# Coordinate-Descent Adaptation over Networks

Chengcheng Wang*†, Yonggang Zhang*, Bicheng Ying‡ and Ali H. Sayed‡
*College of Automation, Harbin Engineering University
†School of Electrical and Electronic Engineering, Nanyang Technological University
‡Department of Electrical Engineering, University of California, Los Angeles

***Abstract*—This work examines the mean-square error performance of diffusion stochastic algorithms under a generalized coordinate-descent scheme. In this setting, the adaptation step by each agent is limited to a random subset of the coordinates of its stochastic gradient vector. The selection of which coordinates to use varies randomly from iteration to iteration and from agent to agent across the network. Such schemes are useful in reducing computational complexity in power-intensive large data applications. The results show that the steady-state performance of the learning strategy is not affected, while the convergence rate suffers some degradation. The results provide yet another indication of the resilience and robustness of adaptive distributed strategies.**

***Index Terms*—Coordinate descent, stochastic partial update, computational complexity, diffusion strategies, stochastic gradient algorithms.**

## I. Introduction and Related Work

Consider a strongly-connected network of $N$ agents, where information can flow in either direction between any two connected agents and, moreover, there is at least one self-loop in the topology [1, p. 436]. We associate a strongly-convex differentiable risk, $J_k(w)$, with each agent $k$ and assume all costs share a common minimizer, $w^o \in \mathbb{R}^M$. This case models important situations where agents work cooperatively towards the same goal. The objective of the network is to determine the unique minimizer $w^o$ of the aggregate cost:

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

It is further assumed that the individual cost functions, $J_k(w)$, are each twice-differentiable and satisfy

$$0 < \nu_d I_M \leq \nabla_w^2 J_k(w) \leq \delta_d I_M \tag{2}$$

where $\nabla_w^2 J_k(w)$ denotes the $M \times M$ Hessian matrix of $J_k(w)$ with respect to $w$, $\nu_d \leq \delta_d$ are positive parameters, and $I_M$ is the $M \times M$ identity matrix. In addition, for matrices $A$ and $B$, the notation $A \leq B$ denotes that $B - A$ is positive semi-definite. The condition in (2) is automatically satisfied by important cases of interest, such as logistic regression or mean-square-error designs [1], [2].

The agents can work cooperatively in an adaptive manner to seek the minimizer $w^o$ of problem (1) by applying the following adapt-then-combine (ATC) form of the diffusion strategy [1], [2]:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,i-1}) & \text{(3a)} \\ \boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i} & \text{(3b)} \end{cases}$$

This implementation has been shown to have superior performance relative to the traditional consensus strategy when used for continuous adaptation and learning with *constant* step-sizes $\mu_k$ [1], [2]. In (3), the vector $\boldsymbol{w}_{k,i}$ denotes the estimate by agent $k$ at iteration $i$ for $w^o$, while $\boldsymbol{\psi}_{k,i}$ is an intermediate estimate. Moreover, an approximation for the true gradient vector of $J_k(w)$, $\widehat{\nabla_{w^\mathsf{T}} J_k}(\cdot)$, is used since it is generally the case that the true gradient vector is not available (e.g., when $J_k(w)$ is defined as the expectation of some loss function and the probability distribution of the data is not known to enable computation of $J_k(\cdot)$ or its gradient vector). The symbol $\mathcal{N}_k$ in (3b) refers to the neighborhood of agent $k$. The coefficients $\{a_{\ell k}\}$ are nonnegative convex combination coefficients that satisfy:

$$a_{\ell k} \geq 0, \;\; \sum_{\ell=1}^{N} a_{\ell k} = 1, \;\; a_{\ell k} = 0, \text{if } \ell \notin \mathcal{N}_k. \tag{4}$$

The main distinction in this work relative to prior studies is that we now assume that, at each iteration $i$, the adaptation step in (3a) has only access to a *random subset* of the entries of the approximate gradient vector. This situation may arise due to missing data or a purposeful desire to reduce the computational burden of the update step. We model this scenario by replacing the approximate gradient vector by

$$\widehat{\nabla_{w^\mathsf{T}} J_k}^{\text{miss}}(\boldsymbol{w}_{k,i-1}) = \boldsymbol{\Gamma}_{k,i} \cdot \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,i-1}) \tag{5}$$

where the random matrix $\boldsymbol{\Gamma}_{k,i}$ is diagonal and consists of Bernoulli random variables $\{\boldsymbol{r}_{k,i}(m)\}$; each of these variables is either zero or one with probability

$$\text{Prob}(\boldsymbol{r}_{k,i}(m) = 0) \triangleq r_k \tag{6}$$

where $0 \leq r_k < 1$ and

$$\boldsymbol{\Gamma}_{k,i} = \text{diag}\{\boldsymbol{r}_{k,i}(1), \boldsymbol{r}_{k,i}(2), \ldots, \boldsymbol{r}_{k,i}(M)\}. \tag{7}$$

In the case when $\boldsymbol{r}_{k,i}(m) = 0$, the $m$-th entry of the gradient vector is missing, and then the $m$-th entry of $\boldsymbol{\psi}_{k,i}$ in (3a) is not updated. Observe that we are attaching two subscripts to $\boldsymbol{r}$: $k$ and $i$, which means that we are allowing the randomness in the update to vary across agents and also over time.

This work was performed while C. Wang was a visiting student at the UCLA Adaptive Systems Laboratory. The work of C. Wang was supported in part by a Chinese Government Scholarship. The work of Y. Zhang was supported in part by the National Natural Science Foundation of China (61371173). The work of B. Ying and A. H. Sayed was supported in part by NSF grants CCF-1524250 and ECCS-1407712.

ISBN 978-0-9928626-7-1 © EURASIP 2017

### A. Relation to Block-Coordinate Descent Methods

If we reduce our formulation (3)–(5) to the single agent case, it will become similar to the randomized block-coordinate descent (RBCD) algorithm [3]–[5] in that the desired cost function is optimized only along a *subset* of the coordinates at each iteration. However, our algorithm offers more randomness in generating the coordinate blocks than the RBCD algorithm, by allowing more random combinations of the coordinates at each time index. Moreover, we are using a random subset of the *stochastic* gradient vector instead of the *true* gradient vector to update the estimate, which is necessary for adaptation and online learning when the true risk function itself is not known. Furthermore, our results consider a general multi-agent scenario involving distributed optimization where *each* individual agent employs random coordinates for its own gradient direction, and these coordinates are generally different from the coordinates used by other agents. In other words, the networked scenario adds significant flexibility into the operation of the agents under model (5).

### B. Relation to Partial Updating Schemes

It is also useful to comment on the differences between our formulation and works that rely on other notions of partial information updates. To begin with, our formulation (5) is different from the models used in [6], [7] where the step-size parameter was modeled as a random Bernoulli variable, $\boldsymbol{\mu}_k(i)$, which could assume the values $\mu_k$ or zero with certain probability. In that case, when the step-size is zero, all entries of $\boldsymbol{w}_{k,i-1}$ will not be updated and adaptation is turned off completely. This is in contrast to the current scenario where only a subset of the entries are left without update and, moreover, this subset varies randomly from one iteration to another. The useful works [8], [9] focus on the special case in which the risks $J_k(w)$ are quadratic in $w$. In [8], it is assumed that only a subset of the weight entries are shared (diffused) among neighbors and that the estimate itself is still updated fully in the adaptation step as shown by (3a). In comparison, the formulation we are considering diffuses all entries of the weight estimates. Similarly, in [9] it is assumed that some entries of the regression vectors are missing, which causes changes to the gradient vectors. In order to undo these changes, an estimation scheme is proposed in [9] to estimate the missing data. In our formulation, more generally, a random subset of the entries of the gradient vector are set to zero at each iteration, while the remaining entries remain unchanged and do not need to be estimated.

There are also other criteria that have been used in the literature to motivate partial updating. For example, in [10], periodic and sequential least-mean-squares (LMS) algorithms are proposed. In [11], [12] the weight vectors are partially updated by following a set-membership approach, where updates occur only when the *innovation* obtained from the data exceeds a predetermined threshold. In [12], [13], only entries corresponding to the largest magnitudes in the regression vector or the gradient vector at each agent are updated. However, such scheduled updating techniques can suffer from non-convergence in the presence of nonstationary signals [14]. Partial update schemes can also be based on dimensionality reduction policies using Krylov subspace concepts [15]–[17]. There are also techniques that rely on energy considerations to limit updates, e.g., [18].

### C. This Work

The objective of the analysis that follows is to examine the effect of *random* partial gradient information on the learning performance and convergence rate of adaptive networks for general risk functions. We clarify these questions by adapting the framework developed in [1], [2]. Note that the main difference between the current work and the prior work in [1] is the appearance of the random matrices $\{\boldsymbol{\Gamma}_{k,i}\}$ defined by (5). In the special case when the random matrices are set to the identity matrices across the agents, i.e., $\{\boldsymbol{\Gamma}_{k,i} \equiv I_M\}$, the current coordinate-descent case will reduce to the full-gradient update studied in [1]. The inclusion of the random matrices $\{\boldsymbol{\Gamma}_{k,i}\}$ adds a non-trivial level of complication because now, agents update only random entries of their iterates at each iteration and, importantly, these entries vary randomly across the agents.

## II. Data Model and Assumptions

Let $\boldsymbol{\mathcal{F}}_{i-1}$ represent the filtration of all random events generated by the processes $\{\boldsymbol{w}_{k,j}\}$ and $\{\boldsymbol{\Gamma}_{k,j}\}$ at all agents up to time $i-1$. In effect, the notation $\boldsymbol{\mathcal{F}}_{i-1}$ refers to the collection of all past $\{\boldsymbol{w}_{k,j}, \boldsymbol{\Gamma}_{k,j}\}$ for all $j \leq i-1$ and all $k$.

*Assumption 1:* (**Conditions on indicator variables**). It is assumed that the indicator variables $\boldsymbol{r}_{k,i}(m)$ and $\boldsymbol{r}_{\ell,i}(n)$ are independent of each other, for all $\ell, k, m, n$. In addition, the variables $\{\boldsymbol{r}_{k,i}(m)\}$ are independent of $\boldsymbol{\mathcal{F}}_{i-1}$ and $\widehat{\nabla_{w^{\mathsf{T}}} J_k}(\boldsymbol{w})$ for any iterates $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$ and for all agents $k$. ■

Let

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) \triangleq \widehat{\nabla_{w^{\mathsf{T}}} J_k}(\boldsymbol{w}_{k,i-1}) - \nabla_{w^{\mathsf{T}}} J_k(\boldsymbol{w}_{k,i-1}) \quad (8)$$

denote the gradient noise at agent $k$ at iteration $i$, based on the *complete* approximate gradient vector, $\widehat{\nabla_{w^{\mathsf{T}}} J_k}(\boldsymbol{w})$. We introduce its conditional second-order moment:

$$\boldsymbol{R}_{s,k,i}(\boldsymbol{w}) \triangleq \mathbb{E}[\boldsymbol{s}_{k,i}(\boldsymbol{w})\boldsymbol{s}_{k,i}^{\mathsf{T}}(\boldsymbol{w})|\boldsymbol{\mathcal{F}}_{i-1}]. \quad (9)$$

The following assumptions are standard and are satisfied by important cases of interest, such as logistic regression risks or mean-square-error risks, as already shown in [1], [2].

*Assumption 2:* (**Conditions on gradient noise**) [1, pp. 496–497]. It is assumed that the first and fourth-order conditional moments of the individual gradient noise processes satisfy the following conditions for any iterates $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$ and for all $k, \ell = 1, 2, \ldots, N$:

$$\mathbb{E}[\boldsymbol{s}_{k,i}(\boldsymbol{w})|\boldsymbol{\mathcal{F}}_{i-1}] = 0 \quad (10)$$

$$\mathbb{E}[\boldsymbol{s}_{k,i}(\boldsymbol{w})\boldsymbol{s}_{\ell,i}^{\mathsf{T}}(\boldsymbol{w})|\boldsymbol{\mathcal{F}}_{i-1}] = 0, \; k \neq \ell \quad (11)$$

$$\mathbb{E}[\|\boldsymbol{s}_{k,i}(\boldsymbol{w})\|^4|\boldsymbol{\mathcal{F}}_{i-1}] \leq \beta_k^4 \|\boldsymbol{w}\|^4 + \sigma_{s,k}^4 \quad (12)$$

almost surely, for some nonnegative scalars $\beta_k^4$ and $\sigma_{s,k}^4$. ■

*Assumption 3:* (**Smoothness conditions**) [1, pp. 552,576]. It is assumed that the Hessian matrix of each individual cost

function, $J_k(w)$, and the covariance matrix of each individual gradient noise process are locally Lipschitz continuous in a small neighborhood around $w = w^o$ in the following manner:

$$\|\nabla_w^2 J_k(w^o + \triangle w) - \nabla_w^2 J_k(w^o)\| \leq \kappa_c \|\triangle w\| \quad (13)$$

$$\|\boldsymbol{R}_{s,k,i}(w^o + \triangle w) - \boldsymbol{R}_{s,k,i}(w^o)\| \leq \kappa_d \|\triangle w\|^\gamma \quad (14)$$

for any small perturbations $\|\triangle w\| \leq \varepsilon$ and for some $\kappa_c \geq 0$, $\kappa_d \geq 0$, and parameter $0 < \gamma \leq 4$. In addition, the notation $\|\cdot\|$ denotes the two-induced norm of a matrix or the Euclidean norm of a vector. ■

## III. Main Results: Stability and Performance

In this and the following sections, we only state the main results due to space limitations. Detailed derivations appear in [19].

*Theorem 1:* (**Network stability**). The second-order and fourth-order moments of the network error vectors $\{\widetilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i}\}$ are stable (bounded) for sufficiently small step-sizes, namely, there exists a small enough $\mu_o$ such that:

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 = O(\mu_{\max}) \quad (15)$$

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^4 = O(\mu_{\max}^2) \quad (16)$$

for any $\mu_{\max} < \mu_o$, where $\mu_{\max} \triangleq \max\{\mu_1, \mu_2, \ldots, \mu_N\}$. ■

In (15)–(16), the notation $\alpha = O(\mu)$ means that $|\alpha| \leq c|\mu|$ for some constant $c > 0$. Result (15) ensures that the mean-square-error (MSE) performance of the network is on the order of $\mu_{\max}$. We can be more explicit and assess the proportionality constant that determines the value of the network mean-square-error to first-order in $\mu_{\max}$. To do so, we first introduce some useful variables. Since the network is strongly-connected, then the combination matrix $A = [a_{\ell k}]$ is primitive. This means, in view of the Perron-Frobenius Theorem [1], [2], that $A$ has a single eigenvalue at one. We denote the corresponding eigenvector by $p$, with entries $p_k$. We normalize the entries of $p$ to add up to one and note that all entries $p_k$ are strictly positive:

$$Ap = p, \;\; \mathbb{1}^{\mathsf{T}} p = 1, \;\; p_k > 0. \quad (17)$$

In (17), the notation $\mathbb{1}$ refers to the vector of size $N$ with all its entries equal to one. We introduce the vector $q = \text{col}\{q_k\}$

$$q \triangleq \text{col}\{\mu_1 p_1, \mu_2 p_2, \ldots, \mu_N p_N\} \quad (18)$$

where $\text{col}\{\cdot\}$ denotes a column vector, and the Hessian matrix of $J_k(w)$ evaluated at $w = w^o$

$$H_k \triangleq \nabla_w^2 J_k(w^o). \quad (19)$$

We also introduce the gradient-noise covariance matrices:

$$G_k \triangleq \lim_{i\to\infty} \boldsymbol{R}_{s,k,i}(w^o) \quad (20)$$

$$G'_k \triangleq \mathbb{E}[\boldsymbol{\Gamma}_{k,i} G_k \boldsymbol{\Gamma}_{k,i}]. \quad (21)$$

Observe that $G_k$ is the limiting covariance matrix of the gradient noise process, while $G'_k$ is a weighted version of it. It can be verified by direct inspection that the entries of $G'_k$ are given by:

$$G'_k(m,n) = \begin{cases} (1-r_k)^2 G_k(m,n), & m \neq n \\ (1-r_k) G_k(m,m), & m = n. \end{cases} \quad (22)$$

Let $\text{MSD}_k$ denote the size of the steady-state mean-square-deviation, $\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, to first-order in $\mu_{\max}$, and let $\text{MSD}_{av}$ denote the average $\text{MSD}_k$ value across all $N$ agents — see [1, p. 582] for expressions and further clarifications. Moreover, we define the convergence rate as the slowest rate at which the error variances, $\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, converge to the steady-state region — see [1, p. 395] for expressions and further clarifications.

*Theorem 2:* (**Network limiting performance**). It holds that, for sufficiently small step-sizes:

$$\text{MSD}_{\text{coor},k} = \text{MSD}_{\text{coor,av}} = \frac{1}{2}\text{Tr}\left(\left(\sum_{k=1}^N q_k(1-r_k)H_k\right)^{-1} \sum_{k=1}^N q_k^2 G'_k\right) \quad (23)$$

where the subscript "coor" denotes the stochastic coordinate-descent diffusion implementation. Moreover, for large enough $i$, the convergence rate of the error variances, $\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, towards the steady-state region (23) is given by

$$\alpha_{\text{coor}} = 1 - 2\lambda_{\min}\left(\sum_{k=1}^N q_k(1-r_k)H_k\right) + O\left(\mu_{\max}^{(N+1)/N}\right) \quad (24)$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue. ■

## IV. Implications and Useful Cases

Consider the case when the missing probabilities are identical across the agents, i.e., $\{r_k \equiv r\}$.

### A. Convergence Time

Consider the full-gradient or coordinate-descent diffusion strategy (3a)–(3b) and (5). Let $T_{\text{grad}}$ and $T_{\text{coor}}$ denote the largest number of iterations that are needed for the error variances, $\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$, to converge to their steady-state regions.

*Corollary 1:* (**Convergence time**). It holds that, for sufficiently small step-sizes:

$$1 \leq \frac{T_{\text{coor}}}{T_{\text{grad}}} \approx \frac{1}{1-r}. \quad (25)$$

■

It follows that the coordinate-descent implementation converges at a slower rate as expected (since it only employs partial gradient information).

### B. Computational Complexity

Assume that the computation required to calculate each entry of the gradient vector $\widehat{\nabla_{w^{\mathsf{T}}} J_k}(\boldsymbol{w}_{k,i-1})$ is identical, and let $c_m \geq 0$ denote the number of multiplications that are needed for each entry. Let $n_k \triangleq |\mathcal{N}_k|$ denote the degree of agent $k$, $M_{\text{grad},k}$ and $M_{\text{coor},k}$ denote the total number of multiplications at agent $k$ for the full-gradient and coordinate-descent implementations, respectively.

ISBN 978-0-9928626-7-1 © EURASIP 2017

*Corollary 2:* (**Computational complexity**). It holds that, for sufficiently small step-sizes:

$$1 \leq \frac{M_{\mathrm{coor},k}}{M_{\mathrm{grad},k}} = (1-r)^{-1}\left(1 - \frac{c_m+1}{c_m+n_k+1}r\right). \quad (26)$$

It is clear that when it is costly to compute the gradient entries, i.e., when $c_m \gg n_k$, then $M_{\mathrm{coor},k}$ and $M_{\mathrm{grad},k}$ will be essentially identical. A similar analysis and conclusion holds if we examine the total number of additions (as opposed to multiplications). ■

Corollaries 1 and 2 show that while the coordinate-descent implementation will take longer to converge, the savings in computation per iteration that it provides is such that the overall computational complexity until convergence remains largely invariant. This is a useful conclusion. It means that in situations where computations at each iteration need to be minimal, then a coordinate-descent variant is recommended and it will be able to deliver the same steady-state performance (to first-order in $\mu_{\max}$, see (31) ahead) with the total computational demand spread over a longer number of iterations.

### C. MSD performance

Note that the MSD value for the stochastic full-gradient diffusion implementation, $\mathrm{MSD}_{\mathrm{grad},k}$, can be evaluated from the expression (23) by setting the missing probabilities $\{r_k \equiv 0\}$ [1, p. 594]. In that case, the matrix $G'_k$ defined in (21) will be replaced by $G_k$ defined in (20), since the random matrices $\{\boldsymbol{\Gamma}_{k,i}\}$ will reduce to $\{\boldsymbol{\Gamma}_{k,i} \equiv I_M\}$. Then, it holds that

$$\mathrm{MSD}_{\mathrm{coor},k} - \mathrm{MSD}_{\mathrm{grad},k} = \frac{r}{2}\mathrm{Tr}\left(\left(\sum_{k=1}^{N} q_k H_k\right)^{-1} \sum_{k=1}^{N} q_k^2 \check{G}_k\right) \quad (27)$$

where

$$\check{G}_k \triangleq \mathrm{diag}\{G_k\} - G_k \quad (28)$$

with the term $\mathrm{diag}\{G_k\}$ being a diagonal matrix that consists of the diagonal entries of $G_k$. It follows by direct inspection that in the case when the matrices $\{H_k\}$ or $\{G_k\}$ are diagonal, we have

$$\mathrm{MSD}_{\mathrm{coor},k} = \mathrm{MSD}_{\mathrm{grad},k}. \quad (29)$$

We show in [19] that the difference in (27) can be positive or negative, i.e., the MSD performance can be better or worse in the stochastic coordinate-descent case in comparison to the stochastic full-gradient case. In addition, we are able to provide a general upper bound for that MSD gap as shown by Corollary 2 of [19]. Recall that the MSD performance is evaluated to first-order in $\mu_{\max}$. Then, it follows from (27):

$$\mathrm{Tr}\left(\left(\sum_{k=1}^{N} q_k H_k\right)^{-1} \sum_{k=1}^{N} q_k^2 \check{G}_k\right) = O(\mu_{\max}). \quad (30)$$

Moreover, the difference between $\mathrm{MSD}_{\mathrm{coor},k}$ and $\mathrm{MSD}_{\mathrm{grad},k}$ is linearly dependent on the missing probability $r$. Then, the MSD gap in (27) can be decreased by using small missing probabilities across the agents.

*Corollary 3:* (**Small missing probabilities**). Let $r = O(\mu_{\max}^{\varepsilon})$ for a small number $\varepsilon > 0$. It holds that

$$\mathrm{MSD}_{\mathrm{coor},k} - \mathrm{MSD}_{\mathrm{grad},k} = O(\mu_{\max}^{1+\varepsilon}) = o(\mu_{\max}) \quad (31)$$

where $\alpha = o(\mu)$ signifies that $\alpha/\mu \to 0$ as $\mu \to 0$. ■

Corollary 3 shows that in the case of small missing probabilities, the steady-state MSD levels of the stochastic coordinate-descent and full-gradient diffusion cases will be the same to first-order in $\mu_{\max}$.

### D. MSE Networks

Consider MSE networks where the risk function that is associated with each agent $k$ is the mean-square-error [1], [2]:

$$J_k(w) = \mathbb{E}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w)^2 \quad (32)$$

where the scalar $\boldsymbol{d}_k(i)$ denotes the desired signal, and $\boldsymbol{u}_{k,i}$ is a (row) regression vector. In these networks, the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ are assumed to be related via the linear regression model

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i) \quad (33)$$

where $\boldsymbol{v}_k(i)$ is zero-mean white measurement noise with variance $\sigma_{v,k}^2$ and assumed to be independent of all other random variables. Assume also that the regression data $\{\boldsymbol{u}_{k,i}\}$ are zero-mean, white over time and space with

$$\mathbb{E}\,\boldsymbol{u}_{k,i}^{\mathsf{T}}\boldsymbol{u}_{\ell,j} \triangleq R_{u,k}\delta_{k,\ell}\delta_{i,j} \quad (34)$$

where $R_{u,k} > 0$, and $\delta_{k,\ell}$ denotes the Kronecker delta sequence. Consider the case when the covariance matrices of the regressors are identical across the network, i.e., $\{R_{u,k} \equiv R_u > 0\}$. Then, it holds that [1, p. 598]:

$$H_k \equiv 2R_u,\ G_k = 4\sigma_{v,k}^2 R_u. \quad (35)$$

In the case of MSE networks, by exploiting the special relation between the matrices $\{H_k\}$ and $\{G_k\}$ in (35), we are able to show that the MSD in the stochastic coordinate-descent case is always larger (i.e., worse) than or equal to that in the stochastic full-gradient diffusion case (although by not more than $o(\mu_{\max})$, as indicated by (31)). We are also able to provide a general upper bound on the MSD gap.

*Corollary 4:* (**MSE networks**). For MSE networks with uniform regression covariance matrices, i.e., $\{R_{u,k} \equiv R_u > 0\}$, it holds that, for sufficiently small step-sizes:

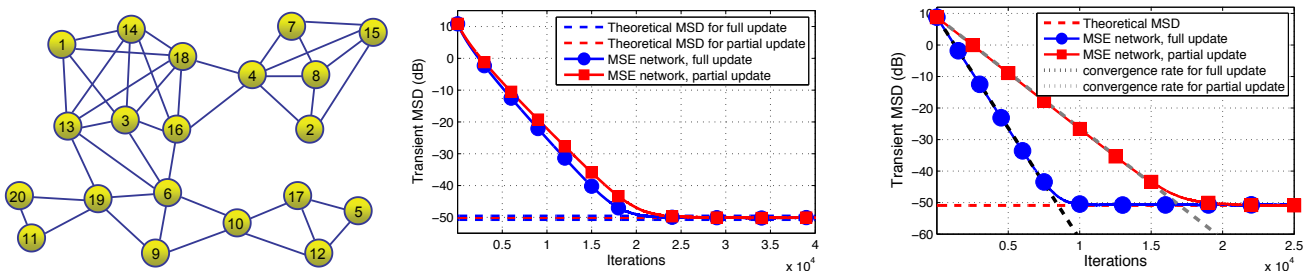
$$0 \leq \mathrm{MSD}_{\mathrm{coor},k} - \mathrm{MSD}_{\mathrm{grad},k} \leq r\left(\sum_{k=1}^{N} q_k\right)^{-1}\left(\sum_{k=1}^{N} q_k^2\sigma_{v,k}^2\right)\left(\frac{\delta_d}{\nu_d} - 1\right)M \quad (36)$$

Moreover, it holds that $\mathrm{MSD}_{\mathrm{coor},k} = \mathrm{MSD}_{\mathrm{grad},k}$ if, and only if, $R_u$ is diagonal in view of (28) and (35). ■

## V. Simulation Results

In this section, we illustrate the results by considering MSE networks, which satisfy condition (2) and Assumptions 1 through 3. The performance of the algorithms is tested in the case when uniform missing probabilities are utilized across the

ISBN 978-0-9928626-7-1 © EURASIP 2017




(a) (b) (c)

Fig. 1. (a) Network topology consisting of $N = 20$ agents. (b) MSD learning curves, averaged over 1000 independent runs, in the case of Corollary 3 when $\{r_k \equiv 0.1\}$. The dashed lines show the theoretical MSD values from (23). (c) MSD learning curves, averaged over 1000 independent runs, in the case of Corollary 4 when the regressors are white. The dashed line along the horizontal axis shows the theoretical MSD value from (23). Those along the learning curves show the reference recursion at rates formulated by (24).

agents. Figure 1(a) shows a network topology with $N = 20$ agents. In the first example, we test the case when the gradient vectors are missing with small probabilities $\{r_k \equiv 0.1\}$ across the agents. The combination matrix $A$ is doubly-stochastic and set according to the Metropolis rule in [1, p. 664]. The parameter vector $w^o$ is randomly generated with $M = 10$. The regressors are generated by the first-order autoregressive model

$$\boldsymbol{u}_{k,i}(m) = \pi_k \boldsymbol{u}_{k,i}(m-1) + \sqrt{1-\pi_k^2}\,\boldsymbol{t}_{k,i}(m) \qquad (37)$$

for any $1 \leq m < M$, and the variances are scaled to be 1. The processes $\{\boldsymbol{t}_{k,i}\}$ are zero-mean, unit-variance, and independent and identically distributed (*i.i.d*) Gaussian sequences. The $\{\pi_k\}$ are generated from a uniform distribution on the interval $(-1, 1)$. The noises, uncorrelated with the regression vectors, are zero-mean white Gaussian sequences with the variances uniformly distributed over $(0.001, 0.1)$. The step-sizes $\{\mu_k\}$ across the agents are generated from a uniform distribution on the interval $(1 \times 10^{-4}, 5 \times 10^{-4})$. Figure 1(b) shows the simulation results, which are averaged over 1000 independent runs. It is clear from the figure that, when the gradient information is missing with small probabilities, the performance of the coordinate-descent case is close to that of the full-gradient diffusion case.

In the second example, we test the case when the regressors are white across the agents. We randomly generate $w^o$ of size $M = 6$. The white regressors are generated from zero-mean white Gaussian sequences, and the powers, which vary from entry to entry, and from agent to agent, are uniformly distributed over $(0.5, 1.5)$. The step-sizes are uniformly distributed over $(1\times10^{-4}, 8\times10^{-4})$. The results, including the theoretical MSD value from (23) in Theorem 2, the simulated MSD learning curves, and the reference recursion at rates from (24), are illustrated by Fig. 1(c), where the results are averaged over 1000 independent runs. It is clear from the figure that, when white regressors are utilized in MSE networks, the stochastic coordinate-descent case converges to the same MSD level as the full-gradient diffusion case, which verifies (29), at a convergence rate formulated in (24).

## REFERENCES


[1] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014. [Online]. Available: http://dx.doi.org/10.1561/2200000051

[2] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[3] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[4] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, vol. 144, no. 1-2, pp. 1–38, 2014.

[5] Z. Lu and L. Xiao, "On the complexity analysis of randomized block-coordinate descent methods," *Mathematical Programming*, vol. 152, pp. 615–642, 2015.

[6] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks-Part I: Modeling and stability analysis," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 811–826, Feb. 2015.

[7] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks-Part II: Performance analysis," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 827–842, Feb. 2015.

[8] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 472–484, Jan. 2014.

[9] M. R. Gholami, E. G. Ström, and A. H. Sayed, "Diffusion estimation over cooperative networks with missing data," in *Proc. IEEE GlobalSIP*, Austin, TX, Dec. 2013, pp. 411–414.

[10] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Trans. Circuits Syst. II*, vol. 44, no. 3, pp. 209–216, Mar. 1997.

[11] S. Werner, M. Mohammed, Y.-F. Huang, and V. Koivunen, "Decentralized set-membership adaptive estimation for clustered sensor networks," in *Proc. IEEE ICASSP*, Las Vegas, NV, 2008, pp. 3573–3576.

[12] S. Werner and Y.-F. Huang, "Time- and coefficient- selective diffusion strategies for distributed parameter estimation," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, 2010, pp. 696–700.

[13] K. Doğancay, O. Tanrıkulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Trans. Circuits Syst. II*, vol. 48, no. 8, pp. 762–769, Aug. 2001.

[14] M. Godavarti and A. O. Hero, "Partial update LMS algorithms," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2382–2399, Jul. 2005.

[15] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction," *IEEE Journal Selected Topics Signal Process.*, vol. 7, no. 2, pp. 257–273, April 2013.

[16] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections: A unifying framework for linear and nonlinear classification and regression tasks," *IEEE Signal Process. Magazine*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[17] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.

[18] O. N. Gharehshiran, V. Krishnamurthy, and G. Yin, "Distributed energy-aware diffusion least mean squares: Game-theoretic learning," *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 5, pp. 821–836, Oct. 2013.

[19] C. Wang, Y. Zhang, B. Ying, and A. H. Sayed, "Coordinate-descent diffusion learning by networked agents," *submitted for publication.* Also available as *arXiv:1607.01838*, Jul. 2016.


ISBN 978-0-9928626-7-1 © EURASIP 2017