambda(\mathcal{B}_{\rm cons}) = \lambda_k(A) - \mu\lambda_m(R_u) \tag{8.38}$$

$$\lambda(\mathcal{B}_{\rm ncop}) = 1 - \mu\lambda_m(R_u) \tag{8.39}$$

Example #8.3

The expressions for $\lambda(\mathcal{B}_{\text{diff}})$ and $\lambda(\mathcal{B}_{\text{ncop}})$ follow directly from the properties of Kronecker products — see Table F.1. The expression for $\lambda(\mathcal{B}_{\text{cons}})$ 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

Example #8.3

$$
\begin{aligned}
\mathcal{B}_{\text{cons}}(x_k \otimes y_m) &= \left[A^{\mathsf{T}} \otimes I_M \;-\; \mu(I_M \otimes R_u)\right](x_k \otimes y_m) \\
&= (A^{\mathsf{T}} x_k \otimes y_m) \;-\; \mu(x_k \otimes R_u y_m) \\
&= (\lambda_k(A) x_k \otimes y_m) \;-\; \mu(x_k \otimes \lambda_m(R_u) y_m) \\
&= \lambda_k(A)(x_k \otimes y_m) \;-\; \mu\lambda_m(R_u)(x_k \otimes y_m) \\
&= (\lambda_k(A) - \mu\lambda_m(R_u))(x_k \otimes y_m) \qquad (8.40)
\end{aligned}
$$

so that $x_k \otimes y_m$ is an eigenvector for $\mathcal{B}_{\text{cons}}$ with eigenvalue $\lambda_k(A) - \mu\lambda_m(R_u)$, as claimed.

■

Example #8.4

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

$$\rho(\mathcal{B}_{\text{diff}}) = \rho(\mathcal{B}_{\text{ncop}}) \tag{8.41}$$

Example #8.4

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:

$$\text{individual agents stable} \implies \text{diffusion networks stable} \tag{8.42}$$

Example #8.4

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):

$$\text{individual agents stable} \not\Rightarrow \text{consensus networks stable} \tag{8.43}$$

Example #8.4

$$\begin{cases} \lambda(\mathcal{B}_{\text{ncop}}) &= 1 - 2\mu\lambda_m(R_u) \\ \lambda(\mathcal{B}_{\text{cons}}) &= \lambda_k(A) - 2\mu\lambda_m(R_u) \\ \lambda(\mathcal{B}_{\text{diff}}) &= \lambda_k(A)\,[1 - 2\mu\lambda_m(R_u)] \end{cases}$$

Potential instability in consensus networks:

$$\begin{cases} \text{individual agents stable} \implies \text{diffusion networks stable} \\ \text{individual agents stable} \not\Rightarrow \text{consensus networks stable} \end{cases}$$

Example #8.4

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.

■

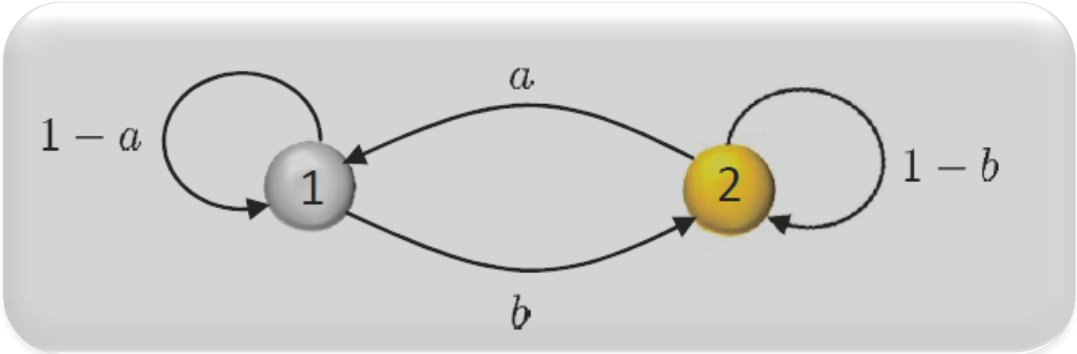
Example #A (2-Agent Network)

$$R_{u,1} = \sigma_{u,1}^2 \, I_M, \quad R_{u,2} = \sigma_{u,2}^2 \, I_M$$

$$0 < \mu_1 \sigma_{u,1}^2 \leq \mu_2 \sigma_{u,2}^2 < 2$$ (individual LMS agents: **mean stable**)

$$A = \begin{bmatrix} 1-a & b \\ a & 1-b \end{bmatrix}$$

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i \;=\; \mathcal{B}\,(\mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1})$$



Example #A (2-Agent Network)

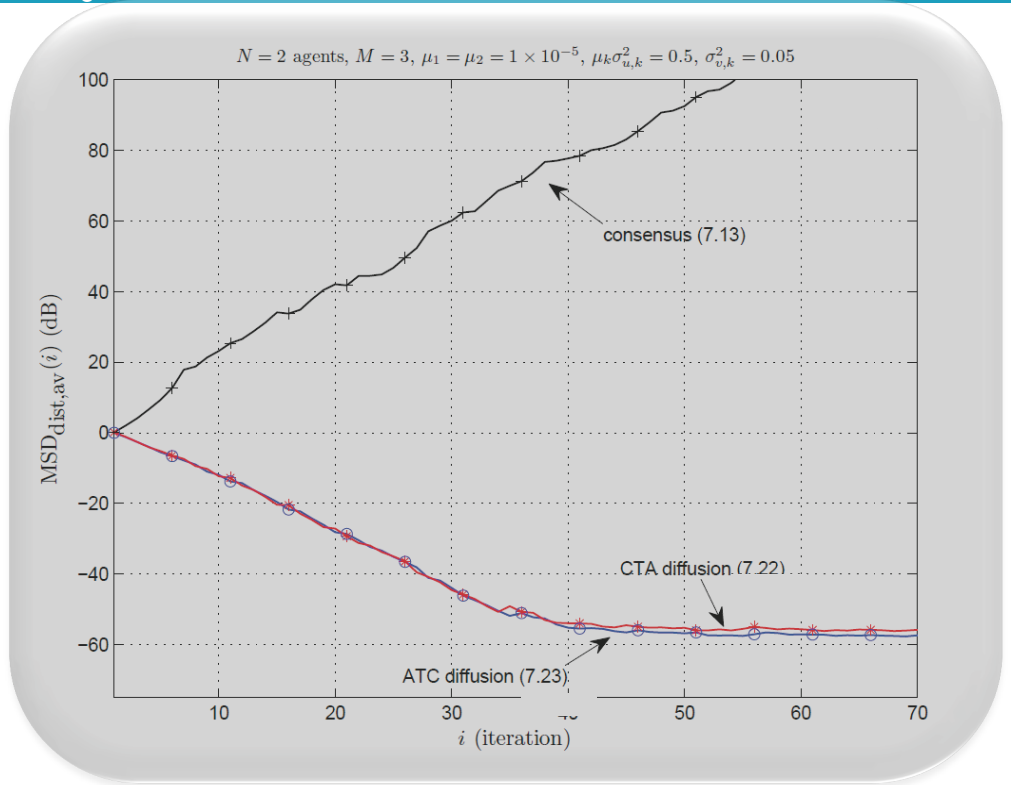
$$\mathcal{B}_{\text{atc}} = \begin{bmatrix} (1-a)\,(1-\mu_1\sigma_{u,1}^2) & a\,(1-\mu_2\sigma_{u,2}^2) \\ b\,(1-\mu_1\sigma_{u,1}^2) & (1-b)\,(1-\mu_2\sigma_{u,2}^2) \end{bmatrix} \otimes I_M$$

$$\mathcal{B}_{\text{cons}} = \begin{bmatrix} (1-a)-\mu_1\sigma_{u,1}^2 & a \\ b & (1-b)-\mu_2\sigma_{u,2}^2 \end{bmatrix} \otimes I_M$$

If $a+b \geq 2-\mu_1\sigma_{u,1}^2 > 0$, then consensus becomes unstable.

In contrast, diffusion is always stable.

Example #A (2-Agent Network)



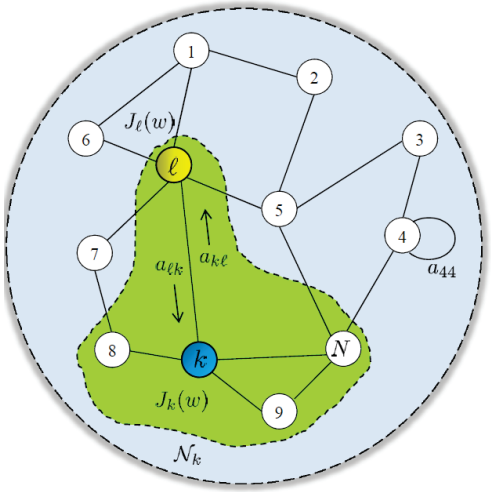
Network Limit Point

ICASSP 2015

Proc. IEEE, vol. 102, no. 4, pp. 460-497, April 2014.
Foundations and Trends in Machine Learning, vol. 7, no. 4-5, pp. 311-801, July 2014.

Aggregate Cost

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



$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} J_k(w) \tag{8.44}$$

Unique Minimizer

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 $J^{\text{glob}}(w)$ by w^o; it is the unique solution to the algebraic equation:

$$\nabla_w J^{\text{glob}}(w^o) = 0 \iff \sum_{k=1}^{N} \nabla_w J_k(w^o) = 0 \qquad (8.45)$$

Complications

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.

Hessian Matrices

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.

Different Minimizers

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

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	$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{cases}$

Unified Description

In a manner similar to (8.5), we can again describe these strategies by means of a single unifying description as follows:

$$\begin{cases} \boldsymbol{\phi}_{k,i-1} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\ \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{\psi}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{o,\ell k}\ \boldsymbol{\phi}_{\ell,i-1} - \mu_k\ \widehat{\nabla_{w^*} J_k}\left(\boldsymbol{\phi}_{k,i-1}\right) \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\ \boldsymbol{\psi}_{\ell,i} \end{cases} \tag{8.46}$$

Unified Description

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 underlying *connected* network topology so that none of their rows are identically zero.

Unified Description

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:

Perron Eigenvector

(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:

$$q \triangleq \text{diag}\{\mu_1, \mu_2, \ldots, \mu_N\} A_2 p \tag{8.50}$$

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

$$q_k > 0, \qquad k = 1, 2, \ldots, N \tag{8.51}$$

Weighted Aggregate Cost

We also represent the step-sizes as scaled multiples of the same factor $\mu_{\max}$, namely,

$$\mu_k \triangleq \tau_k \, \mu_{\max}, \quad k = 1, 2, \ldots, N \tag{8.52}$$

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

$$J^{\text{glob},\star}(w) \triangleq \sum_{k=1}^{N} q_k J_k(w) \tag{8.53}$$

Weighted Aggregate Cost

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

$$\begin{aligned}
\nabla_w^2 J^{\text{glob},\star}(w) &= \sum_{k=1}^{N} q_k \nabla_w^2 J_k(w) \\
&\geq q_{\min} \cdot \left(\sum_{k=1}^{N} \nabla_w^2 J_k(w) \right) \\
&\overset{(6.13)}{\geq} q_{\min} \frac{\nu_d}{h} I_{hM} > 0 \qquad (8.54)
\end{aligned}$$

Weighted Aggregate Cost

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:

$$\nabla_w J^{\text{glob},\star}(w^\star) = 0 \iff \sum_{k=1}^{N} q_k \nabla_w J_k(w^\star) = 0 \qquad (8.55)$$

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:

Weighted Aggregate Cost

$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} J_k(w) \longrightarrow \nabla_w J^{\text{glob}}(w^o) = 0 \iff \sum_{k=1}^{N} \nabla_w J_k(w^o) = 0$$

$$J^{\text{glob},\star}(w) \triangleq \sum_{k=1}^{N} q_k J_k(w) \longrightarrow \nabla_w J^{\text{glob},\star}(w^\star) = 0 \iff \sum_{k=1}^{N} q_k \nabla_w J_k(w^\star) = 0$$

(is also **strongly convex**)

Weighted Aggregate Cost

(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.

Limit Point

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:

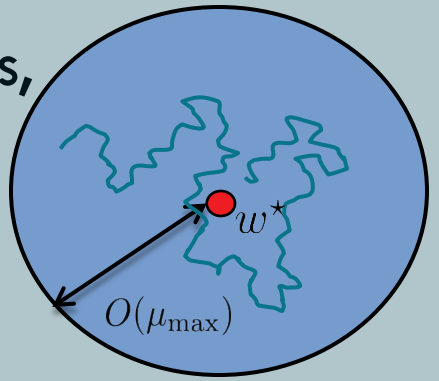
$$\limsup_{i \to \infty} \mathbb{E} \, \|\widetilde{\boldsymbol{w}}_{k,i}\|^2 = O(\mu_{\max}) \tag{8.57}$$

Limit Point

$$\widetilde{\boldsymbol{w}}_{k,i} \triangleq w^\star - \boldsymbol{w}_{k,i}, \quad k = 1, 2, \ldots, N$$

Theorem 9.1: For sufficiently small step-sizes, it holds that

$$\limsup_{i \to \infty} \mathbb{E} \, \|\widetilde{\boldsymbol{w}}_{k,i}\|^2 = O(\mu_{\max})$$



Limit Point

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

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:

$$\bar{J}^{\text{glob},\star}(w) \triangleq \sum_{k=1}^{N} \bar{q}_k J_k(w) \tag{8.59}$$

Example #8.5

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 $J(w)$.

Example #8.5

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

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

$$q_k \stackrel{\Delta}{=} \mu_k p_k > 0, \quad k = 1, 2, \ldots, N \tag{8.62}$$

Example #8.6

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.

■

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.

Pareto Solution

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:

Pareto Solution

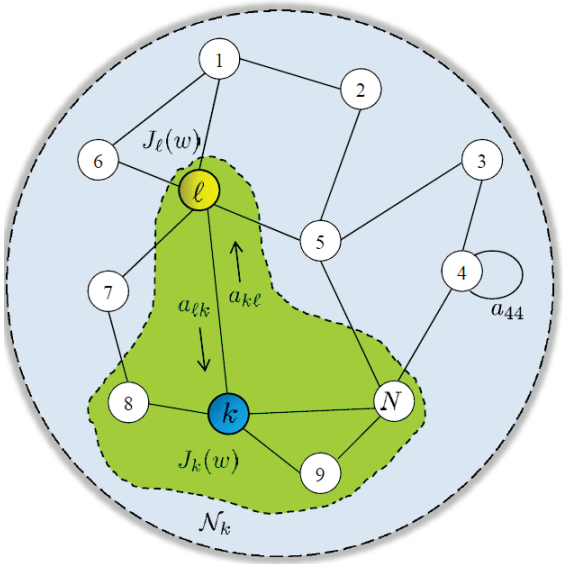
$$J_k(w^\bullet) \leq J_k(w^\star), \quad \text{for all } k \in \{1, 2, \ldots, N\} \tag{8.65}$$

$$J_{k^o}(w^\bullet) < J_{k^o}(w^\star), \quad \text{for at least one } k^o \in \{1, 2, \ldots, N\} \tag{8.66}$$

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.

Pareto Solution

$w^\star$ is **Pareto optimal** if there does not exist $w^\bullet$ that dominates $w^\star$:



$$J_k(w^\bullet) \leq J_k(w^\star), \quad \text{for all } k \in \{1,2,\ldots,N\}$$

$$J_{k^o}(w^\bullet) < J_{k^o}(w^\star), \quad \text{for at least one } k^o \in \{1,2,\ldots,N\}$$

Pareto Solution

→ No agent in a cooperative network can have its performance improved by **moving away** from a Pareto solution without **degrading** the performance of at least one other agent.

Pareto Solution

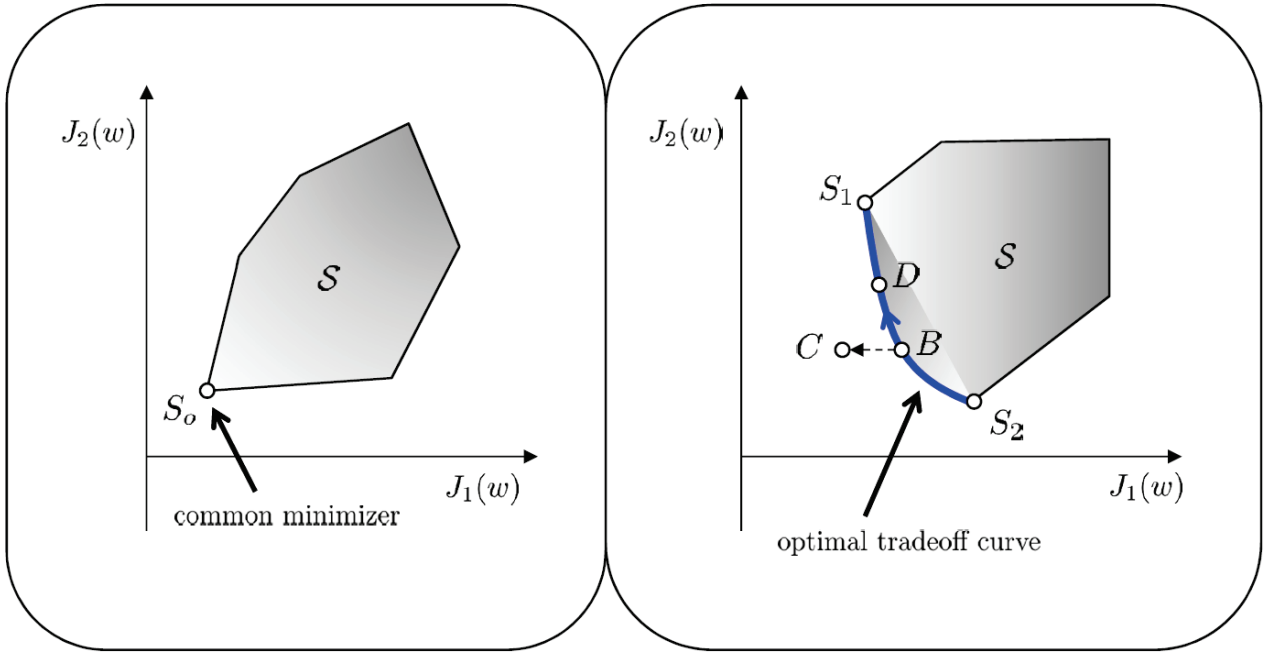


Figure 8.1: Pareto optimal points for the case $N = 2$. In the figure on the left, point S denotes the optimal point where both cost functions are minimized simultaneously. In the figure on the right, all points that lie on the heavy boundary curve are Pareto optimal solutions.

Pareto Solution

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

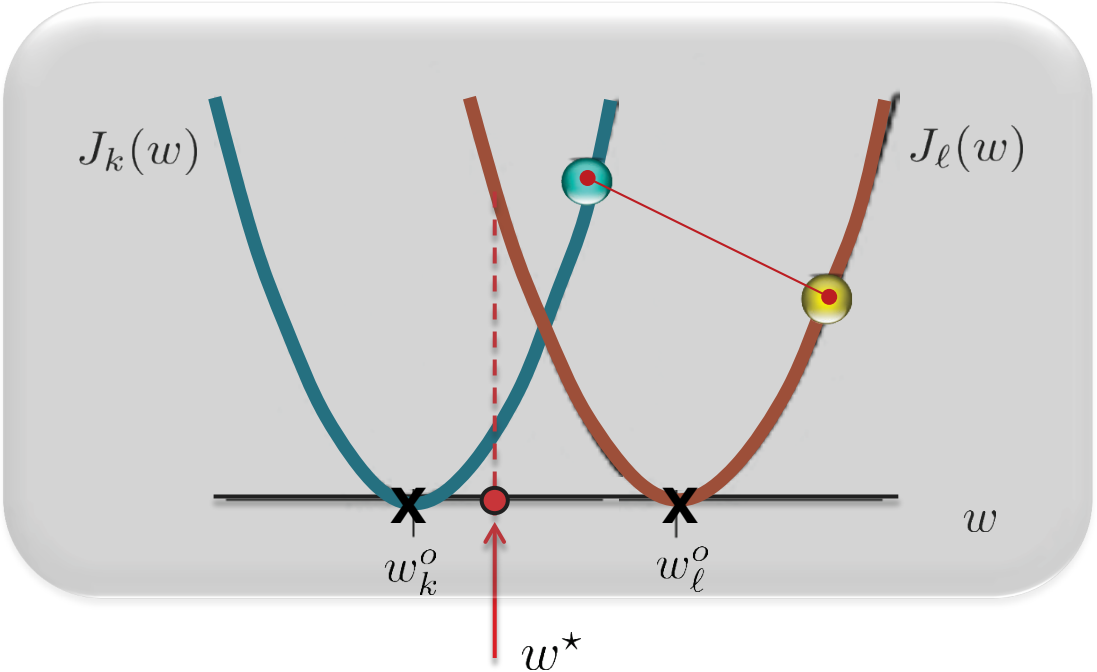
Pareto Solution

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.

Pareto Solution

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].

Example #B (Quadratic Costs)



one Pareto solution

Pareto Solution

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]:

$$J^{\text{glob},\pi}(w) \triangleq \sum_{k=1}^{N} \pi_k \, J_k(w) \tag{8.68}$$

compare with → $J^{\text{glob},\star}(w) \triangleq \sum_{k=1}^{N} q_k J_k(w)$

Pareto Solution

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

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:

$$J_k(w) \;=\; \sigma_d^2 \;-\; r_{du,k}^{*} w - w^{*} r_{du,k} + w^{*} R_{u,k} w, \quad k = 1, 2, \ldots, N \qquad (8.69)$$

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

$$w_k^o \;=\; R_{u,k}^{-1} r_{du,k} \qquad (8.70)$$

Example #8.7

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:

$$w^{\pi} = \left(\sum_{k=1}^{N} \pi_k R_{u,k}\right)^{-1} \left(\sum_{k=1}^{N} \pi_k \, r_{du,k}\right) \tag{8.71}$$

Example #8.7

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.

Example #8.7

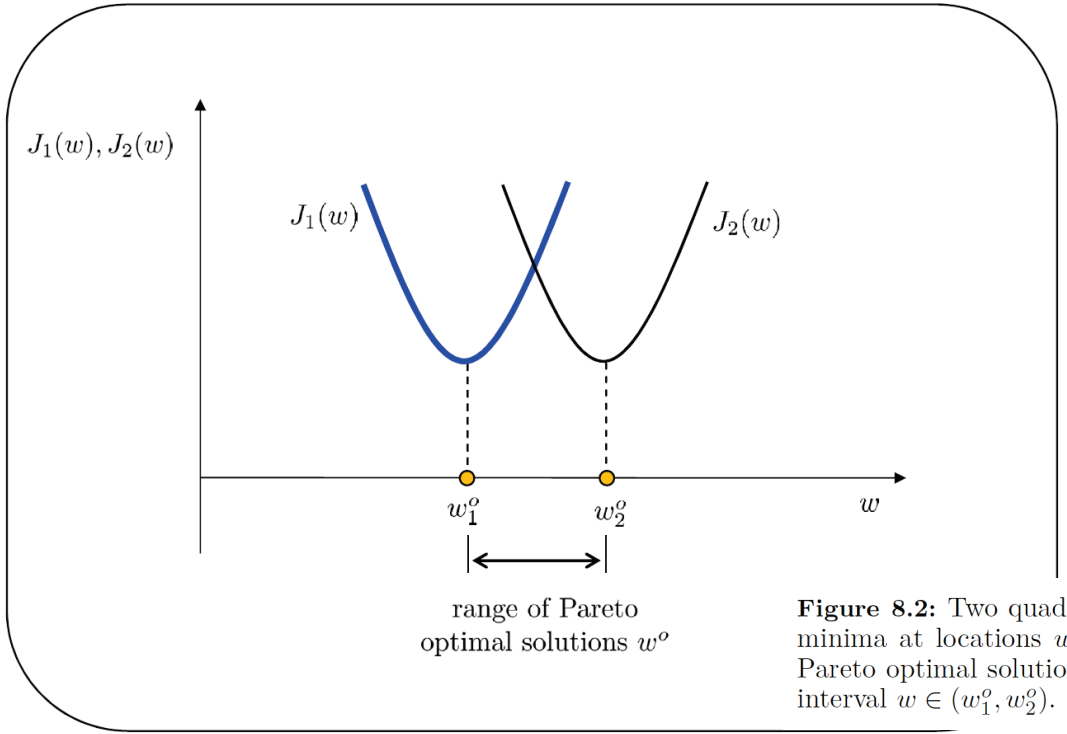


Figure 8.2: Two quadratic cost functions of a scalar real parameter w with minima at locations $w = w_1^o$ and $w = w_2^o$. As shown by (8.75), the set of Pareto optimal solutions in this case consists of all parameters w within the interval $w \in (w_1^o, w_2^o)$.

Example #8.7

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 $\{\sigma^2_{u,1}, \sigma^2_{u,2}\}$ so that expression (8.72) becomes

$$w^{\pi} = \left(\frac{\pi_1 \sigma^2_{u,1}}{\pi_1 \sigma^2_{u,1} + \pi_2 \sigma^2_{u,2}}\right) w_1^o + \left(\frac{\pi_2 \sigma^2_{u,2}}{\pi_1 \sigma^2_{u,1} + \pi_2 \sigma^2_{u,2}}\right) w_2^o \qquad (8.75)$$

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

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

$$r_{du,k} = R_{u,k} w_k^o, \qquad k = 1, 2, \ldots, N \tag{8.77}$$

Example #8.8

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

$$\begin{aligned} \nabla_w J_k(w) &= R_{u,k}w - r_{du,k} \\ &\overset{(8.77)}{=} R_{u,k}(w - w_k^o) \end{aligned} \tag{8.79}$$

Example #8.8

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,

$$\text{consensus:} \quad A_o = A, \quad A_1 = I_N = A_2 \tag{8.80}$$

$$\text{CTA diffusion:} \quad A_1 = A, \quad A_2 = I_N = A_o \tag{8.81}$$

$$\text{ATC diffusion:} \quad A_2 = A, \quad A_1 = I_N = A_o \tag{8.82}$$

In these cases, the Perron eigenvector p defined by (8.49) will correspond to the Perron eigenvector associated with A:

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0 \tag{8.83}$$

Example #8.8

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:

$$\sum_{k=1}^{N} \mu_k p_k R_{u,k}(w^\star - w_k^o) = 0 \tag{8.85}$$

Example #8.8

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:

$$w^\star = \sum_{k=1}^{N} \pi_k w_k^o \tag{8.87}$$

Example #8.8

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:

$$a_{\ell k} = \begin{cases} 1/n_k, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{8.89}$$

Example #8.8

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

$$\pi_k \triangleq n_k \sigma_{u,k}^2 \left(\sum_{k=1}^{N} n_k \sigma_{u,k}^2 \right)^{-1}, \quad k = 1, 2, \ldots, N \tag{8.92}$$

Example #8.8

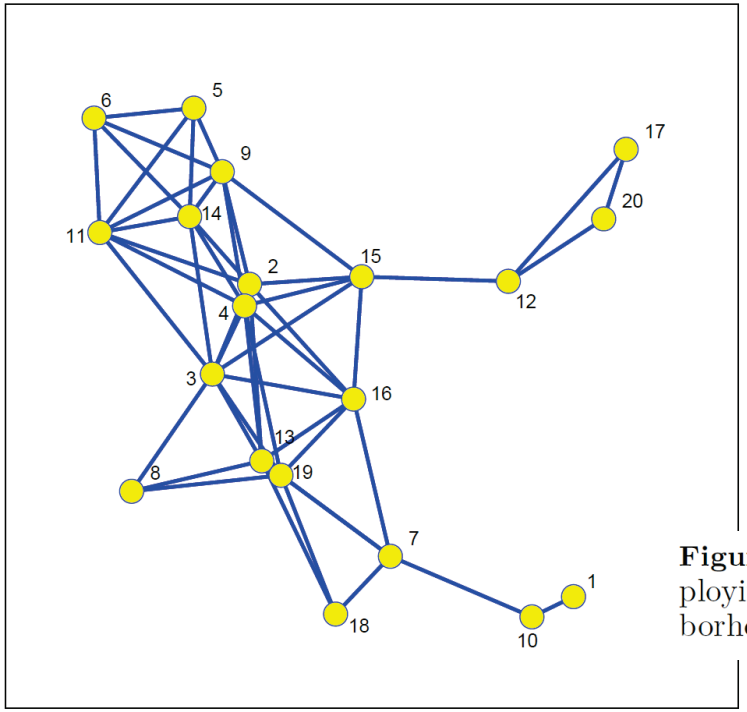


Figure 8.3: A connected network topology consisting of $N = 20$ agents employing the averaging rule (8.89). Each agent k is assumed to belong its neighborhood $\mathcal{N}_k$. It follows that the network is strongly-connected.

Example #8.8

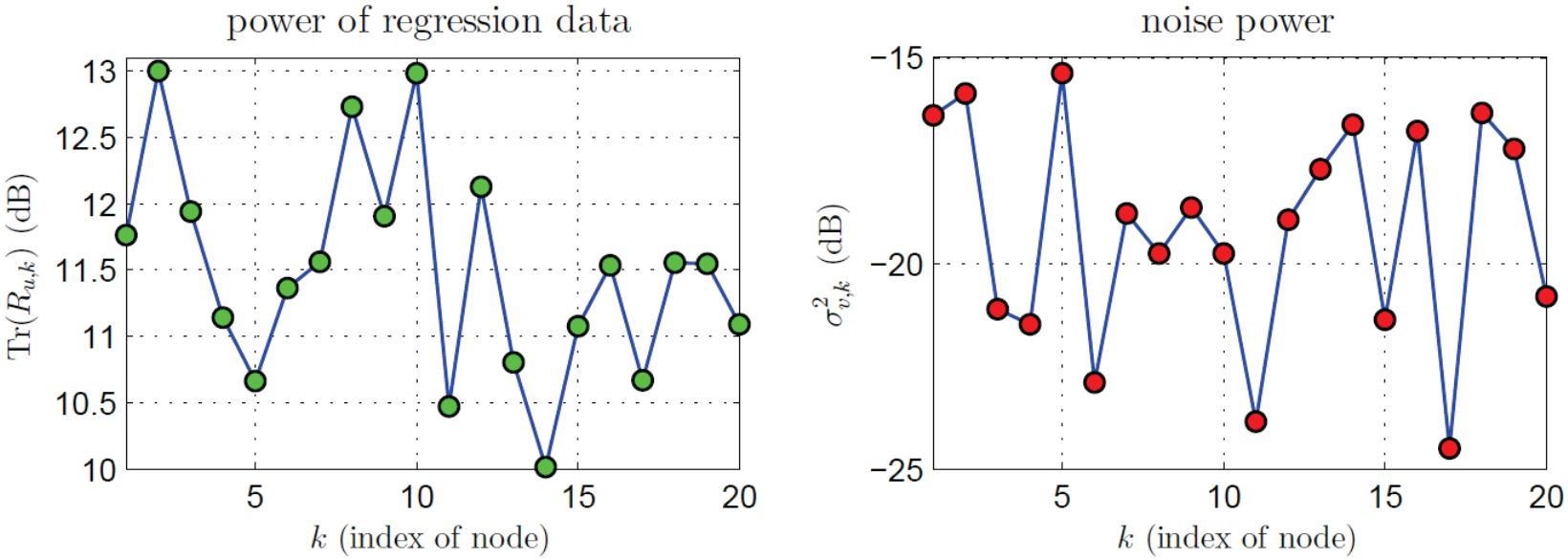


Figure 8.4: Measurement noise profile (right) and regression data power (left) across all agents in the network. The covariance matrices are assumed to be of the form $R_{u,k} = \sigma^2_{u,k} I_M$, and the noise and regression data are Gaussian distributed in this simulation.

Example #8.8

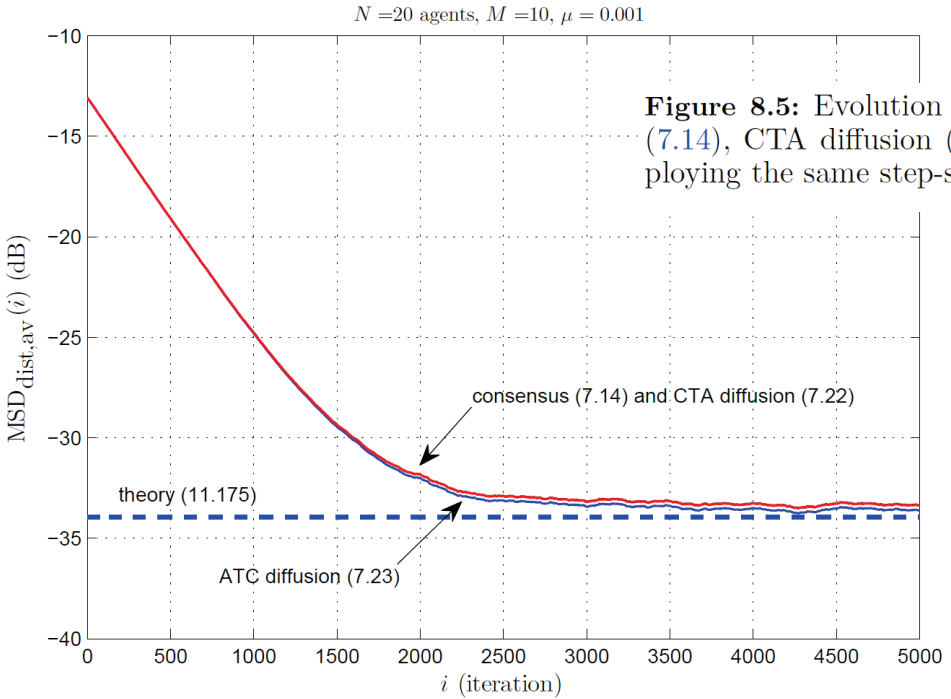


Figure 8.5: Evolution of the learning curves for three strategies: consensus (7.14), CTA diffusion (7.22), and ATC diffusion (7.23), with all agents employing the same step-size $\mu = 0.001$ and the averaging combination policy.

Example #8.8

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)

Example #8.8

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

Example #8.8

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 $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w_k^o + \boldsymbol{v}_k(i)$, with $M = 10$. The unknown vectors $\{w_k^o\}$ 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

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 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:

Example #8.9

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0 \tag{8.94}$$

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

Example #8.9

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:

Example #8.9

$$a_{\ell k} = \begin{cases} \dfrac{\mu_k/q'_k}{\max\{\, n_k\mu_k/q'_k,\ n_\ell\mu_\ell/q'_\ell \,\}}, & \ell \in \mathcal{N}_k\backslash\{k\} \\ 1 - \displaystyle\sum_{m\in\mathcal{N}_k\backslash\{k\}} a_{mk}, & \ell = k \end{cases} \tag{8.96}$$

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):

$$n_k \triangleq |\mathcal{N}_k| \tag{8.97}$$

Example #8.9

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 $J^{\text{glob}}(w)$ and $J^{\text{glob},\star}(w)$ defined by (8.44) and (8.53) will coincide, namely, $w^\star = w^o$, and construction (8.96) will reduce to

Example #8.9

$$a_{\ell k} = \begin{cases} \dfrac{\mu_k}{\max\{\, n_k\mu_k,\ n_\ell\mu_\ell \,\}}, & \ell \in \mathcal{N}_k \backslash \{k\} \\ 1 - \displaystyle\sum_{m \in \mathcal{N}_k \backslash \{k\}} a_{mk}, & \ell = k \end{cases} \tag{8.99}$$

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

Example #8.9

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

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.

Example #8.10

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):

Example #8.10

$$\left\{\begin{array}{rcl} \boldsymbol{\psi}_{k,i} &=& \boldsymbol{w}_{k,i-1} - \frac{\mu_o}{p_k}\,\widehat{\nabla_{w^*} J_k}\,(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} &=& \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{array}\right. \tag{8.102}$$

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.

Example #8.10

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:

Example #8.10

$$r_i = Ar_{i-1}, \quad r_{-1} \neq 0, \quad i \geq 0 \tag{8.104}$$

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

Example #8.10

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):

$$\begin{cases} r_k(i) &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{k\ell}\, r_k(i-1) \\ \boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{k,i-1} - \frac{\mu_o}{r_k(i)}\, \widehat{\nabla_{w^*} J_k}\,(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{\psi}_{\ell,i} \end{cases} \tag{8.105}$$

Example #8.10

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].

■

End of Lecture

Proc. IEEE, vol. 102, no. 4, pp. 460-497, April 2014.
Foundations and Trends in Machine Learning, vol. 7, no. 4-5, pp. 311-801, July 2014.