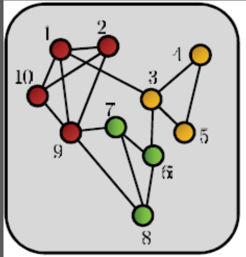
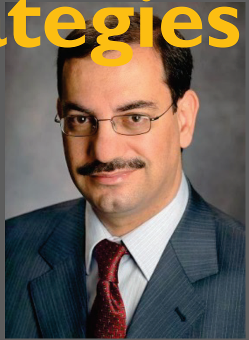


INFERENCE OVER NETWORKS

LECTURE #15: Multi-Agent Distributed Strategies

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.

References

Chapter 7 (Multi-Agent Distributed Strategies, pp. 368-448-469):
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.

Appendix E (Comparison with Consensus Strategies):
A. H. Sayed, ``Diffusion adaptation over networks," in ***Academic Press Library in Signal Processing***, vol. 3, R. Chellapa and S. Theodoridis, *editors*, pp. 323-454, Academic Press, Elsevier, 2014.

Setting

There are several distributed strategies that can be used to seek the minimizer of (6.12), namely,

$$w^o \triangleq \arg\min_{w} \sum_{k=1}^{N} J_k(w) \tag{7.1}$$

Setting

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].

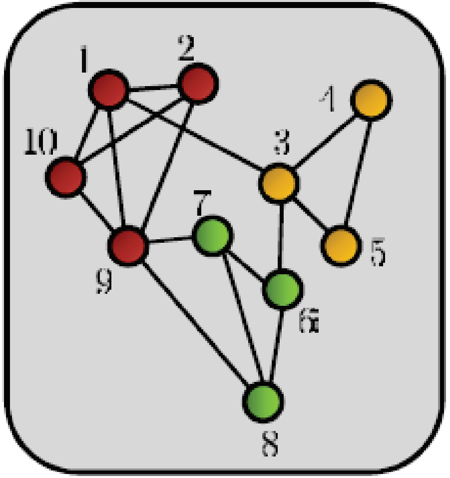
Setting

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:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \frac{\mu}{N}\sum_{k=1}^{N} \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{i-1}), \quad i \geq 0 \qquad (7.2)$$

Centralized Strategy

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



Centralized strategy:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \frac{\mu}{N}\sum_{k=1}^{N} \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{i-1}), \quad i \geq 0$$

→ Can be used to motivate distributed implementations.

Incremental Strategy

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.

Incremental Strategy

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \frac{\mu}{N}\sum_{k=1}^{N}\widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{i-1}), \quad i \geq 0 \qquad (7.2)$$

(still not distributed!)

$$\left\{\begin{array}{lcll}
\boldsymbol{w}_{1,i} & = & \boxed{\boldsymbol{w}_{i-1}} & - \frac{\mu}{N}\widehat{\nabla_{w^*} J}_1(\underline{\boldsymbol{w}_{i-1}}) \\
\boldsymbol{w}_{2,i} & = & \boldsymbol{w}_{1,i} & - \frac{\mu}{N}\widehat{\nabla_{w^*} J}_2(\underline{\boldsymbol{w}_{i-1}}) \\
\boldsymbol{w}_{3,i} & = & \boldsymbol{w}_{2,i} & - \frac{\mu}{N}\widehat{\nabla_{w^*} J}_3(\boldsymbol{w}_{i-1}) \\
\vdots & = & & \vdots \\
\boxed{\boldsymbol{w}_i} & = & \boldsymbol{w}_{N-1,i} & - \frac{\mu}{N}\widehat{\nabla_{w^*} J}_N(\underline{\boldsymbol{w}_{i-1}})
\end{array}\right. \qquad (7.3)$$

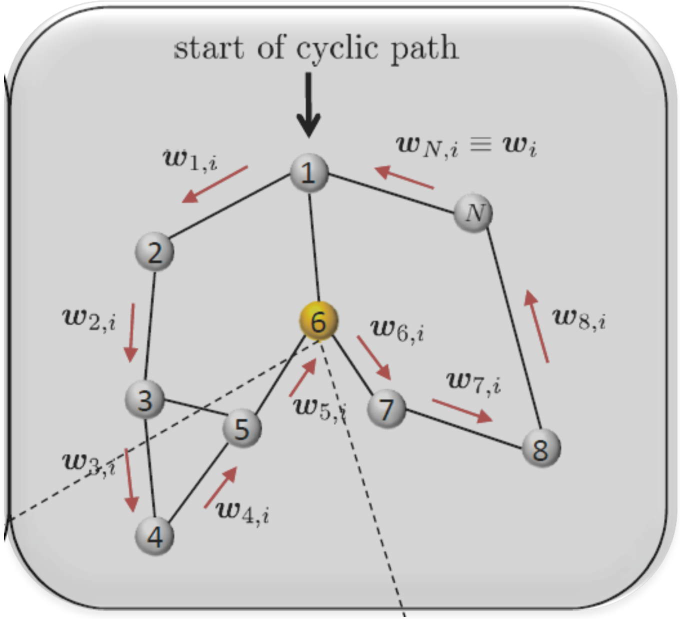
Incremental Strategy

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]).

Incremental Strategy

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k-1,i} - \frac{\mu}{N}\widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{i-1})$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k-1,i} - \frac{\mu}{N}\widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k-1,i})$$



Incremental Strategy

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)

Incremental Strategy

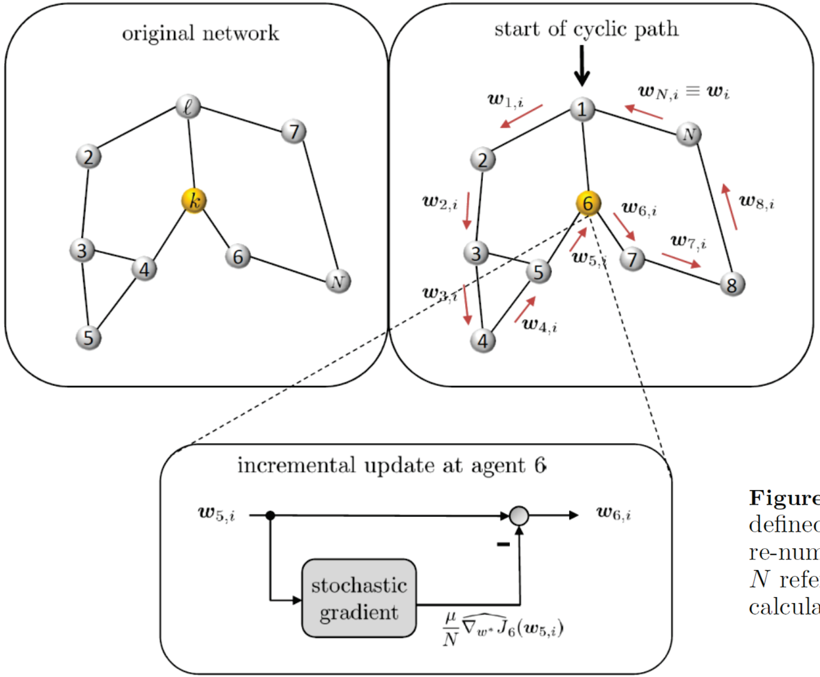


Figure 7.1: Starting from the given network on the left, a cyclic path is defined that visits all agents and is shown on the right. The agents are then re-numbered with agent 1 referring to the start of the cyclic path and agent N referring to its end. The diagram in the bottom illustrates the incremental calculations that are carried out by agent 6.

Example #7.1

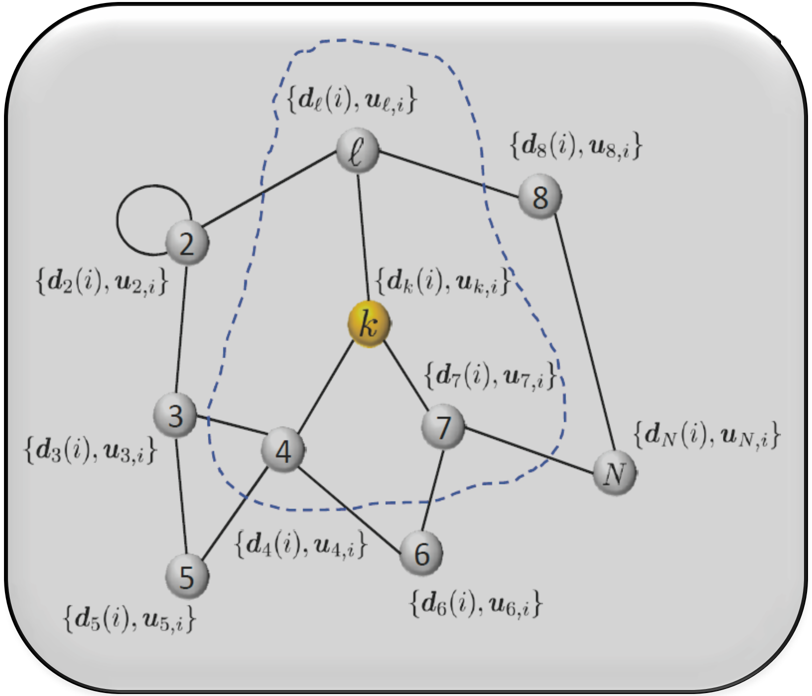
Example 7.1 (Incremental LMS networks). For the MSE network of Example 6.3, once a cyclic path has been determined and the agents renumbered from 1 to N, the incremental strategy (7.6) reduces to the following incremental LMS algorithm from [55, 156, 161, 209]:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k-1,i} + \frac{2\mu}{Nh}\boldsymbol{u}_{k,i}^{*}[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k-1,i}] \tag{7.7}$$

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, $\boldsymbol{u}_{k,i}^{\mathsf{T}}$.

■

Example #7.1


$\{d_\ell(i), u_{\ell,i}\}$
$\{d_8(i), u_{8,i}\}$
$\{d_2(i), u_{2,i}\}$
$\{d_k(i), u_{k,i}\}$
$\{d_7(i), u_{7,i}\}$
$\{d_3(i), u_{3,i}\}$
$\{d_N(i), u_{N,i}\}$
$\{d_4(i), u_{4,i}\}$
$\{d_5(i), u_{5,i}\}$
$\{d_6(i), u_{6,i}\}$
ℓ
8
2
k
3
4
7
N
5
6

Difficulties for Adaptation

- Sensitivity to agent or link failures.
- Determining a cyclic path is generally NP-hard.
- Cooperation between agents is limited.
- For every iteration, it is necessary to perform N incremental steps.

Consensus Strategy

Consensus Strategy

Incremental:

$$\boldsymbol{w}_{k,i} = \underbrace{\boldsymbol{w}_{k-1,i}}_{\text{(coop)}} - \frac{\mu}{N}\widehat{\nabla_{w^*} J}_k(\underbrace{\boldsymbol{w}_{k-1,i}}_{\text{(decen)}})$$

Consensus:

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

Consensus Strategy

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).

Consensus Strategy

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 $k = 1, 2, \ldots, N$:

$$a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad \text{and} \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{7.10}$$

Consensus Strategy

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.,

Consensus Strategy

$$A^{\mathsf{T}}\mathbb{1} = \mathbb{1} \tag{7.11}$$

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).

Consensus Strategy

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.

Consensus Strategy

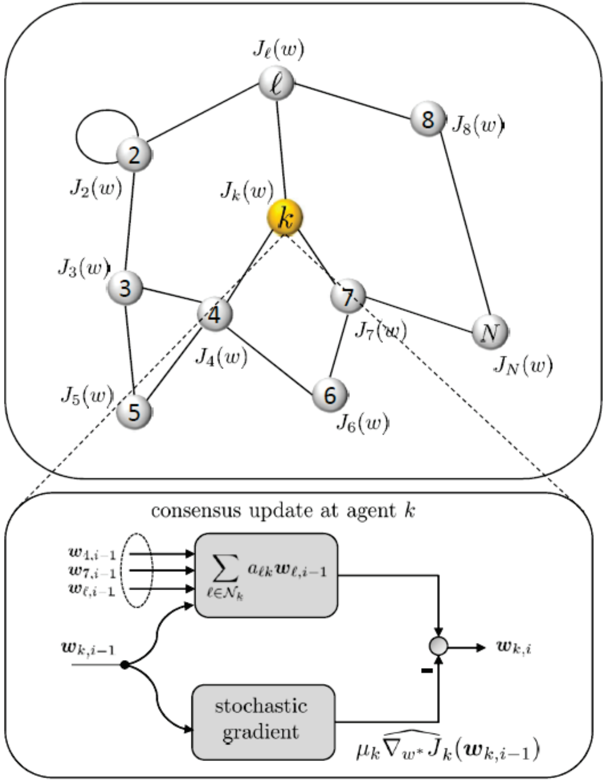
Consensus strategy for adaptation and learning

for each time instant $i \geq 0$:

each agent $k = 1, 2, \ldots, N$ performs the update:

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

end



Consensus Strategy

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

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:

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} + \frac{2\mu_k}{h} \boldsymbol{u}_{k,i}^* [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1}] \tag{7.13}$$

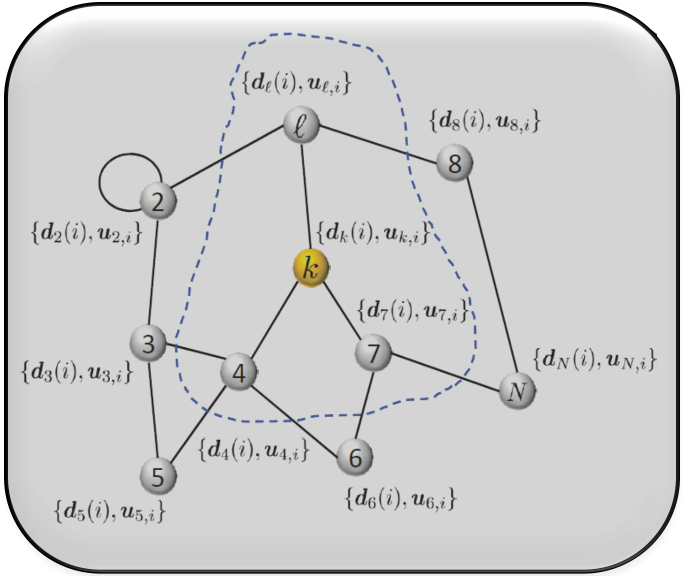
or

$$\begin{cases} \boldsymbol{\psi}_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} + \frac{2\mu_k}{h} \boldsymbol{u}_{k,i}^* [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1}] \end{cases} \tag{7.14}$$

Example #7.2

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1} + 2\mu_k u_{k,i}^{\mathsf{T}} [d_k(i) - u_{k,i} w_{k,i-1}]$$

$$\begin{cases} \psi_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1} \\ w_{k,i} = \psi_{k,i-1} + 2\mu_k u_{k,i}^{\mathsf{T}} [d_k(i) - u_{k,i} w_{k,i-1}] \end{cases}$$



Diffusion Strategy

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.

Asymmetric Update

Incremental:

$$\boldsymbol{w}_{k,i} = \underbrace{\boldsymbol{w}_{k-1,i}}_{\text{(coop)}} - \frac{\mu}{N}\widehat{\nabla_{w^*} J}_k\big(\underbrace{\boldsymbol{w}_{k-1,i}}_{\text{(decen)}}\big)$$

Coop and decen terms are identical

Consensus:

$$\boldsymbol{w}_{k,i} = \underbrace{\sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1}}_{\text{(coop)}} - \mu_k \widehat{\nabla_{w^*} J}_k\big(\underbrace{\boldsymbol{w}_{k,i-1}}_{\text{(decen)}}\big)$$

Coop and decen terms are **asymmetric**

Asymmetric Update

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.

Diffusion Strategy

Asymmetry causes instability → resolved by diffusion.

Combine-then-Adapt (CTA) Diffusion:

$$\boldsymbol{w}_{k,i} = \underbrace{\sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1}}_{\text{(coop)}} - \mu_k \widehat{\nabla_{w^*} J}_k \underbrace{\left(\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{w}_{\ell,i-1} \right)}_{\text{(decen)}}$$

CTA Diffusion

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 $\boldsymbol{\psi}_{k,i-1}$.

CTA Diffusion

$$\boldsymbol{w}_{k,i} = \underbrace{\sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1}}_{\text{(coop)}} - \mu_k \widehat{\nabla_{w^*} J}_k \underbrace{\left(\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{w}_{\ell,i-1} \right)}_{\text{(decen)}}$$

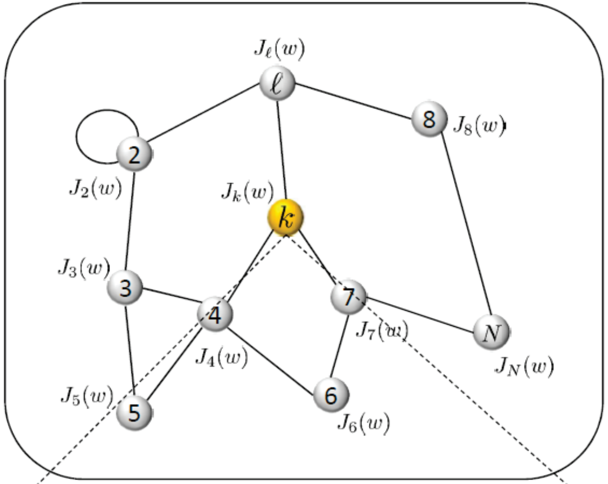
Diffusion strategy for adaptation and learning (CTA)

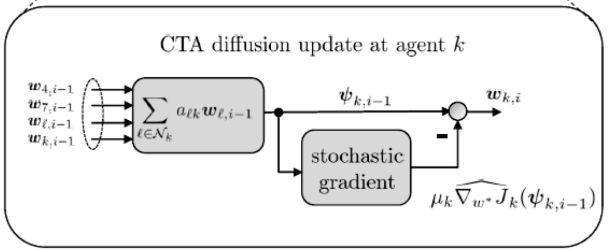
for each time instant $i \geq 0$:

each agent $k = 1, 2, \ldots, N$ performs the update:

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

end





ATC Diffusion

Adapt-then-Combine (ATC) Diffusion

Diffusion strategy for adaptation and learning (ATC)

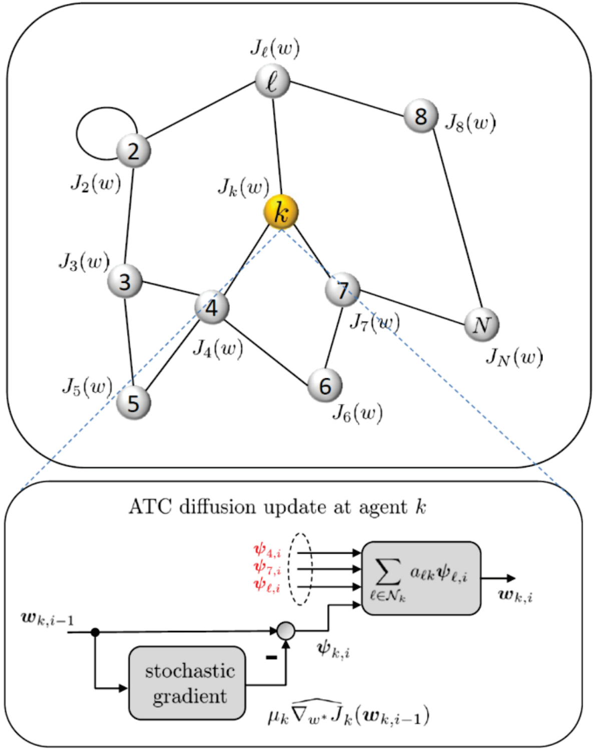
for each time instant $i \geq 0$:

each agent $k = 1, 2, \ldots, N$ performs the update:

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

end

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



Example #7.3

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:

$$\left\{\begin{array}{lcl} \boldsymbol{\psi}_{k,i-1} & = & \sum\limits_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} & = & \boldsymbol{\psi}_{k,i-1} + \dfrac{2\mu_k}{h}\boldsymbol{u}^*_{k,i}\left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k,i-1}\right] \end{array}\right. \quad \text{(CTA)} \qquad (7.22)$$

and

$$\left\{\begin{array}{lcl} \boldsymbol{\psi}_{k,i} & = & \boldsymbol{w}_{k,i-1} + \dfrac{2\mu_k}{h}\boldsymbol{u}^*_{k,i}\left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}\right] \\ \boldsymbol{w}_{k,i} & = & \sum\limits_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{\psi}_{\ell,i} \end{array}\right. \quad \text{(ATC)} \qquad (7.23)$$

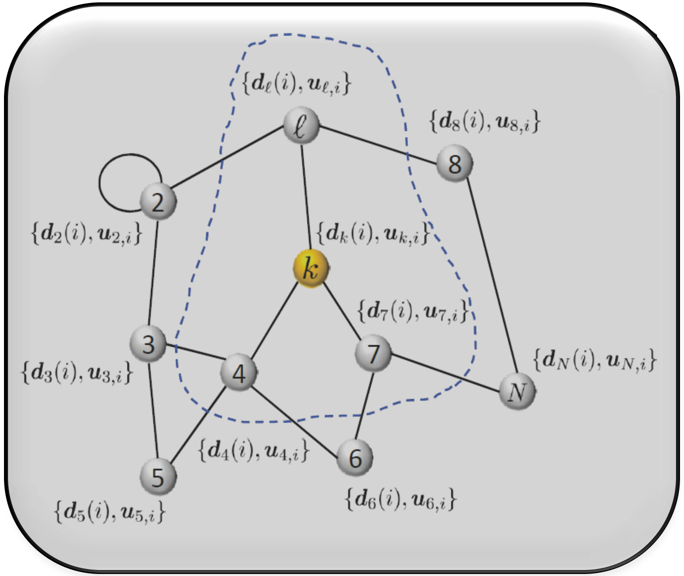
Example #7.3

$$\begin{cases} \psi_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1} \\ w_{k,i} = \psi_{k,i-1} + \frac{2\mu_k}{h} u_{k,i}^* \left[d_k(i) - u_{k,i}\psi_{k,i-1} \right] \end{cases}$$

(CTA)

$$\begin{cases} \psi_{k,i} = w_{k,i-1} + \frac{2\mu_k}{h} u_{k,i}^* \left[d_k(i) - u_{k,i} w_{k,i-1} \right] \\ w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \psi_{\ell,i} \end{cases}$$

(ATC)



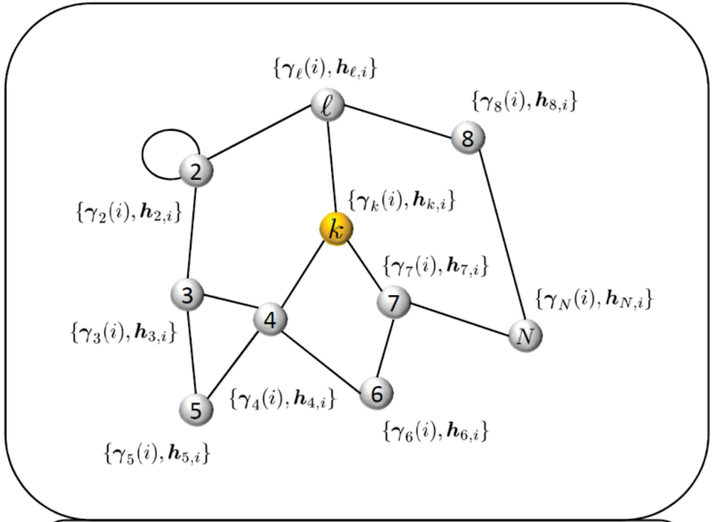
Example #7.4

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 re-

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:

$$J(w) \stackrel{\Delta}{=} \frac{\rho}{2}\|w\|^2 + \mathbb{E}\left\{\ln\left(1 + e^{-\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}} w}\right)\right\} \qquad (7.24)$$

Example #7.4



$$J_k(w) \equiv J(w) = \frac{\rho}{2}\|w\|^2 + \mathbb{E}\left\{\ln\left[1 + e^{-\gamma_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}} w}\right]\right\}$$

logistic network

Figure 7.5: Each agent k receives streaming data $\{\gamma_k(i), \boldsymbol{h}_{k,i}\}$. The agents cooperate to minimize the regularized logistic cost (7.24).

Example #7.4

where $J(w)$ is the same for all agents. The corresponding loss function is

$$Q(w;\boldsymbol{\gamma}_k(i),\boldsymbol{h}_{k,i}) \triangleq \frac{\rho}{2}\|w\|^2 + \ln\left(1+e^{-\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}}w}\right) \qquad (7.25)$$

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 strategy for solving the logistic regression problem cooperatively:

Example #7.4

$$\begin{cases} \boldsymbol{\psi}_{k,i} &= (1-\rho\mu_k)\boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}\left(\dfrac{1}{1+e^{\boldsymbol{\gamma}_k(i)\boldsymbol{h}_{k,i}^{\mathsf{T}}\boldsymbol{w}_{k,i-1}}}\right) \\ \boldsymbol{w}_{k,i} &= \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,i} \end{cases} \tag{7.26}$$

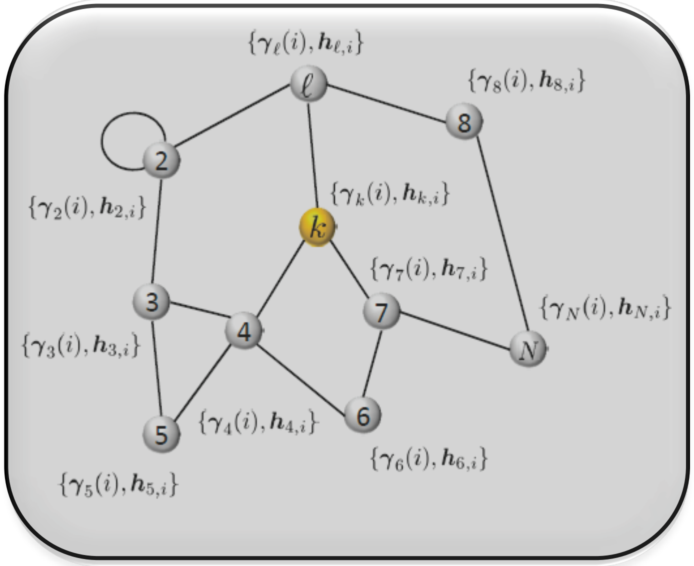
■

Example #7.4

Diffusion Logistic Regression

$$J(w) \triangleq \frac{\rho}{2}\|w\|^2 + \mathbb{E}\left\{\ln\left[1+e^{-\gamma_k(i)h_{k,i}^{\mathsf{T}}w}\right]\right\}$$

$$\begin{cases} \psi_{k,i} = (1-\rho\mu_k)w_{k,i-1} + \mu_k\left(\dfrac{\gamma_k(i)}{1+e^{\gamma_k(i)h_{k,i}^{\mathsf{T}}w_{k,i-1}}}\right)h_{k,i} \\ w_{k,i} = \displaystyle\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\psi_{\ell,i} \end{cases}$$



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]:

Enlarged Cooperation

$$\left\{\begin{array}{lcl} \boldsymbol{\psi}_{k,i-1} & = & \sum\limits_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} & = & \boldsymbol{\psi}_{k,i-1} - \mu_k \sum\limits_{\ell\in\mathcal{N}_k} c_{\ell k}\widehat{\nabla_{w^*} J}_{\ell}(\boldsymbol{\psi}_{k,i-1}) \end{array}\right. \quad \text{(CTA)} \qquad (7.27)$$

and

$$\left\{\begin{array}{lcl} \boldsymbol{\psi}_{k,i} & = & \boldsymbol{w}_{k,i-1} - \mu_k \sum\limits_{\ell\in\mathcal{N}_k} c_{\ell k}\widehat{\nabla_{w^*} J}_{\ell}(\boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} & = & \sum\limits_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{\psi}_{\ell,i} \end{array}\right. \quad \text{(ATC)} \qquad (7.28)$$

Enlarged Cooperation

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.,

$$C\mathbb{1} = \mathbb{1} \tag{7.30}$$

Enlarged Cooperation

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

Discussion

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].

Discussion

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 $\overline{w}$.

Discussion

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

$$|\lambda_2(A)| < 1 \tag{7.34}$$

Discussion

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

Discussion

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, $\overline{w}$.

Discussion

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]).

Discussion

An example of a two-time scale implementation would be an algorithm of the following form:

$$
\left\{
\begin{array}{l}
\boldsymbol{w}_{\ell,i-1}^{(-1)} \;\longleftarrow\; \boldsymbol{w}_{\ell,i-1}, \quad \text{for all agents } \ell \text{ at iteration } i-1 \\
\textbf{for } n = 0, 1, 2, \ldots, J-1 \text{ iterate:} \\
\qquad \boldsymbol{w}_{k,i-1}^{(n)} \;=\; \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{w}_{\ell,i-1}^{(n-1)}, \quad \text{for all } k = 1, 2, \ldots, N \\
\textbf{end} \\
\boldsymbol{w}_{k,i} \;=\; \boldsymbol{w}_{k,i-1}^{(J-1)} \;-\; \mu_k \, \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1})
\end{array}
\right.
\tag{7.37}
$$

Discussion

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).

Discussion

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].

Discussion

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:

Discussion

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \;-\; \mu_k(i)\widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \tag{7.38}$$

or variations thereof, such as replacing $\mu_k(i)$ by some time-variant gain matrix sequence, say, $K_{k,i}$:

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \;-\; K_{k,i} \cdot \widehat{\nabla_{w^*} J}_k(\boldsymbol{w}_{k,i-1}) \tag{7.39}$$

Discussion

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.

Discussion

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].

Discussion

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.

More on Consensus Strategy

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

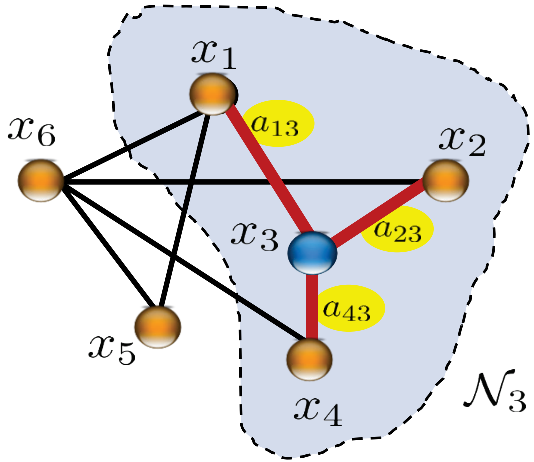
Appendix E (Comparison with Consensus Strategies):

A. H. Sayed, ``Diffusion adaptation over networks," in ***Academic Press Library in Signal Processing***, vol. 3, R. Chellapa and S. Theodoridis, *editors*, pp. 323-454, Academic Press, Elsevier, 2014.

Classical Consensus (1974)

Each agent k has a measurement x_k.

Objective: Compute average value in a distributed manner.



Setting

Consider a connected network consisting of N nodes. Each node has a state or measurement value x_k, possibly a vector of size $M \times 1$. All nodes in the network are interested in evaluating the average value of their states, which we denote by

$$w^o \triangleq \frac{1}{N}\sum_{k=1}^{N} x_k \tag{608}$$

Consensus Strategy

A centralized solution to this problem would require each node to transmit its measurement x_k to a fusion center. The central processor would then compute w^o using (608) and transmit it back to all nodes. This centralized mode of operation suffers from at least two limitations. First, it requires communications and power resources to transmit the data back and forth between the nodes and the central processor; this problem is compounded if the fusion center is stationed at a remote location. Second, the architecture has a critical point of failure represented by the central processor; if it fails, then operations would need to be halted.

Consensus Strategy

The consensus strategy provides an elegant distributed solution to the same problem, whereby nodes interact locally with their neighbors and are able to converge to w^o through these interactions. Thus, consider an arbitrary node k and assign nonnegative weights $\{a_{\ell k}\}$ to the edges linking k to its neighbors $\ell \in \mathcal{N}_k$. For each node k, the weights $\{a_{\ell k}\}$ are assumed to add up to one so that

$$\text{for } k = 1, 2, \ldots, N:$$

$$a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{609}$$

Consensus Strategy

The resulting combination matrix is denoted by A and its $k-$th column consists of the entries $\{a_{\ell k}, \ell = 1, 2, \ldots, N\}$. In view of (609), the combination matrix A is seen to satisfy $A^T\mathbb{1} = \mathbb{1}$. That is, A is left-stochastic. The consensus strategy can be described as follows. Each node k operates repeatedly on the data from its neighbors and updates its state iteratively according to the rule:

$$\boxed{w_{k,n} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, w_{\ell,n-1}, \quad n > 0} \tag{610}$$

Consensus Strategy

where $w_{\ell,n-1}$ denotes the state of node ℓ at iteration $n-1$, and $w_{k,n}$ denotes the updated state of node k after iteration n. The initial conditions are

$$w_{k,o} \;=\; x_k, \qquad k = 1, 2, \ldots, N \tag{611}$$

If we collect the states of all nodes at iteration n into a column vector, say,

$$z_n \;\stackrel{\Delta}{=}\; \text{col}\{w_{1,n},\; w_{2,n},\; \ldots, w_{N,n}\} \tag{612}$$

Then, the consensus iteration (610) can be equivalently rewritten in vector form as follows:

$$\boxed{z_n \;=\; A^T z_{n-1}, \quad n > 0} \tag{613}$$

Consensus Strategy

where

$$\mathcal{A}^T = A^T \otimes I_M \tag{614}$$

The initial condition is

$$z_o \stackrel{\Delta}{=} \text{col}\{x_1,\ x_2,\ \ldots,x_N\} \tag{615}$$

Error Recursion

Note that we can express the average value, w^o, from (608) in the form

$$\boxed{w^o = \frac{1}{N} \cdot (\mathbb{1}^T \otimes I_M) \cdot z_o} \tag{616}$$

where $\mathbb{1}$ is the vector of size $M \times 1$ and whose entries are all equal to one. Let

$$\widetilde{w}_{k,n} = w^o - w_{k,n} \tag{617}$$

denote the weight error vector for node k at iteration n; it measures how far the iterated state is from the desired average value w^o. We collect all error vectors across the network into an $N \times 1$ block column vector whose entries are of size $M \times 1$ each:

Error Recursion

$$\widetilde{w}_n \triangleq \begin{bmatrix} \widetilde{w}_{1,n} \\ \widetilde{w}_{2,n} \\ \vdots \\ \widetilde{w}_{N,n} \end{bmatrix} \tag{618}$$

Then,

$$\boxed{\widetilde{w}_n = (\mathbb{1} \otimes I_M)w^o - z_n} \tag{619}$$

The following result is a classical result on consensus strategies [42–44]. It provides conditions under which the state of all nodes will converge to the desired average, w^o, so that $\widetilde{w}_n$ will tend to zero.

Convergence Condition

***Theorem* E.1. (Convergence to Consensus)** *For any initial states $\{x_k\}$, the successive iterates $w_{k,n}$ generated by the consensus iteration (610) converge to the network average value w^o as $n \to \infty$ if, and only if, the following three conditions are met:*

$$A^T \mathbb{1} = \mathbb{1} \tag{620}$$

$$A\mathbb{1} = \mathbb{1} \tag{621}$$

$$\rho\left(A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T\right) < 1 \tag{622}$$

That is, the combination matrix A needs to be doubly stochastic, and the matrix $A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T$ needs to be stable.

Proof

Proof. (*Sufficiency*). Assume first that the three conditions stated in the theorem hold. Since A is doubly stochastic, then so is any power of A, say, A^n for any $n \geq 0$, so that

$$[A^n]^T \mathbb{1} = \mathbb{1}, \qquad A^n \mathbb{1} = \mathbb{1} \tag{623}$$

Using this fact, it is straightforward to verify by induction the validity of the following equality:

$$\left(A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T\right)^n \;=\; [A^n]^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T \tag{624}$$

Likewise, using the Kronecker product identities

$$(E + B) \otimes C \;=\; (E \otimes C) \;+\; (B \otimes C) \tag{625}$$

$$(E \otimes B)(C \otimes D) \;=\; (EC \otimes BD) \tag{626}$$

$$(E \otimes B)^n \;=\; E^n \otimes B^n \tag{627}$$

Proof

for matrices $\{E, B, C, D\}$ of compatible dimensions, we observe that

$$
\begin{aligned}
(\mathcal{A}^n)^T - \frac{1}{N}\cdot(\mathbb{1}\otimes I_M)\cdot(\mathbb{1}^T\otimes I_M) &= \left[(A^n)^T \otimes I_M\right] - \frac{1}{N}\cdot(\mathbb{1}\mathbb{1}^T\otimes I_M) \\
&= \left[(A^n)^T - \frac{1}{N}\cdot\mathbb{1}\mathbb{1}^T\right]\otimes I_M \\
&\overset{(624)}{=} \left(A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T\right)^n \otimes I_M \\
&= \left[\left(A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T\right)\otimes I_M\right]^n \qquad (628)
\end{aligned}
$$

Iterating (613) we find that

$$
z_n = \left[\mathcal{A}^n\right]^T z_o \qquad (629)
$$

Proof

$$(\mathbb{1} \otimes I_M) w^o = (\mathbb{1} \otimes I_M)(1 \otimes w^o) = \mathbb{1} \otimes w^o$$

and, hence, from (616) and (619),

$$\begin{aligned} \widetilde{w}_n &= -\left[(\mathcal{A}^n)^T - \frac{1}{N} \cdot (\mathbb{1} \otimes I_M) \cdot (\mathbb{1}^T \otimes I_M) \right] \cdot z_o \\ &\overset{(628)}{=} -\left[\left(A^T - \frac{1}{N} \mathbb{1}\mathbb{1}^T \right) \otimes I_M \right]^n \cdot z_o \end{aligned} \tag{630}$$

Now recall that, for two arbitrary matrices C and D of compatible dimensions, the eigenvalues of the Kronecker product $C \otimes D$ is formed of all product combinations $\lambda_i(C)\lambda_j(D)$ of the eigenvalues of C and D [19]. We conclude from this property, and from the fact that $A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T$ is stable, that the coefficient matrix

$$\left(A^T - \frac{1}{N} \cdot \mathbb{1}\mathbb{1}^T \right) \otimes I_M$$

is also stable. Therefore,

$$\widetilde{w}_n \to 0 \quad \text{as } n \to \infty \tag{631}$$

Proof

(*Necessity*). In order for z_n in (629) to converge to $(\mathbb{1} \otimes I_M)w^o$, for any initial state z_o, it must hold that

$$\lim_{n\to\infty} (\mathcal{A}^n)^T \cdot z_o \;=\; \frac{1}{N} \cdot (\mathbb{1} \otimes I_M) \cdot (\mathbb{1}^T \otimes I_M) \cdot z_o \tag{632}$$

for any z_o. This implies that we must have

$$\lim_{n\to\infty} (\mathcal{A}^n)^T \;=\; \frac{1}{N} \cdot (\mathbb{1}\mathbb{1}^T \otimes I_M) \tag{633}$$

or, equivalently,

$$\lim_{n\to\infty} (A^n)^T \;=\; \frac{1}{N}\mathbb{1}\mathbb{1}^T \tag{634}$$

This in turn implies that we must have

$$\lim_{n\to\infty} A^T \cdot (A^n)^T \;=\; A^T \cdot \frac{1}{N}\mathbb{1}\mathbb{1}^T \tag{635}$$

But since

$$\lim_{n\to\infty} A^T \cdot (A^n)^T \;=\; \lim_{n\to\infty} \left(A^{n+1}\right)^T \;=\; \lim_{n\to\infty} (A^n)^T \tag{636}$$

Proof

we conclude from (634) and (635) that it must hold that

$$\frac{1}{N}\mathbb{1}\mathbb{1}^T \;=\; \frac{1}{N}A^T \cdot \mathbb{1}\mathbb{1}^T \tag{637}$$

That is,

$$\frac{1}{N}\left(A^T\mathbb{1} \;-\; \mathbb{1}\right)\cdot \mathbb{1}^T = 0 \tag{638}$$

from which we conclude that we must have $A^T\mathbb{1} = \mathbb{1}$. Similarly, we can show that $A\mathbb{1} = \mathbb{1}$ by studying the limit of $(A^n)^T A^T$. Therefore, A must be a doubly stochastic matrix. Now using the fact that A is doubly stochastic, we know that (624) holds. It follows that in order for condition (634) to be satisfied, we must have

$$\rho\left(A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T\right) < 1 \tag{639}$$

$\square$

Rate of Convergence

From (630) we conclude that the rate of convergence of the error vectors $\{\widetilde{w}_{k,n}\}$ to zero is determined by the spectrum of the matrix

$$A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T \tag{640}$$

Now since A is a doubly stochastic matrix, we know that it has an eigenvalue at $\lambda = 1$. Let us denote the eigenvalues of A by $\lambda_k(A)$ and let us order them in terms of their magnitudes as follows:

$$0 \leq \; |\lambda_M(A)| \; \leq \; \ldots \; \leq \; |\lambda_3(A)| \; \leq \; |\lambda_2(A)| \; \leq \; 1 \tag{641}$$

where $\lambda_1(A) = 1$. Then, the eigenvalues of the coefficient matrix $(A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T)$ are equal to

(since A is doubly-stochastic)

$$\{\; \lambda_M(A), , \ldots, \; \lambda_3(A), \; \lambda_2(A), 0 \;\} \tag{642}$$

It follows that the magnitude of $\lambda_2(A)$ becomes the spectral radius of $A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T$. Then condition (639) ensures that $|\lambda_2(A)| < 1$. We therefore arrive at the following conclusion.

Rate of Convergence

***Corollary* E.1.** (**Rate of Convergence of Consensus**) *Under conditions (620)–(622), the rate of convergence of the successive iterates* $\{w_{k,n}\}$ *towards the network average* w^o *in the consensus strategy (610) is determined by the second largest eigenvalue magnitude of* A*, i.e., by* $|\lambda_2(A)|$ *as defined in (641).*

$\square$

It is worth noting that doubly stochastic matrices A that are also primitive satisfy conditions (620)–(622). This is because, as we already know , the eigenvalues of such matrices satisfy $|\lambda_m(A)| < 1$, for $m = 2, 3, \ldots, N$, so that condition (622) is automatically satisfied.

Rate of Convergence

***Corollary* E.2.** (**Convergence for Primitive Combination Matrices**) *Any doubly-stochastic and* primitive *matrix A satisfies the three conditions (620)–(622) and, therefore, ensures the convergence of the consensus iterates $\{w_{k,n}\}$ generated by (610) towards w^o as $n \to \infty$.*

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.