[Ali H. Sayed, Sheng-Yuan Tu, Jianshu Chen, Xiaochuan Zhao, and Zaid J. Towfic]

Diffusion Strategies for Adaptation and Learning over Networks

© ISTOCKPHOTO.COM/JAMIE FARRANT

[An examination of distributed strategies and network behavior]

Nature provides splendid examples of real-time learning and adaptation behavior that emerges from highly localized interactions among agents of limited capabilities. For example, schools of fish are remarkably apt at configuring their topologies almost instantly in the face of danger [1]: when a predator arrives, the entire school opens up to let the predator through and then coalesces again into a moving body to continue its schooling behavior. Likewise, in bee swarms, only a small fraction of the agents (about 5%) are informed, and these informed agents are able to guide the entire swarm of bees to their new hive [2]. It is an extraordinary property of biological networks that sophisticated behavior is able to emerge from simple interactions among lower-level agents [3].

This article provides an overview of powerful diffusion strategies for adaptation and learning over networks that mimic some of these useful properties. The strategies rely on simple rules involving local adaptation and consultation and are able to deliver enhanced network performance. The presentation is in the context of adaptive networks, which consist of learning agents that are linked together through static or dynamic topologies. The agents interact with each other through in-network processing to solve estimation, inference, or optimization tasks in a fully distributed manner. The continuous sharing and diffusion of information across the network enables the agents to respond in real time to streaming data, to react to drifts in the statistical properties of the data, and to adjust the network topology when necessary. Such adaptive networks are

Digital Object Identifier 10.1109/MSP.2012.2231991
Date of publication: 5 April 2013

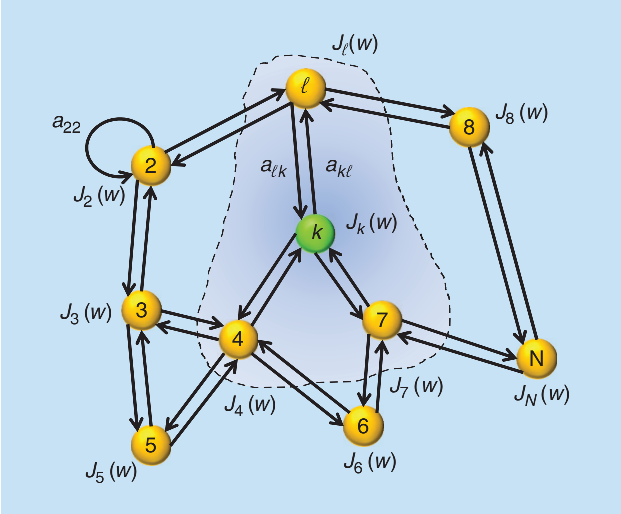
1053-5888/13/$31.00©2013IEEE

well suited to perform decentralized information processing tasks. They are also well suited to model forms of complex behavior exhibited by biological and social or economic networks [4]–[10].

The article explains some of the challenges for adaptation and learning over networks, describes strategies that can address these challenges, and explains how and when cooperation over networks outperforms noncooperative strategies. The article considers applications in distributed sensing (Example 1), intrusion detection (Example 4), target localization (Example 5), online machine learning (Example 6), fish schooling, and distributed optimization.

NETWORK MODEL

We focus on connected networks with N agents, as illustrated in Figure 1. Agents that are able to share information with each other are connected by edges. For emphasis in the figure, the edge between any two neighboring agents is being represented by two directed arrows to indicate that information flows both ways between the agents. The neighborhood of any particular agent, k, is denoted by $\mathcal{N}_k$ and it consists of all agents that are connected to k by edges; we include in this set agent k as well. We assume an undirected graph so that if agent k is a neighbor of agent ℓ, then agent ℓ is also a neighbor of agent k. We assign a pair of nonnegative weights $\{a_{k\ell}, a_{\ell k}\}$ to the edge connecting agents k and ℓ. The scalar $a_{\ell k}$ is used by agent k to scale the data it receives from agent ℓ; this scaling may be interpreted as a measure of trustworthiness that agent k assigns to its interaction with agent ℓ. Likewise, $a_{k\ell}$ is used by agent ℓ to scale the data it receives from agent k. The weights $\{a_{k\ell}, a_{\ell k}\}$ can be different, and one or both of them can be zero, so that the exchange of information over the edge linking k and ℓ need not be symmetric. When at least one a_{kk} is positive for some agent k, then we will say that the connected network is standard.



[FIG1] The neighborhood of agent k consists of a set of agents marked by the highlighted area and includes agents $\{\ell, 4, 7, k\}$.

We assume the network of N agents is interested in estimating in a distributed manner the parameter vector, w^o, of size $M \times 1$, that minimizes a global objective function of the following form:

$$\min_w \sum_{k=1}^{N} J_k(w) \qquad \text{(network objective)}, \qquad (1)$$

where a real-valued strongly convex cost $J_k(w)$ is associated with each agent k. These individual costs can be distinct across the agents or they can be identical. Although the algorithms described in this article apply to more general scenarios [11], [12], [87], we limit our exposition to the important case where the costs $\{J_k(w)\}$ $(k = 1, 2, \ldots, N)$ are minimized at the same w^o. This is a common situation in practice and is rich enough to help convey the main ideas, as illustrated by the various applications we consider in the context of distributed sensing, intrusion detection, machine learning, target localization, fish schooling, and distributed optimization. We can also consider constrained versions of (1) by incorporating convex constraints at each agent k. In [13], we explain how such constrained problems can be reduced to unconstrained problems of the form (1) by using barrier functions [14].

The objective of decentralized processing is to enable the agents to approach the solution of (1) by relying solely on local data, local cooperation with neighbors, and local in-network (as opposed to centralized) processing. Since the agents have a common objective (that of determining w^o), it may be natural to expect cooperation among them to be beneficial in general [we will see that this is not always true—see the discussion after expression (32)]. One important question is how to develop cooperation strategies that can lead to better performance than when each agent attempts to determine w^o on their own. Another important question is how to develop strategies that enable networks to adapt in real time to the continuous streaming of data. This article explains how diffusion strategies achieve these goals.

DATA MODEL

We treat initially the case in which the individual costs $\{J_k(w)\}$ in (1) correspond to mean-square-error (MSE) measures and are, therefore, quadratic in the unknown w. This case is of paramount importance in the context of estimation, adaptation, and learning over networks; it also helps illustrate the main issues underlying distributed strategies. Later, in the section "Distributed Optimization," we explain how the discussion extends to more general cost functions $J_k(w)$.

EXAMPLE 1 (DISTRIBUTED SENSING AND ESTIMATION)

Consider a situation in which N agents are interested in estimating the taps of some communications channel or the parameters of some physical model. Assume the agents are able to independently probe the unknown model and observe its response to excitations. Each agent k probes the model with an input sequence $\{u_k(i)\}$ and measures the response sequence, $\{d_k(i)\}$, in the presence of additive noise. The system dynamics for each agent k is assumed to be described by a moving-average model

$$d_k(i) = \sum_{m=0}^{M-1} \beta_m u_k(i-m) + v_k(i). \qquad (2)$$

If we collect the parameters $\{\beta_m\}$ into an $M \times 1$ vector $w^o = \mathrm{col}\{\beta_0, \beta_1, \ldots, \beta_{M-1}\}$, and the input data into a $1 \times M$ vector $\boldsymbol{u}_{k,i} = [\boldsymbol{u}_k(i) \ \boldsymbol{u}_k(i-1) \cdots \boldsymbol{u}_k(i-M+1)]$, then we can rewrite (2) as $\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i)$. Given streaming measurements $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ $(k = 1, 2, \ldots, N)$ over time $i \geq 0$, the agents would like to cooperate with each other to estimate w^o in a distributed manner by solving a problem of the form (1). Algorithms to achieve this goal are discussed in the sequel. ■

We start our exposition by assuming that each agent k measures realizations of a scalar random process $\boldsymbol{d}_k(i)$ and a $1 \times M$ random process $\boldsymbol{u}_{k,i}$ over $i \geq 0$. All vectors in our presentation are column vectors, with the exception of the regression vector, $\boldsymbol{u}_{k,i}$. It is assumed that $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ are related via a linear model of the following form:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i), \qquad k = 1, 2, \ldots, N. \qquad (3)$$

In this model, the vectors $\{\boldsymbol{u}_{k,i}\}$ represent general regression vectors that are not necessarily constructed as in Example 1 from past input data. In this overview article, and to avoid excessive technicalities, it is sufficient to assume the following conditions on the data.

ASSUMPTIONS A (MODEL CONDITIONS)

The data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$ are zero-mean jointly wide-sense stationary random processes satisfying model (3) and the following conditions:

(A.1) The regression data $\{\boldsymbol{u}_{k,i}\}$ are temporally white and independent over space with $\mathbb{E}\, \boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i} \triangleq R_u > 0$, where the symbol $*$ denotes complex conjugation for scalars and complex-conjugate transposition for matrices.

(A.2) The noise process $\{\boldsymbol{v}_k(i)\}$ is temporally white and independent over space with $\mathbb{E}\, \boldsymbol{v}_k(i) \boldsymbol{v}_k^*(i) \triangleq \sigma_{v,k}^2$.

(A.3) The regression and noise processes $\{\boldsymbol{u}_{\ell,j}, \boldsymbol{v}_k(i)\}$ are independent of each other for all k, ℓ, i, j.

(A.4) All agents employ the same step size μ in their adaptation mechanisms—see, e.g., (5), (14), and (16).

(A.5) The step size, μ, is sufficiently small such that terms that depend on higher powers of μ can be ignored. ■

Observe that we are allowing the noise power, $\sigma_{v,k}^2$, to vary with k. In this way, the quality of the measurements is allowed to vary across the network with some agents collecting noisier data than other agents. It is possible to extend many of the results in this exposition to more general conditions on the data than the ones stated above, such as allowing for space-dependent regression covariances, say, $\{R_{u,k}\}$ instead of R_u, and space-dependent step sizes, say, $\{\mu_k\}$ instead of μ [15], [16]. However, due to space limitations, it is sufficient to rely on the above assumptions; they allow us to quantify the improvement in performance that results from cooperation without biasing the results by differences in the adaptation mechanisms or in the statistical nature of the regression data at the agents.

The temporal whiteness condition (A.1) on the regression data $\{\boldsymbol{u}_{k,i}\}$ need not hold in general; for example, it does not hold when the regressors have shift structure (as happens in Example 1). However, there have been extensive studies in the literature in the context of (adaptive) stochastic gradient methods (e.g., [17] and [18]), showing that MSE results obtained under temporal whiteness conditions match well, to first order in μ, with actual performance for sufficiently small step sizes. More elaborate analyses are possible under weaker conditions [17–[20] but that is beyond the scope of this article.

NONCOOPERATIVE ADAPTIVE STRATEGIES

The objective of the agents is to estimate the unknown vector w^o of model (3) from the streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ $(k = 1, 2, \ldots, N, i \geq 0)$. Let us first examine the case in which the agents act independently of each other. If we multiply both sides of (3) by $\boldsymbol{u}_{k,i}^*$ and take expectations, we readily conclude that w^o can be determined from $w^o = R_u^{-1} r_{du}$, where $r_{du} \triangleq \mathbb{E} \boldsymbol{u}_{k,i}^* \boldsymbol{d}_k(i)$. It is useful to note that this expression for w^o is also the unique minimizer for the following MSE cost function

$$J_k(w) \triangleq \mathbb{E}\, |\, \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w\, |^2. \qquad (4)$$

The main difficulty in having each agent k determine w^o in this manner is that the statistical moments $\{r_{du}, R_u\}$ are rarely known beforehand. Instead, the agents have access to streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ that can be used to approach the minimizer of the above $J_k(w)$ by means of adaptive (or stochastic gradient) algorithms [20]–[22]. There are many adaptive algorithms that can be used for this purpose. It is sufficient for this overview presentation to consider one effective adaptive structure, while noting that the discussion can be extended to other more elaborate structures. One of the most elegant adaptive solutions is the least-mean-squares (LMS) filter [22]. In this solution, each agent k uses its data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ and computes successive estimators for w^o as follows:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu \boldsymbol{u}_{k,i}^* [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1}], \ i \geq 0, \qquad (5)$$

starting from some initial condition, say, $\boldsymbol{w}_{k,-1} = 0$. In (5), the notation $\boldsymbol{w}_{k,i}$ denotes the estimator for w^o at agent k at time i. The term multiplying μ on the right-hand side of (5) corresponds to an instantaneous approximation for the (negative of the complex-conjugate) gradient vector of $J_k(w)$ evaluated at $\boldsymbol{w}_{k,i-1}$ [20]; for later reference, we denote this approximation by $-[\widehat{\nabla_w J_k}(\boldsymbol{w}_{k,i-1})]^*$. If desired, the step size μ can be selected to vary with time; one popular choice is to use sequences $\mu(i)$ that satisfy the following two conditions simultaneously [19], [20]:

$$\sum_{i=0}^{\infty} \mu(i) = \infty, \qquad \sum_{i=0}^{\infty} \mu^2(i) < \infty. \qquad (6)$$

Such sequences converge slowly toward zero; one example is the choice $\mu(i) = \mu/(i+1)$. However, since these sequences die out as $i \to \infty$, then these choices turn off adaptation over time. For this reason, we focus mostly on the constant step size case throughout this article (with the exception of Example 6 on machine learning).

The performance of the LMS filter (5) is well understood. For example, it is known that, as time evolves, the weight estimators $\boldsymbol{w}_{k,i}$ approach w^o. The size of the weight-error vector, $\tilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i}$, is measured by the mean-square-deviation (MSD), which is defined as the steady-state variance value

$$\text{MSD}_k \triangleq \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \qquad (k = 1, 2, \ldots, N). \tag{7}$$

It is known that, for sufficiently small-step sizes [20]–[25] (see, e.g., [20, p. 362])

$$\text{MSD}_{\text{ncop},k} \approx \frac{\mu M}{2} \cdot \sigma_{v,k}^2, \tag{8}$$

where we added the subscript "ncop" to emphasize that this result is for the noncooperative mode of operation, where the agents run the LMS filter (5) individually. It is further known that, for sufficiently small step sizes, the convergence of $\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2$ toward its steady-state value occurs at the rate [20]–[22] (see, e.g., [20, eq. (24.20)]):

$$r \approx 1 - 2\mu \cdot \lambda_{\min}(R_u) \tag{9}$$

in terms of the smallest eigenvalue of R_u; the smaller the value of $r \in (0,1)$, the faster the convergence. Averaging (8) over all agents, we find that the performance of the noncooperative network is given by

$$\text{MSD}_{\text{ncop}}^{\text{network}} \approx \frac{\mu M}{2} \cdot \left(\frac{1}{N}\sum_{k=1}^{N} \sigma_{v,k}^2\right) \tag{10}$$

in terms of the average noise power. It is seen from (8) that the MSD performance across the agents varies in accordance with their noise level: agents with larger noise power have worse performance. However, since all agents are observing data arising from the same underlying model w^o according to (3), it is natural to expect cooperation among the agents to be beneficial. By cooperation we mean that neighboring agents can share information (such as data measurements or estimators). In the section "Diffusion Strategies," we describe strategies that enable agents to carry out adaptation and learning in a cooperative manner to enhance their MSD performance. The strategies will aim at optimizing the following aggregate MSE in a distributed manner, whose optimal solution is still given by the same $w^o = R_u^{-1} r_{du}$

$$\min_w \sum_{k=1}^{N} \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w|^2. \tag{11}$$

CENTRALIZED FUSION-BASED SOLUTION

In preparation for our description of cooperative solutions, we consider initially the problem of minimizing (11) in a centralized manner by using an (LMS) stochastic-gradient algorithm of the form

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu\left(\frac{1}{N}\sum_{k=1}^{N} \boldsymbol{u}_{k,i}^*[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{i-1}]\right). \tag{12}$$

At each time instant i, the central processor receives the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ from all agents and applies recursion (12) to update its estimator for w^o from $\boldsymbol{w}_{i-1}$ to $\boldsymbol{w}_i$. From the results in [26, eq. (39)] and [27, eq. (95)], it can be deduced that, under Assumptions (A), the MSD of (12) is well approximated by

$$\text{MSD}_{\text{cent}} \approx \frac{\mu M}{2} \cdot \frac{1}{N} \cdot \left(\frac{1}{N}\sum_{k=1}^{N} \sigma_{v,k}^2\right). \tag{13}$$

Comparing with (10), we observe an $N-$fold improvement in the MSD performance of the centralized solution relative to the noncooperative solution. Moreover, this improvement in performance is obtained without degradation in convergence rate; the convergence of $\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2$ continues to occur at the same rate as the noncooperative algorithm (5) [which is the reason why the factor $1/N$ is included in (12)]. One interesting property of the diffusion strategies discussed in the next section is that they are able to achieve the improved MSD performance level (13) of the centralized strategy, again at the same convergence rate, by relying solely on localized interactions and without the need for a fusion center—see (32) further ahead. Moreover, through proper selection of the combination weights used in the diffusion implementations, these distributed strategies can be made to outperform the centralized MSD level (13)—see expression (35) and the analysis following it. This statement may seem puzzling at first, but it follows from the fact that the diffusion implementation that leads to this result employs in (34) additional information about the noise variances that is not exploited by the centralized algorithm (12). One can of course modify (12) to include noise variance information as well, or use more powerful centralized solutions at the fusion center than (12) such as least-squares-based solutions or other more sophisticated estimation procedures. We continue with the standard centralized formulation (12) and compare centralized and distributed implementations that employ stochastic-gradient iterations from the same general family of algorithms.

DIFFUSION STRATEGIES

Diffusion strategies enable the solution of (11) in a distributed and adaptive manner. Compared to the noncooperative solution (5), these strategies introduce a useful aggregation step that helps incorporate into the adaptation mechanism information collected from the local neighborhoods. One such diffusion scheme is the combine-then-adapt (CTA) structure, which is described by the following update [28]:

$$\begin{cases} \boldsymbol{\psi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \\ \boldsymbol{w}_{k,i} \;\;= \boldsymbol{\psi}_{k,i-1} + \mu \boldsymbol{u}_{k,i}^*[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k,i-1}] \end{cases} \quad \text{(CTA diffusion)}. \tag{14}$$

The scalars $\{a_{\ell k}\}$ are convex combination coefficients that satisfy the conditions

$$a_{\ell k} \geq 0, \ \sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1, \text{and } a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \quad (k = 1, 2, \ldots, N). \tag{15}$$

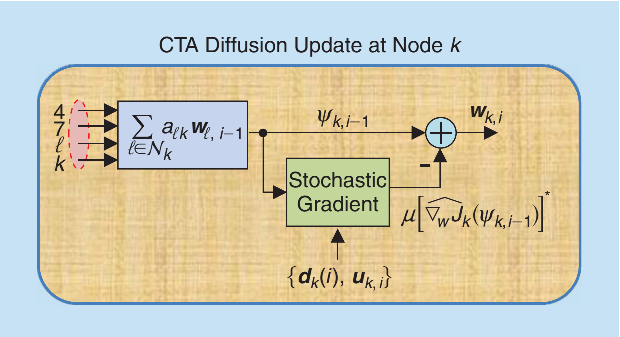
The coefficients $\{a_{\ell k}\}$ are free parameters that are chosen by the designer; their selection influences the performance of the algorithm (see the section "Mean-Square-Error Performance"). If we collect them into the $N \times N$ combination matrix $A \triangleq [a_{\ell k}]$, then condition (15) implies that the entries on each column of A add up to one, i.e., $A^{\top}\mathbb{1} = \mathbb{1}$, where "$\mathbb{1}$" denotes the $N \times 1$ vector with all entries equal to one. We say that A is a left-stochastic matrix. One useful property of left-stochastic matrices is that their spectral radii are equal to one, i.e., $\rho(A) = 1$ (so that the magnitude of any of the eigenvalues of A is bounded by one) [29], [30].

At every instant i, the CTA strategy (14) involves two operations (see Figure 2). The first operation is a combination step where agent k aggregates the estimators from its neighbors to obtain the intermediate estimator $\psi_{k,i-1}$. All other agents in the network are simultaneously performing a similar operation and aggregating the estimators of their neighbors. The second operation in (14) is an adaptation step where agent k uses its data $\{d_k(i), u_{k,i}\}$ to update its intermediate estimator to $w_{k,i}$. Again, all other agents in the network are simultaneously performing a similar operation. The reason for the qualification "diffusion" is that the intermediate state $\psi_{k,i-1}$ in (14) allows information to diffuse through the network by bringing into location k the effect of data beyond the neighborhood of k.

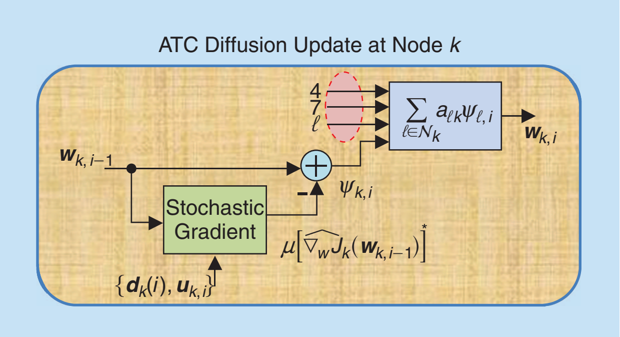
An alternative form of the diffusion strategy (14) can be obtained by switching the order of the combination and adaptation steps to get the following adapt-then-combine (ATC) diffusion strategy [16]:

$$\begin{cases} \psi_{k,i} = w_{k,i-1} + \mu u_{k,i}^{*}[d_k(i) - u_{k,i} w_{k,i-1}] \\ w_{k,i} = \displaystyle\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_{\ell,i} \end{cases} \quad \text{(ATC diffusion).} \tag{16}$$

In this implementation, the first operation is the adaptation step (see Figure 3). The structure of the CTA and ATC strategies (14) and (16) are fundamentally the same: the difference lies in which variable we choose to correspond to the updated weight estimator $w_{k,i}$. In ATC, we choose the result of the combination step to be $w_{k,i}$, whereas in CTA we choose the result of the adaptation step to be $w_{k,i}$. More general diffusion strategies are possible by allowing for enlarged exchange of information among the agents, such as exchanging neighborhood data $\{d_\ell(i), u_{\ell,i}\}$ in addition to the estimators $\{\psi_{\ell,\cdot}\}$. These generalizations appear in [15], [16], and [31]; in addition to the matrix A, they employ a second combination matric C that is required to be right-stochastic (i.e., it has nonnegative entries and $C\mathbb{1} = \mathbb{1}$). The forms (14) and (16) correspond to the choice $C = I$.


CTA Diffusion Update at Node k
4
7
ℓ
k
$\sum_{\ell \in N_k} a_{\ell k} \mathbf{w}_{\ell, i-1}$
$\psi_{k,i-1}$
$\mathbf{w}_{k,i}$
–
Stochastic Gradient
$\mu[\widehat{\nabla_w J_k}(\psi_{k,i-1})]^*$
$\{\mathbf{d}_k(i), \mathbf{u}_{k,i}\}$

[FIG2] The diagram illustrates the CTA strategy (14) for agent k, which uses information from its neighbors $\{4, 7, \ell, k\}$ in the network of Figure 1.


ATC Diffusion Update at Node k
4
7
ℓ
$\sum_{\ell \in N_k} a_{\ell k} \psi_{\ell,i}$
$\mathbf{w}_{k,i}$
$\mathbf{w}_{k,i-1}$
–
$\psi_{k,i}$
Stochastic Gradient
$\mu[\widehat{\nabla_w J_k}(\mathbf{w}_{k,i-1})]^*$
$\{\mathbf{d}_k(i), \mathbf{u}_{k,i}\}$

[FIG3] The diagram illustrates the ATC strategy (16) for agent k, which uses information from its neighbors $\{4, 7, \ell, k\}$ in the network of Figure 1.

The CTA diffusion strategy (14) was proposed in [32]–[34] and [28], and the ATC diffusion structure (16), with adaptation preceding combination, was proposed in the work [35] on distributed least-squares schemes and subsequently in the works [36]–[38] and [16] on distributed MSE and state-space estimation methods. The main motivation for the introduction of these diffusion strategies was the desire to develop distributed schemes that are able to respond in real time to continuous streaming of data at the agents by operating over a single time-scale. The CTA structure of Figure 2, with the additional requirement that the step size μ is a time-dependent sequence $\mu(i)$ that decays toward zero with time as in (6), was later employed by [39]–[41] to solve distributed optimization problems that require all agents to reach agreement—see the section "Distributed Optimization." The ATC form (16), again with a time-dependent sequence $\mu(i)$ that decays toward zero with time, was also employed by [42] to ensure agreement.

We end this section by noting that there is another class of distributed strategies that is based on consensus-type implementations. The traditional consensus solution operates over two separate time-scales: one for collecting data and one for iterating over the data (e.g., [43]–[47]). Such two time-scale implementations hinder adaptation; as mentioned above, diffusion strategies resolve this difficulty by relying on single time-scale iterations that process the data in real time as they arrive at the agents. Motivated by a procedure for distributed optimization from [48, eq. (7.1)], some works subsequently proposed single time-scale implementations for consensus strategies as well. The consensus implementation for solving (11) takes the following form if we set the step size to a constant value to enable continuous adaptation and learning; see, e.g., [46] and [49]–[51]

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} + \mu \boldsymbol{u}^*_{k,i}[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \quad \text{(consensus)}. \tag{17}$$

It can be easily verified that the diffusion strategies (14) and (16) have exactly the same computational complexity as the above consensus strategy in terms of the number of additions and multiplications required per iteration to move from time $i-1$ to time i. However, diffusion strategies differ in an important and interesting way from the consensus implementation. For example, by comparing the diffusion implementations (14) and (16) with the consensus implementation (17), we observe that diffusion evaluates first an intermediate state variable $\psi_{k,}$ and then uses it in the subsequent step. The net effect for diffusion are updates of the form

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}(\boldsymbol{w}_{\ell,i-1} + \mu \boldsymbol{u}^*_{\ell,i}[\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}\boldsymbol{w}_{\ell,i-1}]) \quad \text{(ATC)} \tag{18}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} + \mu \boldsymbol{u}^*_{k,i}\left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\boldsymbol{w}_{\ell,i-1}\right] \quad \text{(CTA)}. \tag{19}$$

Note, for instance, that in the CTA case, the convex combination of the neighborhood weights appears inside the rightmost error term in (19). In contrast, the consensus algorithm only uses $\boldsymbol{w}_{k,i-1}$ to evaluate the error term in (17). This asymmetry in the consensus update is responsible for an anomaly in its behavior. As the discussion in the next section reveals (see Examples 2 and 3), some care is needed when using the consensus strategy (17) for adaptation; the algorithm can fail even when all individual agents are able to solve the inference task on their own in a stable manner. This phenomenon does not occur for diffusion strategies: stability of the individual agents ensures stability of the diffusion network irrespective of the combination topology. Diffusion strategies lead to enhanced stability and provide lower MSD and faster convergence rate than the consensus strategy.

MEAN-SQUARE-ERROR PERFORMANCE

We now examine under what conditions network cooperation leads to improved MSD performance in comparison to noncooperation. It turns out that while the average performance over the network is improved relative to noncooperation, the performance of each individual agent does not necessarily improve unless the combination weights are chosen properly. To reveal these properties, we carry out the MSE analysis of the consensus and diffusion strategies in a unified manner. We again introduce the weight error vectors $\tilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i}$, and define the MSD of each agent as in (7). The network performance is defined as the average MSD level

$$\text{MSD}^{\text{network}} \triangleq \frac{1}{N}\sum_{k=1}^{N} \text{MSD}_k. \tag{20}$$

NETWORK ERROR DYNAMICS

We collect the weight-error vectors from across all agents into the block vector $\tilde{\boldsymbol{w}}_{k,i} \triangleq \text{col}\{\tilde{\boldsymbol{w}}_1, \tilde{\boldsymbol{w}}_2, \ldots, \tilde{\boldsymbol{w}}_{N,i}\}$. We further let Σ denote an arbitrary $N \times N$ block Hermitian nonnegative-definite matrix that we are free to choose, with $M \times M$ block entries. Different choices for Σ enable us to extract different types of information about the performance of the agents and the network [15], [16]. Subtracting w^o from both sides of the diffusion recursions (14) and (16), as well as the consensus recursion (17) and the noncooperative recursion (5), and using model (3), we can establish the following mean and MSE relations under Assumptions (A) over $i \geq 0$:

$$\mathbb{E}\tilde{\boldsymbol{w}}_i = \mathcal{B} \cdot \mathbb{E}\tilde{\boldsymbol{w}}_{i-1} \tag{21}$$

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_\Sigma = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{B}^*\Sigma\mathcal{B}} + \text{Tr}(\Sigma\mathcal{Y}), \tag{22}$$

where the notation $\|x\|^2_\Sigma$ denotes the weighted square quantity $x^*\Sigma x$, and the quantities $\{\mathcal{B}, \mathcal{Y}\}$ are given by the following expressions for the various algorithms (in terms of the Kronecker product operation $\otimes$):

$$\begin{cases} \mathcal{B}_{\text{atc}} = A^\top \otimes (I_M - \mu R_u), & \mathcal{Y}_{\text{atc}} = \mu^2(A^\top R_v A \otimes R_u) \\ \mathcal{B}_{\text{cta}} = A^\top \otimes (I_M - \mu R_u), & \mathcal{Y}_{\text{cta}} = \mu^2(R_v \otimes R_u) \\ \mathcal{B}_{\text{cons}} = A^\top \otimes I_M - \mu(I_N \otimes R_u), & \mathcal{Y}_{\text{cons}} = \mu^2(R_v \otimes R_u) \\ \mathcal{B}_{\text{ncop}} = I_N \otimes (I_M - \mu R_u), & \mathcal{Y}_{\text{ncop}} = \mu^2(R_v \otimes R_u), \end{cases} \tag{23}$$

where $R_v \triangleq \text{diag}\{\sigma^2_{v,1}, \ldots, \sigma^2_{v,N}\}$. Observe that $\mathcal{B}_{\text{atc}} = \mathcal{B}_{\text{cta}}$, so we denote this coefficient matrix by $\mathcal{B}_{\text{diff}}$ for the diffusion strategies. Furthermore, the MN eigenvalues of the above matrices $\mathcal{B}$ are given by the following expressions in terms of the eigenvalues of $\{A, R_u\}$ for $k = 1, 2, \ldots, N$ and $m = 1, 2, \ldots, M$:

$$\begin{aligned} \lambda(\mathcal{B}_{\text{diff}}) &= \lambda_k(A) \cdot [1 - \mu\lambda_m(R_u)], \\ \lambda(\mathcal{B}_{\text{cons}}) &= \lambda_k(A) - \mu\lambda_m(R_u), \\ \lambda(\mathcal{B}_{\text{ncop}}) &= 1 - \mu\lambda_m(R_u). \end{aligned} \tag{24}$$

CONVERGENCE IN THE MEAN

Since A is left-stochastic and, therefore, $\rho(A) = 1$, we readily conclude from (21) and (24) that the estimators $\{\boldsymbol{w}_{k,i}\}$ resulting from the diffusion strategies (14) and (16) will be asymptotically unbiased (i.e., $\mathbb{E}\tilde{\boldsymbol{w}}_{k,i} \to 0$ as $i \to \infty$) if the step size parameter μ is sufficiently small and satisfies

$$\mu < 2/\lambda_{\max}(R_u) \qquad \text{(for diffusion strategies)} \tag{25}$$

in terms of the largest eigenvalue of R_u. Observe the interesting fact that condition (25) for the mean stability of diffusion networks does not depend on the combination matrix A. Moreover, result (25) is the same condition that is required for the mean stability of the individual LMS filters (5), as can be easily seen from the expression for the eigenvalues of $\mathcal{B}_{\text{ncop}}$ in (24). Therefore, if the individual filters (or agents) are stable in the mean, then the diffusion network will also be stable in the mean regardless of the choice of A and the corresponding topology. This useful conclusion does not hold for consensus networks. As can be seen from the eigenstructure of $\mathcal{B}_{\text{cons}}$ in (24), even if all individual agents are stable, with μ satisfying (25), the spectral radius of $\mathcal{B}_{\text{cons}}$ can still be larger than one depending on A and the network topology [52], [53]. The following example illustrates this anomaly.

EXAMPLE 2 (DIFFUSION ENHANCES STABILITY)

Consider a network consisting of two cooperating agents, as shown in Figure 4; this case is sufficient to illustrate the aforementioned problem. For simplicity, we assume the weight vector w^o is a scalar and $R_u = \sigma_u^2$ (i.e., $N = 2, M = 1$). Assume Agent 1 uses combination weights $\{1-a, a\}$, while Agent 2 uses combination weights $\{b, 1-b\}$ with $a, b \in [0,1]$. Then, using (23), we get

$$\mathcal{B}_{\text{diff}} = (1-\mu\sigma_u^2)\cdot\begin{bmatrix}1-a & a\\ b & 1-b\end{bmatrix},$$

$$\mathcal{B}_{\text{cons}} = \begin{bmatrix}1-a-\mu\sigma_u^2 & a\\ b & 1-b-\mu\sigma_u^2\end{bmatrix}. \qquad (26)$$

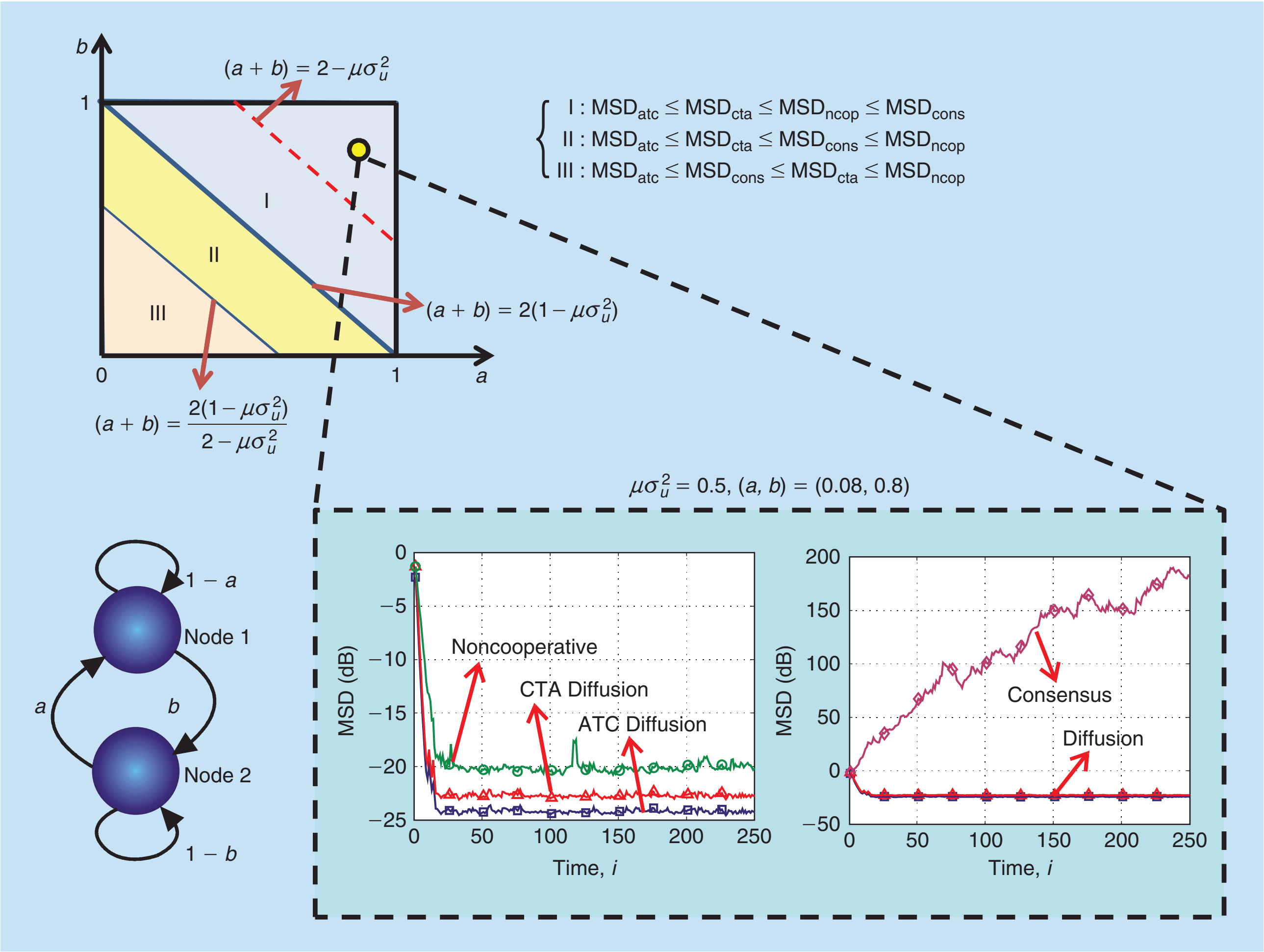
We assume $\mu\sigma_u^2 < 2$ so that both individual agents are mean stable. Then, by (25), the diffusion network will also be stable in the mean for any choice of $\{a, b\}$. However, there are choices for $\{a, b\}$ that cause the consensus network to become unstable even though the individual agents are stable. Indeed, note that $\lambda_{\min}(\mathcal{B}_{\text{cons}}) = 1 - a - b - \mu\sigma_u^2$, which becomes smaller than -1 (and, hence, unstable) for choices $\{a, b\}$ that satisfy $a + b > 2 - \mu\sigma_u^2$. More general scenarios are studied in [52] and [53]. ■

The above example illustrates that the order and the manner by which information is processed over a network (e.g., consensus versus diffusion) is critical. This observation is in line with studies in the social sciences in relation to the wisdom of groups where, paraphrasing, it is generally remarked that it is important to deliver the "right information in the right way at the right time and place" to the agents [54].

CONVERGENCE RATE

Assume now that the diffusion and consensus networks are stable in the mean [i.e., their matrices $\mathcal{B}$ in (21) are stable]. Iterating (22) and taking the limit as $i \to \infty$, we conclude that

$$\lim_{i\to\infty} \mathbb{E}\|\tilde{w}_i\|_\Sigma^2 = \sum_{j=0}^{\infty} \text{Tr}(\mathcal{B}^j \mathcal{Y} \mathcal{B}^{*j} \Sigma). \qquad (27)$$



[FIG4] Comparison of network MSDs. The consensus strategy is unstable when (a, b) lie above the dashed line in region I (see Example 2). The simulation employs $\mu\sigma_u^2 = 0.5$ and $(a, b) = (0.8, 0.8)$.

The convergence rate of the series is governed by $[\rho(\mathcal{B})]^2$, in terms of the spectral radius of $\mathcal{B}$. Using $\rho(A) = 1$, it can be verified from comparing the eigenvalue expressions in (24) that the convergence rate of diffusion networks is still given by (9), while the convergence rate of consensus networks is slower since $\rho(\mathcal{B}_{\text{diff}}) \le \rho(\mathcal{B}_{\text{cons}})$ [52,] [53]. Therefore, diffusion strategies do not only enhance stability, but they also improve the convergence rate.

MSD PERFORMANCE

By selecting $\Sigma = I_{MN}/N$ or $\Sigma = e_k \otimes I_M \equiv \mathcal{J}_k$ in (27), where e_k is the kth basis vector of size $N \times 1$ with its kth entry equal to one, we arrive at the following expressions for the MSD performance of the network and its individual agents for sufficiently small step sizes:

$$\text{MSD}^{\text{network}} \approx \frac{1}{N}\sum_{j=0}^{\infty} \text{Tr}(\mathcal{B}^j \mathcal{Y} \mathcal{B}^{*j}), \quad \text{MSD}_k \approx \sum_{j=0}^{\infty} \text{Tr}(\mathcal{B}^j \mathcal{Y} \mathcal{B}^{*j} \mathcal{J}_k). \tag{28}$$

When the combination matrix A is symmetric or close-to-symmetric (i.e., diagonalizable with left-eigenvectors that are practically orthogonal to each other), the above expressions can be used to establish that diffusion networks lead to better MSD performance (i.e., smaller MSD values) than consensus networks. With some effort, the following result can be deduced from the above expressions [52], [53].

THEOREM 1 (COMPARING MSD PERFORMANCE)

Assume A is symmetric or close-to-symmetric. Then, the ATC diffusion strategy achieves the lowest network MSD

$$\begin{aligned} & \text{MSD}_{\text{atc}}^{\text{network}} \le \text{MSD}_{\text{cta}}^{\text{network}} \le \text{MSD}_{\text{ncop}}^{\text{network}} \\ \text{and} \quad & \text{MSD}_{\text{atc}}^{\text{network}} \le \text{MSD}_{\text{cons}}^{\text{network}}. \end{aligned} \tag{29}$$

Proof

See [52] and [53]. The argument involves introducing the eigendecompositions of A and R_u into (28) and comparing the resulting expressions for the various strategies. ■

EXAMPLE 3 (DIFFUSION ENHANCES PERFORMANCE)

We reconsider the two-agent network from Example 2 (with $N = 2, M = 1$) and assume $0 < \mu\sigma_u^2 < 1$ so that both agents are stable in the mean. Furthermore, to ensure the mean stability of the consensus strategy, the parameters $(a, b) \in [0, 1]$ are assumed to satisfy $a + b < 2 - \mu\sigma_u^2$. The eigenvalues of A in this case are at $\lambda_1(A) = 1$ and $\lambda_2(A) = 1 - a - b$. After some algebra [52], [53], expression (28) allows us to partition the $[0, 1] \times [0, 1]$ square into the three regions shown in Figure 4. The ATC diffusion strategy performs the best in all regions, while the performance of consensus relative to the other strategies is dependent on the region; consensus becomes unstable in the area above the dotted line. ■

Given the superior performance of diffusion networks over consensus networks, we continue our presentation by focusing on diffusion strategies. We can simplify the MSD expressions (28) for diffusion strategies into a useful and revealing form for standard networks. For such networks, the left-stochastic combination matrix A will be what is called a primitive matrix [15], [29], [30], which in turn implies that A will have a unique eigenvalue at one while all other eigenvalues will have magnitude strictly less than one. Let p denote the right eigenvector of A that is associated with the eigenvalue at one and whose entries are normalized to add up to one. It follows from the Perron-Frobenius theorem [29] that the entries of p are positive and smaller than one; we denote them by $\{p_k, k = 1, 2, \ldots, N\}$:

$$Ap = p, \quad p^{\top}\mathbb{1} = 1, \quad 0 < p_k < 1. \tag{30}$$

THEOREM 2 (STANDARD DIFFUSION NETWORKS)

For standard diffusion networks, the performance of each individual agent is approximately equal to the network MSD and they are both well approximated by

$$\text{MSD}_{\text{diff},k} \approx \text{MSD}_{\text{diff}}^{\text{network}} = \frac{\mu M}{2} \cdot \left(\sum_{k=1}^{N} p_k^2 \sigma_{v,k}^2\right) + O(\mu^2) \quad (k = 1, 2, \ldots, N). \tag{31}$$

Proof

See [27]. The argument involves introducing the Jordan canonical decomposition of A and the eigendecomposition of R_u into (28), and then exploiting the fact that $(N - 1)$ eigenvalues of A have magnitude strictly less than one. ■

Comparing (31) with (13) in the centralized case, we observe that the effect of diffusion cooperation is to scale the noise variances by the factors $\{p_k^2\}$ instead of $1/N^2$; these factors are determined by the combination policy A. Note further that, for sufficiently small step sizes, diffusion strategies equalize the MSD performance across the agents (even though some agents may have more noise than other agents). This result is not inconsistent with the fact that ATC outperforms CTA, as revealed by (29). This is because the fine difference in performance between these two strategies arises at higher-order terms in μ, which are incorporated into $O(\mu^2)$ in (31) [27].

DO ALL AGENTS BENEFIT FROM COOPERATION?

We now consider the interesting question whether network cooperation is beneficial to all agents. We examine two questions:

1) How much improvement in network MSD is attained through cooperation?

2) Does the MSD performance of each individual agent in the diffusion network improve relative to the noncooperative case?

We will see that the answer to inquiry 1) is that the MSD level improves by N–fold (i.e., it becomes N times smaller) for the diffusion network compared to noncooperation. We will also see that the answer to inquiry 2) is negative in general unless the combination policy A is chosen properly!

DOUBLY STOCHASTIC COMBINATION MATRICES

Consider first the case in which A is doubly stochastic for a standard diffusion network; a doubly stochastic matrix has nonnegative entries and the entries on each of its rows and columns add up to one so that $A^{\top}\mathbb{1} = A\mathbb{1} = \mathbb{1}$. Then, the right-eigenvector p defined by (30) is given by $p = \mathbb{1}/N$ and (31) reduces to

$$\mathrm{MSD}_{\mathrm{diff},k} \approx \mathrm{MSD}_{\mathrm{diff}}^{\mathrm{network}} \approx \frac{\mu M}{2} \cdot \frac{1}{N} \cdot \left(\frac{1}{N} \sum_{k=1}^{N} \sigma_{v,k}^2 \right). \qquad (32)$$

Comparing with (10), we conclude that $\mathrm{MSD}_{\mathrm{diff}}^{\mathrm{network}} = 1/N \cdot \mathrm{MSD}_{\mathrm{ncop}}^{\mathrm{network}}$, which confirms that diffusion networks attain the MSD level (13) of the centralized solution and they both outperform the noncooperative strategy by a factor of N. But how does the performance of the individual agents compare in the diffusion and noncooperative strategies? From (8) we observe that if the noise variance is uniform across all agents, i.e., $\sigma_{v,k}^2 = \sigma_v^2$ $(k = 1, 2, \ldots, N)$, then the MSD of the individual diffusion agents will also be smaller by the same factor N than their noncooperative performance. However, when the noise profile varies across the agents, the performance of the individual diffusion and noncooperative agents cannot be compared directly and one can be larger than the other depending on the noise profile. For example, from (8) and (32), for the performance of agent k to improve over its noncooperative behavior, it must hold that $1/N \sum_{k=1}^{N} \sigma_{v,k}^2 < N\sigma_{v,k}^2$. For example, for $N = 2$, $\sigma_{v,1}^2 = 1$ and $\sigma_{v,2}^2 = 9$, Agent 1 will not benefit from cooperation while Agent 2 will.

LEFT-STOCHASTIC COMBINATION MATRICES

The next question is whether it is possible to select combination matrices A that will ensure that diffusion networks will outperform noncooperative strategies both in terms of the overall network performance and the individual agent performance, even when the noise variances are different across the agents. It turns out that we can construct left-stochastic matrices A that achieve this goal by minimizing the network MSD given by (31)

$$A^o \triangleq \arg\min_{A \in \mathbb{A}} \sum_{k=1}^{N} p_k^2 \sigma_{v,k}^2 \qquad \text{subject to (30)}, \qquad (33)$$

where $\mathbb{A}$ denotes the set of all $N \times N$ primitive left-stochastic matrices whose entries $\{a_{\ell k}\}$ satisfy conditions (15). Using a construction procedure developed in [55] and [56], it was argued in [27] that one choice for A^o is the following left-stochastic matrix, which we refer to as the Hastings rule:

$$a_{\ell k}^o = \begin{cases} \dfrac{\sigma_{v,k}^2}{\max\{n_k \sigma_{v,k}^2, n_\ell \sigma_{v,\ell}^2\}}, & \ell \in \mathcal{N}_k \backslash \{k\} \\ 1 - \sum_{m \in \mathcal{N}_k \backslash \{k\}} a_{mk}^0, & \ell = k \end{cases} \qquad \text{(Hastings rule)}, \qquad (34)$$

where n_k denotes the cardinality of $\mathcal{N}_k$ (also called the degree of agent k and is equal to the number of its neighbors). The resulting minimum (optimal) MSD value is

$$\mathrm{MSD}_{\mathrm{diff,opt}}^{\mathrm{network}} \approx \mathrm{MSD}_{\mathrm{diff,opt},k} \approx \frac{\mu M}{2} \cdot \frac{1}{\sum_{k=1}^{N} \sigma_{v,k}^{-2}}. \qquad (35)$$

It can be easily verified that the above expression satisfies $\mathrm{MSD}_{\mathrm{diff,opt}}^{\mathrm{network}} \le 1/N \cdot \mathrm{MSD}_{\mathrm{ncop}}^{\mathrm{network}}$, so that the MSD of the diffusion network with the optimal left-stochastic A is at least N–fold smaller than the noncooperative scenario. Moreover, and importantly, comparing (35) with (8), it is clear that $\mathrm{MSD}_{\mathrm{diff,opt,k}} \le \mathrm{MSD}_{\mathrm{ncop,k}}$, so that the individual agent performance in the optimized diffusion network is also improved across all agents relative to the noncooperative case.

ADAPTIVE COMBINATION WEIGHTS

Several other combination policies $\{a_{\ell k}\}$ have been proposed in the literature for combining information over graphs, such as the averaging rule, Metropolis rule, Laplacian rule, maximum-degree rule, and relative-degree rule; see, e.g., [15] and [16]. However, in all these constructions, the expressions for the combination weights are defined solely in terms of the degrees of the agents and their neighbors. Such selections can be limiting for adaptation over networks because they ignore the noise profile across the agents. It is important to design combination rules that enable agents to give more or less weight to their neighbors depending on how noisy their information is.

The Hastings rule (34) is one example of a combination policy that takes into account the noise level at the agents. In this rule, the interaction between agents k and ℓ is dependent on the noise levels at these locations alone. Another construction is derived in [57], where the interaction between agents k and ℓ depends on the noise profile across the entire neighborhood of agent k. In this construction, neighbors with smaller relative noise power are assigned larger weights

$$a_{\ell k}^o = \frac{\sigma_{v,\ell}^{-2}}{\sum_{m \in \mathcal{N}_k} \sigma_{v,m}^{-2}}, \quad \ell \in \mathcal{N}_k \quad \text{(relative-variance rule)}. \qquad (36)$$

The noise variances that are needed in (34) or (36) are generally unavailable. One way for agent k to estimate the noise variances of its neighbors is to use the following filter in an ATC implementation—a similar filter can be used for CTA [15], [27], [57]

$$\gamma_{\ell k}^2(i) = (1-\nu)\, \gamma_{\ell k}^2(i-1) + \nu \| \psi_{\ell,i} - w_{k,i-1} \|^2 \quad (\ell \in \mathcal{N}_k), \quad (37)$$

where $\nu \in (0,1)$ is a small positive coefficient, e.g., $\nu = 0.1$. It can be verified that, as $i \to \infty$ and under Assumptions (A), the expected value of $\gamma_{\ell k}^2(i)$ approaches $\mu^2 \sigma_{v,\ell}^2 \,\mathrm{Tr}(R_u)$ and is therefore proportional to $\sigma_{v,\ell}^2$. The variables $\{\gamma_{\ell k}^2(i)\}$ can then be used by agent k to adapt the combination weights in (34) or (36) over time.

EXAMPLE 4 (DETECTING INTRUDERS AND CLUSTERING)

Allowing diffusion networks to adjust their combination coefficients in real time enables the agents to assign smaller or larger weights to their neighbors depending on how well they contribute to the inference task. This capability can be exploited by the network to exclude harmful neighbors (such as intruders) [58]. For example, the ATC diffusion strategy

(16) with adaptive combination weights would take the following form:

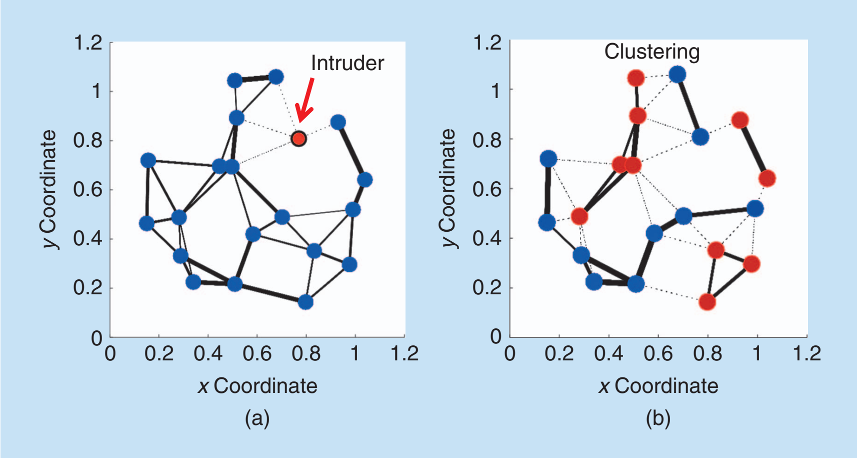
```
ATC diffusion strategy with adaptive combination weights.
  set  $\gamma^2_{\ell k}(-1) = 0$ for all $k = 1, 2, \ldots, N$ and $\ell \in N_k$.
  for $i \geq 0$ and for every agent $k$ do:
      $e_k(i) = d_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}$
      $\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu \boldsymbol{u}_{k,i} e_k(i)$
      $\gamma^2_{\ell k}(i) = (1-\nu)\,\gamma^2_{\ell k}(i-1) + \nu \| \boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1} \|^2, \ \ell \in \mathcal{N}_k$      (38)
      $a_{\ell k}(i) = \dfrac{\gamma^{-2}_{\ell k}(i)}{\sum_{m \in \mathcal{N}_k} \gamma^{-2}_{mk}(i)}, \ \ \ell \in \mathcal{N}_k$
      $\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}(i)\, \boldsymbol{\psi}_{\ell,i}$
  end
```

Figure 5 illustrates the ability of networks running algorithm (38) to detect intrusion, and also to perform agent clustering. The figure shows a network with $N = 20$. One of the agents, say, agent ℓ_o, is an intruder and it feeds its neighbors irrelevant data such as sending them wrong estimators $\psi_{\ell_o,i}$. In some other applications, agent ℓ_o may not be an intruder but is simply subject to measurements $\{\boldsymbol{d}_{\ell_o}, \boldsymbol{u}_{\ell_o,i}\}$ that arise from a different model $w^\star$ than the model w^o. Figure 5(a) shows the state of the combination weights after 300 diffusion iterations: the thickness of the edges reflect the size of the combination weights assigned to them; thicker edges correspond to larger weights. Observe how the edges connecting to the intruder are essentially cutoff by the algorithm. Figure 5(b) illustrates the ability of diffusion strategies to perform agent clustering (i.e., to cluster together agents that are influenced by the same model). Agents do not know beforehand which of their neighbors are influenced by which model. They also do not know which model is influencing their own data. By allowing agents to adapt their combination coefficients, it becomes possible for the agents to cut their links over time to neighbors that are sensing a different model than their own. The net effect is that agents end up being clustered in two groups; the blue agents belong to one group and the red agents belong to a second group. Cooperation between the members of the same group then leads to proper estimation of $\{w^\star, w^o\}$. ■

IS MORE INFORMATION ALWAYS BENEFICIAL FOR COOPERATION?

We assumed in our presentation so far that the agents are homogeneous in that they all have similar processing capabilities and are able to have continuous access to data measurements. However, it is generally observed in nature that the behavior of biological networks is often driven more heavily by a small fraction of informed agents as happens, for example, with bees and fish [2]. This phenomenon motivates us to examine in this section diffusion networks where only a fraction of the agents are informed, while the remaining agents are uninformed. Informed agents collect data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ continuously and perform in-network processing tasks (consultation and adaptation), while uninformed agents only participate in the consultation tasks (they do not perform adaptation because they do not receive data). The results below reveal some interesting facts [59], [60]. When the set of informed agents is enlarged, the convergence rate of the network becomes faster albeit at the expense of some possible deterioration in mean-square performance. In other words, the MSD performance of the network does not necessarily improve with a larger proportion of informed agents. These results are in line with observations in the social sciences in relation to how information propagates over social networks such as the assertion that "too much information and communication can make a network of agents less intelligent" [54]. The results are also in line with the popular saying that "too many cooks spoil the broth."

We illustrate these phenomena for standard ATC diffusion strategies of the form (16). We model uninformed agents by setting their step size to zero, i.e., we now use an agent-dependent step size and set $\mu_k = \mu$ for informed agents and $\mu_k = 0$ for uninformed agents. In this way, uninformed agents do not perform the adaptation step in (16) but continue to perform the aggregation step. To facilitate the presentation, we assume, without loss of generality, that the informed agents are labeled $\mathcal{N}_I = \{1, 2, \ldots, N_I\}$, while the remaining uninformed agents are labeled $\{N_{I+1}, \ldots, N\}$. We assume the network has at least one informed agent so that $N_I \geq 1$. We can now repeat the MSE analysis of the section "Mean-Square-Error Performance." Since we are dealing with standard networks, then there always exists a path of finite length from an informed agent to every other agent in the network. As a result, it can be shown that, as long as the informed agents employ small step sizes μ that satisfy (25), then the estimators $\boldsymbol{w}_{k,i}$ will continue to be asymptotically unbiased across *all* agents (both informed and uninformed).



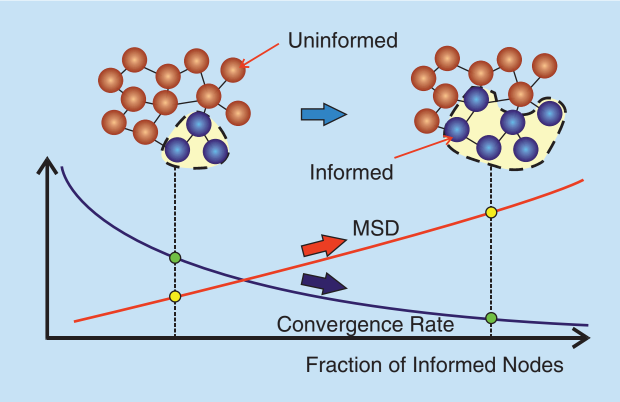
[FIG5] **(a) Illustration of how diffusion cuts the links to the intruder. (b) Illustration of the clustering ability of the network.**

Moreover, for sufficiently small step sizes, the convergence rate and the MSD level are now given by [59] and [60]:

$$r \approx 1 - 2\mu\lambda_{\min}(R_u)\cdot\left(\sum_{k=1}^{N_I} p_k\right) \qquad (39)$$

$$\text{MSD}_{\text{diff}}^{\text{network}} \approx \frac{\mu M}{2}\cdot\left(\frac{1}{\sum_{k=1}^{N_I} p_k}\right)\cdot\sum_{k=1}^{N_I} p_k^2\sigma_{v,k}^2 \qquad (40)$$

in terms of the entries $\{p_k\}$ of the eigenvector p defined by (30). Note that the above MSD expression reduces to (31) when $N_I = N$ (i.e., all agents are informed) since, by definition, the entries of p add up to one. Note further that since the entries of p are positive for primitive left-stochastic matrices A, it is clear from (39) that if the set of informed agents is enlarged from $\mathcal{N}_{I,1}$ to $\mathcal{N}_{I,2} \supset \mathcal{N}_{I,1}$, then the convergence rate improves (i.e., faster convergence with r becoming smaller). However, from (40), the network MSD may decrease, remain unchanged, or increase depending on the noise variances $\{\sigma_{v,k}^2\}$ at the new informed agents (see Figure 6); it is further explained in [60] how the deterioration of the network MSD can be avoided through proper selection of the combination weights.



[FIG6] **Enlarging the set of informed agents improves convergence rate but does not necessarily improve the MSD performance.**

BIOLOGICAL NETWORKS

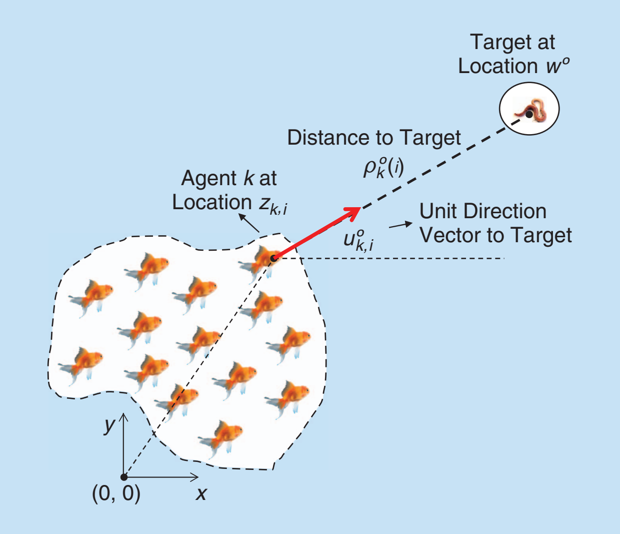
We now illustrate one application of diffusion strategies to the modeling of biological networks. Many biological systems exhibit sophisticated levels of adaptation and coordination, which result in remarkable and observable forms of collective motion and self-organization [4]. Examples include fish joining together in schools [1], bees swarming toward a new hive [2], and birds flying in formation [61]. In all of these cases, global patterns of behavior emerge from localized interactions among the individual agents.

There have been intensive studies in the literature on models for the collective motion of animal groups, most notably by using consensus-based models where agents continuously average the velocity vectors of their neighbors and move along the direction of the average (e.g., [3] and [62]–[64]). Recent experiments on the behavioral rules of fish schools challenge this approach [65], especially since it neglects the fact that the most informed agents in an animal group tend to modulate their information into their speeds. For example, fish move faster when they feel danger. As such, agents need to pay more attention to fast-moving neighbors and assign larger weights to them. In this section, following [66] and [67], we explain how to model mobile agents and how to incorporate the speed of agents into the design of the combination weights. We start with an example from [15] on target localization.

EXAMPLE 5: (TARGET LOCALIZATION)

Consider a situation in which N agents are interested in moving toward a target (such as a nutrition source); we can also consider situations where the agents want to move away from a target (such as a predator). In this example, we assume a static target and mobile agents. The unknown location of the target in the Cartesian plane is represented by the 2×1 vector w^o. The locations of the agents at time i are denoted by the 2×1 vectors $\{z_{k,i}\}$ $(k = 1, 2, \ldots, N)$; see Figure 7. We assume the agents are aware of their location vectors. The distance between agent k and the target at time i is denoted by $\rho_k^o(i) = \|w^o - z_{k,i}\|$. The 1×2 unit-norm direction vector pointing from agent k toward the target is given by $u_{k,i}^o = (w^o - z_{k,i})^\top / \|w^o - z_{k,i}\|$, so that we can also write $\rho_k^o(i) = u_{k,i}^o(w^o - z_{k,i})$. In practice, agents have noisy observations of $\{\rho_k^o(i), u_{k,i}^o\}$, which we denote by $\{\boldsymbol{\rho}_k(i), \boldsymbol{u}_{k,i}\}$. Assuming sufficiently small perturbations in $\boldsymbol{u}_{k,i}$ relative to $u_{k,i}^o$, it can be argued that these noisy observations are related to each other via the approximate model $\boldsymbol{\rho}_k(i) \approx \boldsymbol{u}_{k,i}(w^o - z_{k,i}) + \boldsymbol{v}_k(i)$ [15], [66], where $\boldsymbol{v}_k(i)$ denotes noise. If we now introduce the adjusted signal $\boldsymbol{d}_k(i) \triangleq \boldsymbol{\rho}_k(i) + \boldsymbol{u}_{k,i} z_{k,i}$, then we arrive at the same linear model (3) for the available measurement variables $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ in terms of the target location w^o:

$$\boldsymbol{d}_k(i) \approx \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i). \qquad (41)$$



[FIG7] **The distance from agent k to the target at time i is $\rho_k^o(i)$ and the unit-norm direction vector is $u_{k,i}^o$.**

The variables $\{d_k(i), u_{k,i}\}$ in (41) do not have zero means. Nevertheless, this situation can still be handled by the same diffusion mechanisms described before (see [15]). ■

MOBILE ADAPTIVE NETWORKS

We now describe a class of mobile adaptive networks that can be used to model different forms of coordinated animal behavior such as fish schooling, bee swarming, and bird flight formations [66], [68]. We focus on fish schooling due to its rich dynamic behavior [66]. Thus, consider a collection of N agents with noisy measurements $\{d_k(i), u_{k,i}\}$ that are related to some unknown target location w^o as in (41). We refer to the listing (42) and its steps (a)–(g). The agents cooperate, say, via a CTA diffusion strategy to estimate w^o. This cooperation is represented by steps (a)–(b) in (42) with combination weights $\{a^w_{\ell k}\}$, and where the neighborhood $\mathcal{N}_{k,i}$ is now allowed to vary with time (since the agents are mobile). For example, the neighborhood of a moving agent k may be defined as the set of agents that are within a certain radius of agent k at time i.

CTA diffusion adaptation for motion control over mobile networks.

```
For each agent $k$, start with $\{w_{k,-1} = 0, s^g_{k,-1} = s_{k,-1}, z_{k,0}, s_{k,0}\}$.
for $i \geq 0$ and for each agent $k$ do:
    Estimation of target location:
      (a) $\psi_{k,i-1} = \sum_{\ell \in \mathcal{N}_{k,i}} a^w_{\ell k}\, w_{\ell,i-1}$
      (b) $w_{k,i} = \psi_{k,i-1} + \mu u^*_{k,i}[d_k(i) - u_{k,i}\, \psi_{k,i-1}]$
    Estimation of velocity of center of mass:
      (c) $\phi_{k,i-1} = \sum_{\ell \in \mathcal{N}_{k,i}} a_{\ell k}\, s^g_{\ell,i-1}$                    (42)
      (d) $s^g_{k,i} = \phi_{k,i-1} + \nu[s_{k,i} - \phi_{k,i-1}]$
    Motion control (velocity vector and positioning):
      (e) $\boldsymbol{\delta}_{k,i} = \sum_{\ell \in \mathcal{N}_{k,i}\setminus\{k\}} b_{\ell k} \cdot (\| z_{\ell,i} - z_{k,i} \| - \tau) \cdot \frac{z_{\ell,i} - z_{k,i}}{\| z_{\ell,i} - z_{k,i} \|}$
      (f) $s_{k,i+1} = \alpha \cdot h(w_{k,i} - z_{k,i}) + \beta \cdot s^g_{k,i} + \eta \cdot \boldsymbol{\delta}_{k,i}$
      (g) $z_{k,i+1} = z_{k,i} + \Delta t \cdot s_{k,i+1}$
end
```

In the mobile network, it is assumed that every agent k updates its position according to step (g) in (42), where Δt represents the time step and $s_{k,i+1}$ is the velocity vector of the agent. Several factors influence the determination of $s_{k,i+1}$ by agent k such as the desire to move toward the target w^o, the desire to move in coordination with the other agents, and the desire to avoid collisions. These three objectives are represented by the three factors that appear on the right-hand side of the expression for $s_{k,i+1}$ in step (f); they are combined by means of three nonnegative weights $\{\alpha, \beta, \eta\}$. The first factor $h(w_{k,i} - z_{k,i})$ is meant to assist agent k in moving toward the target; this factor returns a vector that points toward the estimated target location, say, as follows (to prevent singularity, we let $x/\| x \| \triangleq 0$ whenever $x = 0$):

$$h(w_{k,i} - z_{k,i}) \triangleq \begin{cases} w_{k,i} - z_{k,i}, & \text{if } \| w_{k,i} - z_{k,i} \| \leq f \\ f \cdot \dfrac{w_{k,i} - z_{k,i}}{w_{k,i} - z_{k,i}}, & \text{otherwise} \end{cases} \tag{43}$$

for some positive scaling factor f used to bound the speed in pursuing the target.

The second factor $s^g_{k,i}$ that appears in step (f) refers to the estimator at agent k for the average velocity across the network. This factor is estimated by means of a second CTA diffusion strategy, represented by steps (c)–(d) in (42); this factor helps the agents attain uniform velocity and move coherently. The third factor $\boldsymbol{\delta}_{k,i}$ that appears in step (f) is computed in step (e). Its purpose is to induce attraction and repulsion behavior similar to the procedure suggested in [64] and [69] to help agents avoid collisions by maintaining some safe distance $\tau > 0$ from their neighbors. For example, if $\| z_{\ell,i} - z_{k,i} \| < \tau$, meaning that agents ℓ and k are closer to each other than the distance τ, then a factor pointing in the opposite direction of the vector $(z_{\ell,i} - z_{k,i})$ is added to the velocity vector of agent k to help it move away from agent ℓ. The opposite effect occurs when $z_{\ell,i} - z_{k,i} > \tau$. The scalars $\{b_{\ell k}\}$ in the expression for $\boldsymbol{\delta}_{k,i}$ are nonnegative weights over $\mathcal{N}_{k,i}$ satisfying $b_{kk} = 0$ and $\sum_{\ell \in \mathcal{N}_{k,i}\setminus\{k\}} b_{\ell k} = 1$; these scalars allow agent k to assign different weights to different neighbors (based, for example, on their speeds). Figure 8 illustrates the resulting maneuver of a mobile network in the plane using $N = 100$, $\alpha = \beta = 0.5$, $\eta = 1$, and $\mu = \nu = 0.5$. Similar models can be developed to incorporate avoidance of predators and to model cooperative hunting by predators.

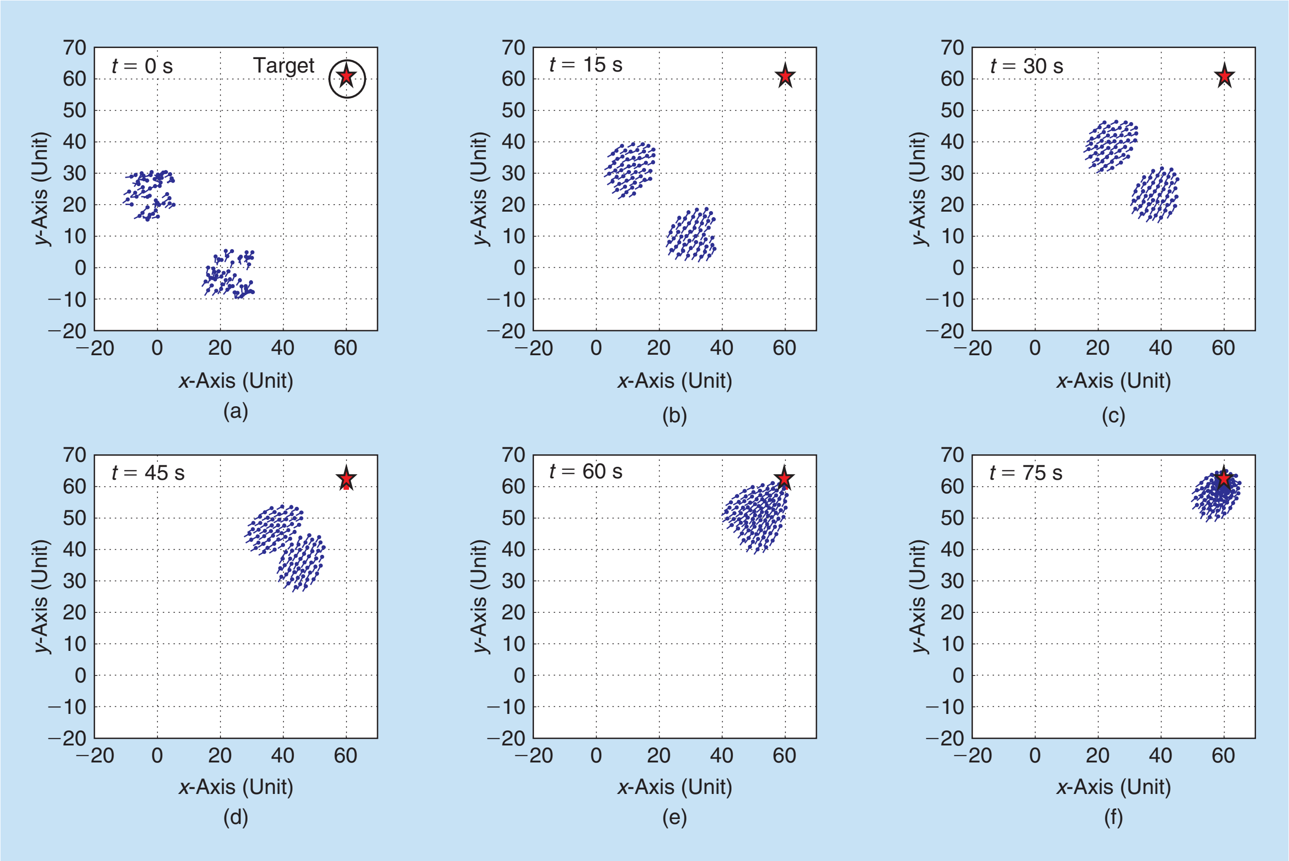
FLOW OF INFORMATION

The choice of the combination weights $\{a_{\ell k}, b_{\ell k}\}$ in (42) influences the speed with which information flows through the network. In earlier works [63], [70], uniform combination rules (or averaging strategies) were employed in consensus-based implementations such as $a_{\ell k} = 1/n_k$ for $\ell \in \mathcal{N}_k$ and $b_{\ell k} = 1/(n_k - 1)$ for $\ell \in \mathcal{N}_k \setminus \{k\}$. However, these choices do not incorporate information about the speed of the neighbors and whether they are informed. It was argued in [67] that the weight $a_{\ell k}$ (and likewise $b_{\ell k}$) can instead be chosen by agent k in proportion to the probability that its neighbor ℓ is deemed informed from observing its speed. This argument led to the following construction for the weights $\{a_{\ell k}, b_{\ell k}\}$:

$$a_{\ell k} \propto \left(1 + e^{\frac{c_1 - c_0}{\sigma_n^2}\left(\frac{c_1 + c_0}{2} - \| s_\ell \|\right)}\right)^{-1} \tag{44}$$

in terms of three parameters $\{c_0, c_1, \sigma_n^2\}$: c_0 represents the speed of an agent when it is not informed, c_1 represents the faster speed of the agent when it becomes informed (or alarmed), and σ_n^2 is a noise variance used to model speed perturbations. The resulting combination rule has a sigmoidal shape and places higher weights on faster-moving neighbors; see Figure 9.

The combination rule (44) is similar to the decision-making process in animal groups. When an agent makes a decision (such as the decision to turn around and start moving in the opposite direction), the probability of other agents following suit increases if the number of neighbors making a similar decision increases. This phenomenon is called quorum response in animal group behavior, and it was used in [71] to suggest



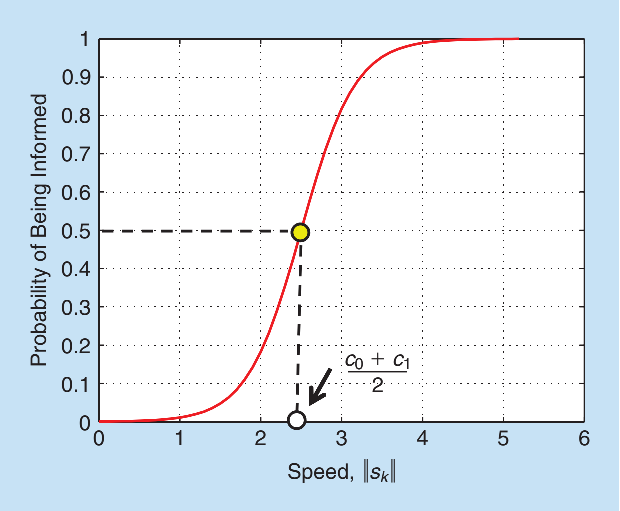
[FIG8] Parts (a)–(f) show maneuvers of a mobile network with 100 agents in $\mathbb{R}^2$ over time.

another sigmoidal-type construction for the combination weights (see [67] for comparisons).

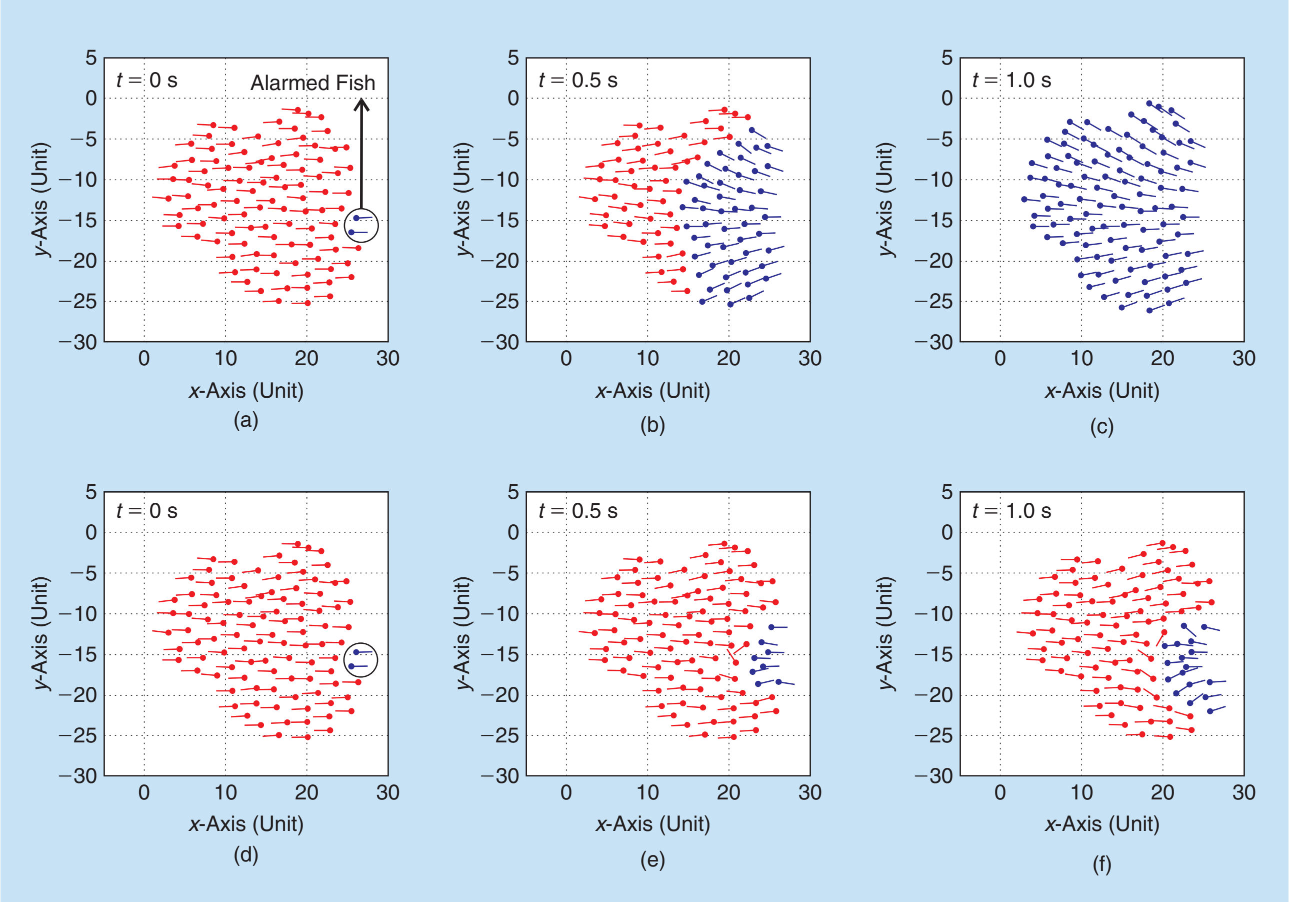
We illustrate the performance of the combination rule (44) by considering an experiment performed in [72]: when a few agents on the boundary of a perimeter are frightened, these agents change their motion rapidly and reverse their orientation. The behavior propagates through the network very quickly. To examine this effect, we consider the sigmoidal rule (44) with $(c_1, c_0, \sigma_n^2) = (4, 1, 1)$. The parameters $(\mu, \nu, \alpha, \beta, \eta)$ are set to $(0.5, 0.5, 0.5, 0.5, 1)$ for the alarmed (informed) agents, and $(0, 0.5, 0, 1, 1)$ for the uninformed agents. Figure 10 shows simulation results for a network with $N = 100$ agents.

DISTRIBUTED OPTIMIZATION

The previous sections illustrated the application of diffusion strategies to the solution of distributed MSE estimation problems of the form (11), where the individual costs are quadratic in w. The same diffusion strategies, and stochastic-gradient arguments, can be used to solve distributed optimization problems with more general individual costs $J_k(w)$ that are not necessarily quadratic. Thus, refer again to Figure 1 and consider the problem of determining, in a collaborative and distributed manner, the $M \times 1$ vector w^o that minimizes the global objective (1), where w is now assumed to be real for simplicity. The real-valued costs $\{J_k(w)\}$ are assumed to be differentiable and strongly convex functions of w, so that the aggregate cost in (1) is also strongly convex and its minimizer w^o is unique. We again focus on the important case where the component functions $\{J_k(w)\}$ are minimized at the same w^o; examples abound where



[FIG9] The larger the speed of an agent, the larger the weight assigned to it.



(a) (b) (c)

(d) (e) (f)

[FIG10] The state of alarm propagates much more rapidly through the diffusion network employing the sigmoidal rule (44) as seen in (a). Agents moving toward the positive (negative) *x*-direction are shown in red (blue): (a) sigmoidal combination rule and (b) uniform combination rule.

agents need to work cooperatively to attain a common objective (such as tracking the same target, locating the same food source, or evading the same predator). This scenario is equally common in machine learning problems [73]–[76], where data samples often arise from the same underlying distribution. The case where the $\{J_k(w)\}$ may have different minimizers is studied in [11], [12], and also [39] and [41]. It is shown in [11] and [12] that the same diffusion strategies of this section are still applicable and agents would converge to Pareto-optimal solutions.

We illustrate the optimization procedure using ATC diffusion; a similar description applies to CTA. It is explained in [31] that the following recursions are the natural extension of the ATC diffusion strategy (16) to the solution of optimization problems of the form (1)—compare with Figure 3:

$$\begin{cases} \psi_{k,i} = \boldsymbol{w}_{k,i-1} - \mu[\widehat{\nabla_w J_k}(\boldsymbol{w}_{k,i-1})]^\top \\ \psi_{k,i} = \sum\limits_{\ell\in\mathcal{N}_k} a_{\ell k}\psi_{\ell,i} \end{cases} \quad \text{(ATC diffusion)}, \qquad (45)$$

where $\widehat{\nabla_w J_k}(\boldsymbol{w}_{k,i-1})$ represents an estimate for the row gradient vector of $J_k(w)$ evaluated at $w_{k,i-1}$, and μ is a positive step size (we can again allow the step size to vary with the agent index or with time or both). More general diffusion strategies are possible where the agents do not only share their estimators $\{\psi_{\ell,\cdot}\}$ but they also share their gradient approximations $\{\widehat{\nabla_w J_\ell}(\cdot)\}$ by using a second right-stochastic combination matrix C [15], [31], [11], [12]. We again note that diffusion strategies such as (45) differ from the corresponding consensus-based solution for the same optimization problem, and which takes the following form (e.g., [48] and [49]):

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{w}_{\ell,i-1} - \mu_k(i)\cdot\left[\widehat{\nabla_w J_k}(\boldsymbol{w}_{k,i-1})\right]^\top \quad \text{(consensus)} \qquad (46)$$

usually with a time-dependent step size sequence, $\mu_k(i)$, that satisfies (6) and with a doubly stochastic matrix A.

We can assess how close the estimators $\{\boldsymbol{w}_{k,i}\}$ generated by the diffusion strategy (45) get to the optimal w^o that solves (1). For this purpose, and to be consistent with the presentation in the earlier sections where we assumed that the covariance matrix R_u was uniform across all agents, we likewise assume for the current set-up that the individual costs $\{J_k(w)\}$ have the same Hessian matrices at $w = w^o$, i.e., we assume $\nabla_w^2 J_k(w^o) \triangleq R$ for all k. This situation is common, for example, in machine learning applications where all agents are attempting to cooperatively optimize the same cost function; see Example 6 below. For performance results under more general conditions, readers are referred to [31], [11], [12]. We further let $s_k(w^o)$ denote the gradient noise at agent k at location $w = w^o$, i.e., $s_k(w^o) \triangleq [\widehat{\nabla_w J_k}(w^o) - \nabla_w J_k(w^o)]^\top$, and denote its covariance matrix by $\mathbb{E}s_k(w^o)s_k^\top(w^o) \triangleq R_{s,k}$. Then, following arguments from [12], [31], [27], and [77], and under reasonable technical conditions on the mean and variance of the gradient noise [31], it can be verified that, for standard diffusion networks and for sufficiently small step sizes, the network MSD is approximated by an expression similar to (30)–(31), specifically,

$$\mathrm{MSD}_{\mathrm{diff}}^{\mathrm{network}} \approx \mathrm{MSD}_{\mathrm{diff},k} \approx \frac{\mu}{2} \cdot \left(\sum_{k=1}^{N} p_k^2 \,\mathrm{Tr}(R_{s,k} R^{-1}) \right). \qquad (47)$$

EXAMPLE 6: (DISTRIBUTED ONLINE LEARNERS)

Consider a standard network of N learners, as in Figure 1. Each learner k receives vector samples $\{\boldsymbol{x}_{k,i}, i \geq 0\}$ that arise from some fixed probability distribution χ. The goal of the network is to learn the vector w^o that optimizes the risk function $J(w) \triangleq \mathbb{E}Q(w, \boldsymbol{x}_{k,i})$, defined in terms of some loss function $Q(\cdot,\cdot)$. To measure the performance at each agent over time, we consider the excess-risk $\mathrm{ER}_k(i) \triangleq \mathbb{E}\{J(\boldsymbol{w}_{k,i-1}) - J(w^o)\}$, where $\boldsymbol{w}_{k,i-1}$ denotes the estimator at agent k at time $i-1$ for w^o. One way to estimate w^o is for each agent k to run a stochastic gradient algorithm independently of the other agents, say,

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu(i) \cdot [\nabla_w Q(\boldsymbol{w}_{k,i-1}, \boldsymbol{x}_{k,i})]^\top, \; i \geq 0 \qquad \text{[no cooperation]} \qquad (48)$$

with a possibly iteration-dependent step size sequence $\mu(i)$, and where $\nabla_w Q(\boldsymbol{w}_{k,i-1}, \boldsymbol{x}_{k,i})$ is used to approximate $\nabla_w J(\boldsymbol{w}_{k,i-1})$. It was shown in [78] that for a strongly convex risk function $J(w)$, the noncooperative scheme (48) achieves a convergence rate of the order of $O(1/i)$ under some conditions on the gradient noise and the step size sequence $\mu(i)$. Another way to estimate w^o is for the agents to transmit their data to a central processor, which executes the following centralized algorithm:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu(i) \cdot \left(\frac{1}{N} \sum_{k=1}^{N} [\nabla_w Q(\boldsymbol{w}_{i-1}, \boldsymbol{x}_{k,i})]^\top \right), i \geq 0 \qquad \text{[centralized].} \qquad (49)$$

It can be shown that the convergence rate of this implementation is $O(1/Ni)$ for step sizes of the form $\mu(i) = \mu/(i+1)$ and for some conditions on μ. In other words, the centralized implementation (49) provides an N-fold increase in convergence rate relative to the noncooperative solution (48). The point we want to illustrate in this example is that diffusion strategies provide distributed solutions that enable every agent in the network to converge at the same rate as the centralized solution (49) [77]. We illustrate this point by considering the ATC diffusion strategy (45) with an iteration-dependent step size

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu(i) \cdot [\nabla_w Q(\boldsymbol{w}_{k,i-1}, \boldsymbol{x}_{k,i})]^\top \\ \boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i}. \end{cases} \qquad (50)$$

Some algebra would show that, for large enough i, the excess-risk can be expressed as a weighted mean-square error, $\mathrm{ER}_k(i) \approx (1/2)\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i-1}\|_R^2$, with the weighting matrix $R = \nabla_w^2 J(w^o)$ [77]. We can extend the analysis from the section "Mean-Square-Error Performance" to the case of time-dependent step sizes of the form $\mu(i) = \mu/(i+1)$, and conclude that for standard networks, the excess-risk at agent k and for large enough i can be approximated by (now $R_{s,k} = R_s$ for $k = 1, 2, \ldots, N$) [77]:

$$\mathrm{ER}_k(i) \approx \frac{\mu}{4} \cdot \frac{\mathrm{Tr}(R_s)}{i} \cdot \left(\sum_{k=1}^{N} p_k^2 \right), \quad \text{for large } i, \qquad (51)$$

where p is the right-eigenvector of A defined by (30). When A is doubly stochastic, we have $p = 1/N$ and the above result would then imply that every agent in the network will improve its excess risk [in comparison to the noncooperative solution (48)] by a factor of N. In addition, each agent will converge at the same $O(1/Ni)$ rate as the centralized algorithm (49). ■

CONCLUSIONS

There are several other aspects of diffusion strategies for adaptation and learning over networks that were not covered in this article due to space limitations. For instance, we ignored in our presentation the effect of perturbations during the exchange of information over edges among neighboring agents. Noise over the communication links can be due to various factors including thermal noise and imperfect channel information. Studying the degradation in mean-square performance that results from these noisy exchanges, and developing adaptive combination policies that counter the effect of the degradation, can be pursued by extending the mean-square analysis of the section "Mean-Square-Error Performance." Readers can refer to [57], [15], and [79], and the references therein for details. Studies on consensus-based solutions with noisy exchanges appear in [80] and [81]. Several other extensions and variations of diffusion strategies are possible. Among these variations we mention strategies that endow agents with temporal processing abilities, in addition to their spatial cooperation abilities [15], [82], [83]. For example, in the ATC implementation (16), rather than have each agent k rely solely on current weight estimators received from its neighbors, $\{\boldsymbol{\psi}_{\ell,i}, \ell \in \mathcal{N}_k\}$, agent k can also be allowed to store and process past weight estimators. We can also apply diffusion strategies to solve recursive least-squares and state-space estimation problems in a distributed manner [84], [36]; consensus-based solutions for state-space estimation appear in [85] and [86]. Finally, one interesting conclusion that stands out in our presentation is that the performance of diffusion strategies is largely determined by the right-eigenvector of the combination policy corresponding to the eigenvalue at one. This observation can be used to propose useful procedures to control the convergence behavior of distributed strategies and to design effective combination procedures [27], [77], [12].

ACKNOWLEDGMENTS

This work was supported in part by NSF grants CCF-0942936 and CCF-1011918.

AUTHORS

Ali H. Sayed (sayed@ee.ucla.edu) is a professor of electrical engineering at the University of California, Los Angeles, where he directs the Adaptive Systems Laboratory. He is an author of over 400 scholarly publications and five books, and his research involves several areas including adaptation and learning, network science, biologically inspired designs, and information processing theories. His work received several recognitions including the 2012 Technical Achievement Award from the IEEE Signal Processing Society, the 2005 Terman Award from the American Society for Engineering Education, a 2005 Distinguished Lecturer from the IEEE Signal Processing Society, the 2003 Kuwait Prize, and the 1996 IEEE Fink Prize. He has also been awarded several Best Paper Awards from the IEEE and is a Fellow of both the IEEE and the American Association for the Advancement of Science.

Sheng-Yuan Tu (shinetu@ucla.edu) received the B.S. and M.S. degrees in electrical engineering from the National Taiwan University, Taiwan, in 2005 and 2007, respectively. He is currently completing his Ph.D. degree in electrical engineering at the University of California, Los Angeles. His research interests include distributed signal processing and self-organization in biological systems.

Jianshu Chen (cjs09@ucla.edu) received his B.S. and M.S. degrees in electronic engineering from Harbin Institute of Technology in 2005 and Tsinghua University in 2009, respectively. He is currently working toward the Ph.D. degree in electrical engineering at the University of California, Los Angeles. His research interests include statistical signal processing, adaptive networks, and distributed optimization.

Xiaochuan Zhao (xiaochuanzhao@ucla.edu) received the B.S. and M.S. degrees in electrical engineering from the Beijing University of Posts and Telecommunications (BUPT), China, in 2006 and 2009, respectively. He is currently pursuing the Ph.D. degree in electrical engineering at the University of California, Los Angeles. His research interests include distributed signal processing, statistical signal processing, and communication theory. He received the UCLA Graduate Division Fellowship in 2009.

Zaid J. Towfic (ztowfic@ee.ucla.edu) received his B.S. degrees in electrical engineering, computer science, and mathematics from the University of Iowa in 2007. He received his M.S. degree in electrical engineering from the University of California, Los Angeles in 2009. From 2007 to 2008, he worked for Rockwell Collins' Advanced Technology Center (Cedar Rapids, Iowa) working on MIMO, embedded sensing, and digital modulation classification. He is currently pursuing his Ph.D. degree in electrical engineering at the University of California, Los Angeles. His research interests include distributed signal processing, self-healing circuitry, stochastic optimization, and machine learning.

REFERENCES

[1] B. L. Partridge, "The structure and function of fish schools," *Sci. Amer.*, vol. 246, no. 6, pp. 114–123, June 1982.

[2] M. Beekman, R. L. Fathke, T. D. Seeley, "How does an informed minority of scouts guide a honey bee swarm as it flies to its new home?" *Animal Behav.*, vol. 71, no. 1, pp. 161–171, Jan. 2006.

[3] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavior model," in *ACM Proc. Computer Graphics and Interactive Techniques*, 1987, pp. 25–34.

[4] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-Organization in Biological Systems*. Princeton, NJ: Princeton Univ. Press, 2003.

[5] M. Jackson, *Social and Economic Networks*. Princeton, NJ: Princeton Univ. Press, 2008.

[6] J. Fowler and N. Christakis, "Cooperative behavior cascades in human social networks," *Proc. Nat. Acad. Sci.*, vol. 107, no. 12, pp. 5334–5338, 2010.

[7] C. P. Chamley, *Rational Herds*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[8] D. Acemoglu and A. Ozdaglar, "Opinion dynamics and learning in social networks," *Dyn. Games Appl.*, vol. 1, no. 1, pp. 3–49, Mar. 2011.

[9] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-Bayesian social learning," *Game. Econ. Behav.*, vol. 76, no. 1, pp. 210–225, Sept. 2012.

[10] X. Zhao and A. H. Sayed, "Learning over social networks via diffusion adaptation," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Nov. 2012, pp. 1–5.

[11] J. Chen and A. H. Sayed, "Distributed Pareto-optimal solutions via diffusion adaptation," in *Proc. IEEE Statistical Signal Processing Workshop*, Ann Arbor, MI, Aug. 2012, pp. 1–4.

[12] J. Chen and A. H. Sayed, "On the limiting behavior of distributed optimization strategies," in *Proc. 50th Annu. Allerton Conf. Communication, Control, and Computing*, Allerton, IL, Oct. 2012, pp. 1–8.

[13] Z. Towfic, J. Chen, and A. H. Sayed, "Distributed throughput optimization over P2P mesh networks using diffusion adaptation," in *Proc. IEEE ICC*, Ottawa, Canada, June 2012, pp. 648–652.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[15] A. H. Sayed, "Diffusion adaptation over networks," in *E-Reference Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Amsterdam, The Netherlands: Elsevier, 2013 (see also arXiv:1205.4220v1 [cs.MA], May 2012).

[16] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Processing*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[17] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. New York: Springer, 2010.

[18] M. Nevelson and R. Hasminskii, *Stochastic Approximation and Recursive Estimation*. Providence, RI: American Mathematical Society, 1973.

[19] O. Macchi, *Adaptive Processing: The Least Mean Squares Approach with Applications in Transmission*. Hoboken, NJ: Wiley, 1995.

[20] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ: Wiley, 2008.

[21] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice Hall, 2002.

[22] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1985.

[23] S. Jones, R. C. III, and W. Reed, "Analysis of error-gradient adaptive linear estimators for a class of stationary dependent processes," *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 318–329, Mar. 1982.

[24] W. A. Gardner, "Learning characterisitcs of stochastic-gradient-descent algorithms: A general study, analysis, and critique," *Signal Processing*, vol. 6, no. 2, pp. 113–133, Apr. 1984.

[25] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 1, pp. 222–230, Feb. 1985.

[26] F. Cattivelli and A. H. Sayed, "Analysis of spatial and incremental LMS processing for distributed estimation," *IEEE Trans. Signal Processing*, vol. 59, no. 4, pp. 1465–1480, Apr. 2011.

[27] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.

[28] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[29] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[30] S. U. Pillai, T. Suel, and S. Cha, "The Perron–Frobenius theorem: Some of its applications," *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 62–75, Mar. 2005.

[31] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Processing*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[32] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," in *Proc. Adaptive Sensor Array Processing Workshop*, MIT Lincoln Laboratory, MA, June 2006, pp. 1–5.

[33] A. H. Sayed and C. G. Lopes, “Adaptive processing over distributed networks,” *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E90-A, no. 8, pp. 1504–1510, Aug. 2007.

[34] C. G. Lopes and A. H. Sayed, “Diffusion least-mean-squares over adaptive networks,” in *Proc. IEEE ICASSP*, Honolulu, Hawaii, Apr. 2007, vol. 3, pp. 917–920.

[35] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, “A diffusion RLS scheme for distributed estimation over adaptive networks,” in *Proc. IEEE Workshop on Signal Processing Advances Wireless Communication (SPAWC)*, Helsinki, Finland, June 2007, pp. 1–5.

[36] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, “Diffusion recursive least-squares for distributed estimation over adaptive networks,” *IEEE Trans. Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[37] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS algorithms with information exchange,” in *Proc. Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2008, pp. 251–255.

[38] F. S. Cattivelli and A. H. Sayed, “Diffusion mechanisms for fixed-point distributed Kalman smoothing,” in *Proc. EUSIPCO*, Lausanne, Switzerland, Aug. 2008, pp. 1–4.

[39] S. S. Ram, A. Nedic, and V. V. Veeravalli, “Distributed stochastic subgradient projection algorithms for convex optimization,” *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

[40] P. Bianchi, G. Fort, W. Hachem, and J. Jakubowicz, “Convergence of a distributed parameter estimator for sensor networks with local averaging of the estimates,” in *Proc. IEEE ICASSP,* Prague, Czech, pp. 3764–3767, May 2011.

[41] K. Srivastava and A. Nedic, “Distributed asynchronous constrained stochastic optimization,” *IEEE J. Select. Topics. Signal Processing*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

[42] S. S. Stankovic, M. S. Stankovic, and D. S. Stipanovic, “Decentralized parameter estimation by consensus based stochastic approximation,” *IEEE Trans. Autom. Contr.*, vol. 56, no. 3, pp. 531–543, Mar. 2011.

[43] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sept. 2004.

[44] L. Xiao, S. Boyd, and S. Lall, “A space-time diffusion scheme peer-to-peer least-squares-estimation,” in *Proc. Information Processing in Sensor Networks* (IPSN), Nashville, TN, Apr. 2006, pp. 168–176.

[45] S. Barbarossa, and G. Scutari, “Bio-inspired sensor network design,” *IEEE Signal Processing Mag.*, vol. 24, no. 3, pp. 26–35, May 2007.

[46] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[47] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson, “Subgradient methods and consensus algorithms for solving convex optimization problems,” in *Proc. IEEE CDC*, Cancun, Mexico, Dec. 2008, pp. 4185–4190.

[48] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 1st ed. Singapore: Athena Scientific, 1997.

[49] A. Nedic and A. Ozdaglar, “Cooperative distributed multi-agent optimization,” in *Convex Optimization in Signal Processing and Communications*, Y. Eldar and D. Palomar, Eds. Cambridge, U.K.: Cambridge Univ. Press, pp. 340–386, 2010.

[50] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Trans. Autom. Contr.*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[51] S. Kar and J. M. F. Moura, “Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs,” *IEEE J. Select. Topics Signal Processing*, vol. 5, no. 4, pp. 674–690, Aug. 2011.

[52] S.-Y. Tu and A. H. Sayed, “Diffusion networks outperform consensus networks,” in *Proc. IEEE Statistical Signal Processing Workshop*, Ann Arbor, MI, Aug. 2012, pp. 1–4.

[53] S.-Y. Tu and A. H. Sayed, “Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks,” *IEEE Trans. Signal Processing*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[54] J. Surowiecki, *The Wisdom of the Crowds*. New York: Doubleday, 2004.

[55] W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1970.

[56] S. Boyd, P. Diaconis, and L. Xiao, “Fastest mixing Markov chain on a graph,” *SIAM Rev.*, vol. 46, no. 4, pp. 667–689, Dec. 2004.

[57] X. Zhao, S.-Y. Tu, and A. H. Sayed, “Diffusion adaptation over networks under imperfect information exchange and non-stationary data,” *IEEE Trans. Signal Processing*, vol. 60, no. 7, pp. 3460–3475, July 2012.

[58] X. Zhao and A. H. Sayed, “Clustering via diffusion adaptation over networks,” in *Proc. Int. Workshop Cognitive Information Processing (CIP)*, Parador de Baiona, Spain, May 2012, pp. 1–6.

[59] S.-Y. Tu and A. H. Sayed, “On the effects of topology and node distribution on learning over complex adaptive networks,” in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2011, pp. 1166–1171.

[60] S.-Y. Tu and A. H. Sayed, “On the influence of informed agents on learning and adaptation over networks,” *IEEE Trans. Signal Processing*, vol. 61, 2013 (see also arXiv:1203.1524v1 [cs.IT], Mar. 2012).

[61] D. Hummel, “Aerodynamic aspects of formation flight in birds,” *J. Theor. Biol.*, vol. 104, no. 3, pp. 321–347, 1983.

[62] T. Vicsek, A. Czirook, E. Ben-Jacob, O. Cohen, and I. Shochet, “Novel type of phase transition in a system of self-driven particles,” *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, Aug. 1995.

[63] A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Trans. Autom. Contr.*, vol. 48, no. 6, pp. 988–1001, June 2003.

[64] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” *IEEE Trans. Autom. Contr.*, vol. 51, pp. 401–420, Mar. 2006.

[65] Y. Katz, C. C. Ioannou, K. Tunstrom, C. Huepe, and I. D. Couzin, “Inferring the structure and dynamics of interactions in schooling fish,” *Proc. Nat. Acad. Sci.*, vol. 108, no. 46, pp. 18720–18725, Nov. 2011.

[66] S.-Y. Tu and A. H. Sayed, “Mobile adaptive networks,” *IEEE J. Select. Topics Signal Processing*, vol. 5, no. 4, pp. 649–664, Aug. 2011.

[67] S.-Y. Tu and A. H. Sayed, “Effective information flow over mobile adaptive networks,” in *Proc. Int. Workshop Cognitive Information Processing (CIP)*, Parador de Baiona, Spain, May 2012, pp. 1–6.

[68] F. Cattivelli and A. H. Sayed, “Modeling bird flight formations using diffusion adaptation,” *IEEE Trans. Signal Processing*, vol. 59, no. 5, pp. 2038–2051, May 2011.

[69] S. Janson, M. Middendorf, and M. Beekman, “Honeybee swarms: How do scouts guide a swarm of uninformed bees?” *Animal Behav.*, vol. 70, pp. 349–358, 2005.

[70] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, “Collective memory and spatial sorting in animal groups,” *J. Theor. Biol.*, vol. 218, pp. 1–11, 2002.

[71] D. J. T. Sumpter and S. C. Pratt, “Quorum responses and consensus decision making,” *Philos. Trans. Roy. Soc. B*, vol. 364, pp. 743–753, Dec. 2009.

[72] D. J. T. Sumpter, J. Buhl, D. Biro, and I. D. Couzin, “Information transfer in moving animal groups,” *Theory Biosci.*, vol. 127, no. 2, pp. 177–186, 2008.

[73] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2007.

[74] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. New York: Academic Press, 2008.

[75] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, “Optimal distributed online prediction,” in *Proc. Int. Conf. Machine Learning (ICML)*, Bellevue, WA, June 2011, pp. 713–720.

[76] J. B. Predd, S. B. Kulkarni, and H. V. Poor, “Distributed learning in wireless sensor networks,” *IEEE Signal Processing Mag.*, vol. 23, no. 4, pp. 56–69, July 2006.

[77] Z. Towfic, J. Chen, and A. H. Sayed, “On the generalization ability of distributed online learners,” in *Proc. IEEE Workshop Machine Learning for Signal Processing* (MLSP), Santander, Spain, Sept. 2012, pp. 1–6.

[78] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, “Robust stochastic approximation approach to stochastic programming,” *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574–1609, Jan. 2009.

[79] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers, “Steady state analysis of diffusion LMS adaptive networks with noisy links,” *IEEE Trans. Signal Processing*, vol. 60, no. 2, pp. 974–979, Feb. 2012.

[80] S. Kar and J. M. F. Moura, “Distributed consensus algorithms in sensor networks: Link failures and channel noise,” *IEEE Trans. Signal Processing*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[81] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Performance analysis of the consensus-based distributed LMS algorithm,” *EURASIP J. Adv. Signal Process.*, pp. 1–19, 2009, 10.1155/2009/981030, Article ID 981030.

[82] J.-W. Lee, S.-E. Kim, W.-J. Song, and A. H. Sayed, “Spatio-temporal diffusion strategies for estimation and detection over networks,” *IEEE Trans. Signal Processing*, vol. 60, no. 8, pp. 4017–4034, Aug. 2012.

[83] S. Chouvardas, K. Slavakis, and S. Theodoridis, “Adaptive robust distributed learning in diffusion sensor networks,” *IEEE Trans. Signal Processing*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[84] F. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Trans. Autom. Contr.*, vol. 55, no. 9, pp. 2069–2084, Sept. 2010.

[85] R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” in *Proc. IEEE CDC*, Shangai, China, 2009, pp. 7036–7042.

[86] U. A. Khan and J. M. F. Moura, “Distributing the Kalman filter for large-scale systems,” *IEEE Trans. Signal Processing*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

[87] J. Chen and A. H. Sayed, “Distributed Pareto optimization via diffusion strategies,” *IEEE J. Selected Topics in Signal Processing*, vol. 7, Apr. 2013.