LEARNING BY NETWORKED AGENTS UNDER PARTIAL INFORMATION

Chung-Kai Yu and Ali H. Sayed
Department of Electrical Engineering
University of California, Los Angeles

ABSTRACT
In many scenarios of interest, agents may only have access to partial information about an unknown model or target vector. Each agent may be sensing only a subset of the entries of a global target vector, and the number of these entries can be different across the agents. If each of the agents were to solve an inference task independently of the other agents, they would not benefit from neighboring agents that may be sensing similar entries. This work develops cooperative distributed techniques that enable agents to cooperate even when their interactions are limited to exchanging estimates of select few entries. In the proposed strategies, agents are only required to share estimates of their common entries, which results in a significant reduction in communication overhead. Simulations show that the proposed approach improves both the performance of individual agents and the entire network through cooperation.

Index Terms— Adaptive network, constrained optimization, penalty function, diffusion strategy, distributed estimation.

1. INTRODUCTION AND RELATED WORK

Adaptive networks enable agents to share information and to solve distributed optimization and inference tasks in an online and decentralized manner. In most prior works, agents are assumed to have a common minimizer and cooperate to estimate it by using effective distributed strategies such as the consensus strategy (e.g., [1–8]) or the diffusion strategy (e.g., [9–12]). When the agents do not share a minimizer, it was shown in [9,10,13,14] that the network converges to a Pareto optimal solution. When it is desired instead that agents, or clusters of agents within the network, should converge to their respective models, rather than the Pareto solution, then multi-task diffusion strategies become useful and can be used to attain this objective [15,16].

In these earlier contributions, it is generally assumed that the target vector for each agent has the same size and, moreover, that the agents sense data that is affected by all entries of their target vectors. In this work, we relax these conditions and consider a broader scenario where individual agents sense only a subset of the entries of the global target vector, and where different agents can sense subsets of different sizes. This formulation allows us to model the important situation in which agents may only have access to partial information about an unknown model or target vector. If each of the agents were to solve an inference task independently of the other agents, they would not benefit from cooperation with neighboring agents that may be sensing common entries. This work develops cooperative distributed techniques that enable agents to cooperate even when their interactions are limited to exchanging estimates of select few entries. To attain this objective, we allow for some entries of the global target vector to be observable by more than one agent so that cooperation across the network is justified.

Our approach will be based on formulating a constrained optimization problem for recovering partial entries of the global target vector. However, rather than solve this problem directly, we will introduce a penalized version using a quadratic term to penalize the violation of the constraints. We will then develop a diffusion learning solution to solve the optimization problem in a distributed manner by relying on two incremental steps. In the adaptation step, agents descend along the negative direction of the gradients of their costs. And in the penalty step, they descend along the negative direction of the gradients of their penalties. When the exact gradient information is unavailable, the observed data is used to compute instantaneous approximations for the gradient vectors. In the penalty step, agents will only share the common entries of their estimates with neighbors to reduce the communication costs. The order of executing the two incremental steps results in the Adapt-then-Penalize (ATP) or Penalize-then-Adapt (PTA) diffusion strategies.

We remark that this work considers a more general scenario than the partial diffusion formulation proposed in [17]. There, all agents sense data driven by the same target vector, cooperate to estimate this target vector, and exchange only part of their entries. In our formulation, each agent will be sensing data driven by different local target vectors and these can be of different sizes. In this way, agents are able to cooperate even if their target vectors are only partially common. We then show that sufficiently small step-sizes ensure mean and mean-square stability. We illustrate the results by means of computer simulations.

Notation: We use lowercase letters to denote vectors and scalars, uppercase letters for matrices, plain letters for deterministic variables, and boldface letters for random variables.

2. PROBLEM FORMULATION

At each time instant \( i \geq 0 \), each agent \( k \) is assumed to have access to a scalar measurement \( d_k(i) \in \mathbb{R} \) and a regression vector \( u_{k,i} \in \mathbb{R}^{1 \times M_k} \) with covariance matrix \( R_{u,k} = \ldots \)
\[ \mathbb{E}u_{k,i}^T \mathbf{u}_{k,i} > 0. \] The regressors \( \{ \mathbf{u}_{k,i} \} \) are assumed to have zero mean and to be temporally white and spatially independent. The data \( \{ d_k(i), \mathbf{u}_{k,i} \} \) are assumed to be related via the linear regression model:
\[ d_k(i) = \mathbf{u}_{k,i} \mathbf{w}_k^o + \mathbf{v}_k(i) \tag{1} \]
where \( \mathbf{w}_k^o \in \mathbb{R}^{M_k \times 1} \) is the target vector to be estimated by agent \( k \). The variable \( \mathbf{v}_k(i) \in \mathbb{R} \) is a zero-mean white-noise process with variance \( \mathbb{E} \mathbf{v}_k^2(i) = \sigma^2_{\mathbf{v}_k} \) and assumed to be spatially independent. We further assume that the random processes \( u_{k,i} \) and \( v_k(j) \) are spatially and temporally independent for any \( k, \ell, i, \) and \( j \). We assume that each entry of \( \mathbf{w}_k^o \) is determined by a grand target vector \( \mathbf{w}^o \in \mathbb{R}^{M \times 1} \), i.e., the relation between \( \mathbf{w}_k^o \) and \( \mathbf{w}^o \) can be described by
\[ \mathbf{w}_k^o = D_k \mathbf{w}^o \tag{2} \]
where \( D_k \) is a matrix of size \( M_k \times M \) and defined as
\[ D_k(s, m) = \begin{cases} 1, & \text{if } w_k^o(s) \leftrightarrow w^o(m) \\ 0, & \text{otherwise} \end{cases} \tag{3} \]
The notation \( x \leftrightarrow y \) denotes that the value of \( y \) is assigned to \( x \). We are therefore considering a distributed inference problem where each agent has partial information about a grand target vector, i.e., the data at each agent is influenced by only some entries of \( \mathbf{w}^o \). Observe that the size \( M_k \) of the vector \( \mathbf{w}_k^o \) is allowed to change with the node index, \( k \), so that some agents may be influenced by more entries than other agents.

Now, given a network topology, two neighboring agents \( \{ k, \ell \} \) may share one or more common target entries, e.g., there can exist some index \( m \in \{ 1, \ldots, M \} \) such that
\[ w_k^o(s) \leftrightarrow w^o(m), \quad w_\ell^o(s') \leftrightarrow w^o(m), \quad \ell \in \mathcal{N}_k \setminus \{ k \} \tag{4} \]
where \( \mathcal{N}_k \) is the neighborhood of agent \( k \). We are therefore motivated to consider the following constrained optimization problem for each agent \( k \):
\[
\begin{aligned}
& \min_{ \mathbf{w}_k } \quad J_k(\mathbf{w}_k) \triangleq \frac{1}{2} \mathbb{E}|d_k(i) - \mathbf{u}_{k,i} \mathbf{w}_k|^2 \\
& \text{subject to} \quad \mathbf{w}_k(s) = \mathbf{w}_\ell(s'), \ell \in \mathcal{N}_k \setminus \{ k \}, \\
& \quad s \in \{ 1, \ldots, M_k \}, \quad s' \in \{ 1, \ldots, M_\ell \} 
\end{aligned} \tag{5} \]
The indices \( s \) and \( s' \) in (5) refer only to the common entries in \( \mathbf{w}_k^o \) and \( \mathbf{w}_\ell^o \). We provide an example in Fig. 1 to illustrate the setting defined in (5). For example, agents #1 and #2 share the common target entry \( \mathbf{w}^o(2) \); it is the leading element in the target vector for agent #2 and the trailing element in the target vector for agent #1. Observe that agents can share target entries even if they are not neighbors, as is the case with agents #1 and #3; they both share entry \( \mathbf{w}^o(1) \). However, the constraints for the common target entries can only exist between neighboring agents. For convenience and for later use, we collect the constraints for each agent \( k \) into a constraint set \( \mathcal{S}_k \):
\[
\mathcal{S}_k \triangleq \left\{ (\ell, s, s') \mid \mathbf{w}_k(s) = \mathbf{w}_\ell(s'), \ell \in \mathcal{N}_k \setminus \{ k \} \\
\quad s \in \{ 1, \ldots, M_k \}, \quad s' \in \{ 1, \ldots, M_\ell \} \right\} 
\tag{6} \]

![Fig. 1. An example to illustrate distributed inference under partial information exchange.](image)

### 3. PENALTY-BASED LEARNING

One way to solve problem (5) is to reformulate it using penalty functions. Specifically, instead of solving (5), we consider the penalized version:
\[
\min_{ \mathbf{w}_k } \quad J_k^p(\mathbf{w}_k) \triangleq \frac{1}{2} \mathbb{E}|d_k(i) - \mathbf{u}_{k,i} \mathbf{w}_k|^2 + p_k(\mathbf{w}_k) \tag{7} \]
where the notation \( \mathbf{w}_k \triangleq \text{col}\{ \mathbf{w}_k(\ell) ; \ell \in \mathcal{N}_k \} \) aggregates all unknowns in the neighborhood \( \mathcal{N}_k \) and \( p_k(\mathbf{w}_k) \) is a quadratic penalty function used to penalize agent \( k \) when any constraint \( \mathbf{w}_k(s) = \mathbf{w}_\ell(s') \) is violated, i.e.,
\[
p_k(\mathbf{w}_k) \triangleq \sum_{(\ell, s, s') \in \mathcal{S}_k} \rho_k(\ell, s, s') \cdot |\mathbf{w}_k(s) - \mathbf{w}_\ell(s')|^2 \tag{8} \]
where \( \rho_k(\ell, s, s') \) is a positive penalty parameter used to control the punishment level of violating the constraint \( \mathbf{w}_k(s) = \mathbf{w}_\ell(s') \). Other choices for the penalty function are possible. It is sufficient for our purposes in this article to illustrate the main construction and results using (8).

### 3.1. Entry-Wise Diffusion Implementation

Following the approach from [18], the optimization problem (7) can be solved in two incremental steps: we first adapt with respect to \( J_k(\mathbf{w}_k) \) and then adapt with respect to \( p_k(\mathbf{w}_k) \). For this purpose, we start by noting that the gradient vector of \( J_k(\mathbf{w}_k) \) with respect to \( \mathbf{w}_k \) is given by
\[
\nabla_{\mathbf{w}^i_k} J_k(\mathbf{w}_k) = R_{u,k} \mathbf{w}_k - r_{du,k} \tag{9} \]
where \( r_{du,k} = \mathbb{E} \mathbf{d}_k(i) \mathbf{u}^T_{k,i} \mathbf{u}^i_k \mathbf{u}^i_k \). When the gradient of \( J_k(\mathbf{w}_k) \) is unavailable, we can approximate it by using the instantaneous approximations \( r_{du,k} \approx \mathbf{d}_k(i) \mathbf{u}^T_{k,i} \mathbf{u}^i_k \mathbf{u}^i_k \) and \( R_{u,k} \approx \mathbf{u}^T_{k,i} \mathbf{u}^i_k \mathbf{u}^i_k \):
\[
\nabla_{\mathbf{w}^i_k} J_k(\mathbf{w}_k) = \mathbf{u}^T_{k,i} [\mathbf{u}^i_k \mathbf{w}_k - \mathbf{d}_k(i)] \tag{10} \]
By doing so, we arrive at the adapt-then-penalize (ATP) diffusion strategy:
\[
\begin{aligned}
& \{ \psi_{k,i}^j = w_{k,i-1} + \mu_k \mathbf{u}^T_{k,i} [d_k(i) - \mathbf{u}_{k,i} w_{k,i-1}] \} \\
& \{ w_{k,i} = \psi_{k,i} - \mu_k \nabla_{\mathbf{w}^i_k} p_k(\psi_{k,i}) \} \tag{11} \end{aligned} \tag{12} \]
where \( \psi_{k}^j \triangleq \text{col}\{ \psi_{k,i} ; \ell \in \mathcal{N}_k \} \). By differentiating the penalty function \( p_k(\psi_{k,i}) \) with respect to \( \mathbf{w}_k \), it can be verified
that step (12) can be rewritten as

\[ w_{k,i} = \sum_{\ell \in \mathbb{N}_k} A_{k}^T \psi_{\ell,i}, \tag{13} \]

where \( A_{k} \) is the \( M_k \times M_k \) matrix with entries defined by

\[ A_{k}(s', s) = \begin{cases} 2\mu_{k}\rho_{k}(\ell, s, s'), & \text{if } (\ell, s, s') \in \mathbb{S}_k \\ 0, & \text{otherwise} \end{cases} \tag{14} \]

for \( k \neq \ell \), and

\[ A_{kk} = \text{diag} \left\{ 1 - \sum_{(\ell, s, s') \in \mathbb{S}_k} 2\mu_{k}\rho_{k}(\ell, 1, s'), \ldots, 1 - \sum_{(\ell, s, s') \in \mathbb{S}_k} 2\mu_{k}\rho_{k}(\ell, M_k, s') \right\}. \tag{15} \]

Note that

\[ [A_{1k}^T \ldots A_{Nk}^T] \mathbf{1} = \mathbf{1} \tag{16} \]

where \( \mathbf{1} \) denotes the vector with all one entries. Observe that the matrix \( A_{k} \) defines the combination weights between agent \( k \) and the common entries with agent \( \ell \) from its neighborhood; we assume the step-sizes \( \{\mu_{k}\} \) are sufficiently small such that the diagonal entries in \( A_{k} \) are nonnegative so that \( A = [A_{k}] \) is a left-stochastic matrix. It turns out that the particular forms (14) and (15) are not critical. It is sufficient to select an arbitrary left-stochastic matrix \( A \) as long as the zero-structure of its block components \( A_{k} \) and property (16) are satisfied. We can also switch the order of the incremental steps in (11)–(12) and arrive at the penalize-then-adapt (PTA) diffusion strategy:

\[
\begin{cases}
\psi_{k,i} = \sum_{\ell \in \mathbb{N}_k} A_{k}^T w_{\ell,i-1} \\
\psi_{k,i} = \psi_{k,i} + \mu_{k} u_{k,i}^T [d_{k}(i) - u_{k,i} \psi_{k,i}] 
\end{cases}
\tag{17}
\]

The PTA case corresponds to the choice \( A_1 = A \) and \( A_2 = I \) while the ATP case corresponds to \( A_1 = I \) and \( A_2 = A \). To continue, we note the following useful properties:

\[
A_2 A_1 = A, \quad A^T D = D, \quad A_1^T D = D, \quad A_2^T D = D, \quad A^T \mathbf{1} = \mathbf{1} \tag{31}
\]

From (24) and (28) we can write

\[
\begin{align*}
\psi_{i} - \phi_{i} &= D w^o - A_1^T w_{i-1} = A_1^T (w^o - w_{i-1}) \\
\psi_{i} - \phi_{i} &= \psi_{i} - \mathcal{M} \mathcal{R}_{i} \psi_{i} - M s_{i} \\
\psi_{i} &= \psi_{i} - M s_{i} \\
\end{align*}
\tag{32}
\]

and similarly,

\[
\begin{align*}
\psi_{i} - \phi_{i} &= D w^o - A_1^T w_{i-1} = A_1^T (w^o - w_{i-1}) \\
\psi_{i} - \phi_{i} &= \psi_{i} - \mathcal{M} \mathcal{R}_{i} \psi_{i} - M s_{i} \\
\psi_{i} &= \psi_{i} - M s_{i} \\
\end{align*}
\tag{33}
\]

Therefore, we arrive at the network error recursions:

\[
\begin{align*}
\bar{\psi}_{i} &= \bar{\psi}_{i} - \mathcal{M} \mathcal{R}_{i} \bar{\psi}_{i} - M s_{i} \\
\bar{\psi}_{i} &= \mathcal{M} \mathcal{R}_{i} \bar{\psi}_{i} - M s_{i} \\
\bar{\psi}_{i} &= \mathcal{M} \mathcal{R}_{i} \bar{\psi}_{i} - M s_{i} \\
\end{align*}
\tag{34}
\]

where \( \bar{\psi}_{i} \) is the error term for the other error vectors. Combining (34)–(36) we conclude that the error vector \( \bar{\psi}_{i} \) evolves according to the recursion:

\[
\bar{\psi}_{i} = \mathcal{B}_{i} \bar{\psi}_{i-1} - G s_{i} \tag{37}
\]
where $B_i \triangleq A_i^T(I-MR_i)A_i^T$ and $G \triangleq A_i^T M$. It was shown in [11] that the traditional ATC and CTA diffusion algorithms have a network error recursion of the same general form as (37), except that now we have one critical difference. Expression (37) is more general and allows agents to have different sizes for their target vectors $\{w_o^i\}$. Furthermore, the matrices $A_1$ and $A_2$ now reflect refined connections: two connected agents only share a subset of their entries, which can be a single entry in the extreme case. Therefore, cooperation between agents is limited to entry-wise exchanges, as opposed to full vector exchanges in traditional implementations. Following similar arguments to those in [11], we can derive conditions on the step-size parameters to ensure mean-square convergence and stability. Proofs are omitted for brevity. For any nonnegative symmetric matrix $\Sigma$, we let $\sigma = \text{vec}(\Sigma)$ and use the compact notation $||x||_F^2$ to refer to the squared weighted quantity $x^T \Sigma x$.

**Theorem 1. (Mean-square-error stability)** For sufficiently small step-sizes, i.e., for $\mu_k < \mu_o$ for some small enough $\mu_o$, it holds that $E\bar{w}_i \to 0$, such that the estimates are asymptotically unbiased. Moreover, the weighted error variance satisfies the recursion:

$$E||\bar{w}_1||^2 = E||\bar{w}_{i-1}||^2 + [\text{vec} (Y^T)]^T \sigma$$

where $F \triangleq E(B_1^T \otimes B_1^T)$ with the Kronecker product $\otimes$, $Y \triangleq GS\Sigma^T$, and

$$S \triangleq \text{diag}\{\sigma^2_{v,1} R_{u,1}, \ldots, \sigma^2_{v,N} R_{u,N}\}$$

(39)

5. SIMULATIONS

We consider a network with $N = 10$ agents. Each agent $k$ is estimating a target vector $w_o^k$, which is a subvector of the grand target vector $w_o$ of size $M = 10$. For each agent $k$, we assume that $R_{u,k}$ is diagonal and its diagonal entries are determined by a grand diagonal covariance matrix $R_u$, i.e.,

$$R_{u,k} = D_k R_u D_k^T$$

(40)

Figure 2 shows the entries of the grand target vector $w_o$, the diagonal entries of the grand covariance matrix $R_u$, and the noise variance $\{\sigma^2_{v,k}\}$ at the agents. The network topology and the relations between $\{w_o^k\}$ and $w_o$ are shown in Fig. 3. We set the step-size to $\mu_k = \mu = 0.005$ and the penalty parameter to $\rho_k = 0.005$. In Fig. 3, we simulate the learning curves of instantaneous network mean-square deviation (MSD), which is defined as

$$\text{MSD}_k = \frac{1}{N} \sum_{k=1}^{N} E||w_o^k - w_{k,i}||^2$$

(41)

We observe that both diffusion ATP and PTA algorithms exhibit better steady-state MSD performance than the non-cooperative case without imposing constraints and penalties. To examine the individual performance, we compare the steady-state individual MSD for each agent in Fig. 5. It is seen that all agents benefit from exchange of information with neighbors. The difference between diffusion ATP and PTA algorithms is negligible in the figures.
6. REFERENCES


