PROXIMAL DIFFUSION FOR STOCHASTIC COSTS WITH NON-DIFFERENTIABLE REGULARIZERS

Stefan Vlaski and Ali H. Sayed

Department of Electrical Engineering
University of California, Los Angeles

ABSTRACT

We consider networks of agents cooperating to minimize a global objective, modeled as the aggregate sum of regularized costs that are not required to be differentiable. Since the subgradients of the individual costs cannot generally be assumed to be uniformly bounded, general distributed subgradient techniques are not applicable to these problems. We isolate the requirement of bounded subgradients into the regularizer and use splitting techniques to develop a stochastic proximal diffusion strategy for solving the optimization problem by continuously learning from streaming data. We represent the implementation as the cascade of three operators and invoke Banach's fixed-point theorem to establish that, despite gradient noise, the stochastic implementation is able to converge in the mean-square-error sense within $O(\mu)$ from the optimal solution, for a sufficiently small step-size parameter, μ.

Index Terms— Distributed optimization, diffusion strategy, proximal operator, gradient noise, fixed point, regularization.

1. INTRODUCTION AND RELATED WORK

We consider a network of N agents, where each agent k is equipped with a risk $J_k(w)$, which is the expectation of some loss function and written as $J_k(w) = \mathbb{E}Q_k(w)$. The individual agents run a distributed strategy, such as consensus [1–4] or diffusion [5–7], to compute estimates for the global minimizer of some aggregate cost function specified further ahead in (4). It is sufficient for the purposes of this work to focus on the case in which each agent k runs the following Adapt-then-Combine form of diffusion [5]:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1}) \tag{1a}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^{N} a_{\ell k}\boldsymbol{\phi}_{\ell,i} \tag{1b}$$

where the $\{a_{\ell k}\}$ are convex combination coefficients that satisfy

$$a_{\ell k} \geq 0, \quad \sum_{\ell\in\mathcal{N}_k} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{2}$$

with $\mathcal{N}_k$ denoting the neighborhood of agent k. In (1b), the symbol $\boldsymbol{w}_{k,i}$ denotes the iterate that is computed by agent k at iteration i, while $\boldsymbol{\psi}_{k,i}$ is an intermediate state resulting from the self-learning step (1a). Moreover, the notation $\widehat{\nabla_w J}_k(\cdot)$ denotes a stochastic approximation for the true gradient vector of $J_k(\cdot)$, which is generally unknown, since the statistical properties of the data are not assumed to be known in the adaptive context. The difference between the true gradient of $J_k(\cdot)$ and its approximation is called gradient noise. In (1a)–(1b), we are using boldface letters for the variables $\{\boldsymbol{w}_{k,i}, \boldsymbol{\phi}_{k,i}\}$ to highlight the fact that they are random quantities; we will be using normal font to represent deterministic quantities.

When the network is strongly-connected (i.e. connected with at least one self-loop), the left-stochastic combination matrix $A = [a_{\ell k}]$ will be primitive with a single eigenvalue at one, while all other eigenvalues will lie strictly inside the unit circle [5, 8, 9]. We denote the left and right eigenvectors of A that are associated with the eigenvalue at one by:

$$\mathbb{1}^{\mathsf{T}}A = \mathbb{1}^{\mathsf{T}}, \quad Ap = p \tag{3}$$

and normalize the entries of p to add up to one. It follows from the Perron-Frobenius Theorem [8, 9] that all entries of p are strictly positive. We denote the individual entries of p by $\{p_k\}$. It is shown in [5, 6] that, under some reasonable technical conditions on the cost functions and gradient noise, the iterate $\boldsymbol{w}_{k,i}$ by each agent k converges in the mean-square sense to the unique minimizer, w^o, of the following weighted aggregate cost:

$$w^o = \arg\min_{w} \sum_{k=1}^{N} p_k J_k(w) \tag{4}$$

within $O(\mu)$, namely,

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu) \tag{5}$$

so that all agents are able to approach the same global minimizer for a sufficiently small step-size.

1.1. Regularized Costs

In many situations, there is prior information available about w^o (such as knowing that w^o is sparse, or that it is constrained to a certain region in space, or that it is close to some value). One way to exploit this information is to employ regularization to favor solutions with the desired properties. We therefore modify the cost function for every agent k as follows:

$$J_k^{\text{reg}}(w) \triangleq J_k(w) + \delta\mu^{\nu}R_k^{\text{org}}(w) \triangleq J_k(w) + R_k(w) \tag{6}$$

where $\{\delta, \nu\}$ are non-negative parameters and the regularization function $R_k^{\text{org}}(\cdot)$ does not need to be differentiable. Note that we allow for two parameters: δ represents absolute scaling and is generally chosen small, while the exponent $\nu \geq 0$ allows the scaling to depend on the step-size parameter of the algorithm.

Some useful distributed subgradient techniques are developed in [3]. However, these techniques require the subgradients of each cost $J_k^{\text{reg}}(w)$ to be uniformly bounded for all w and they do not exploit the composite structure of (6). As such, they are not applicable

This work was supported in part by NSF grants CCF-1011918 and ECCS-1407712. Emails: {svlaski, sayed}@ucla.edu

to several situations of interest (even those involving mean-square-error costs). References [10–13] provide variations for such problems using diffusion strategies for the special case of mean-square-error costs.

The main purpose of this article is to develop a distributed strategy that is applicable to cost functions of the form (6) where only the regularizer's subgradient is required to be uniformly bounded. This situation is satisfied in many cases of interest, such as in ℓ_1-regularization and variations thereof. We achieve this objective by relying on the use of splitting techniques to propose an extension of the diffusion strategy for regularized aggregate costs. Splitting techniques are common in the *deterministic* optimization literature [14–19], where it is assumed that the individual costs are known beforehand so that their gradients and/or proximal projections can be computed precisely. This is rarely the case in adaptive environments. The reason is that the expectation of the loss functions cannot be computed beforehand because the statistical distribution of the data is rarely known. Only data realizations are available. We will explain how a *stochastic* approximation variant can be motivated and then examine the impact of gradient noise on the learning ability of the resulting distributed solution. The key conclusion will be that the proposed proximal diffusion strategy is able to approach the global minimizer, w^o, with a mean-square-error that is sufficiently small and within $O(\mu)$.

2. PROXIMAL DIFFUSION STRATEGY

To begin with, we recall that, in the purely deterministic context, the proximal operator relative to $R_k(\cdot)$ with step-size μ is defined by [17]:

$$\text{prox}_{\mu R_k}(x) \triangleq \arg\min_u \left(R_k(u) + \frac{1}{2\mu}\|x-u\|_2^2 \right) \tag{7}$$

Evaluating Eq. (7) at $x = w_{k,i-1} - \mu\nabla_w J_k(w_{k,i-1})$, which is the result of a gradient-descent step applied to $J_k(w)$, yields the proximal gradient descent iteration:

$$w_{k,i} = \text{prox}_{\mu R_k}\left\{w_{k,i-1} - \mu\nabla_w J_k(w_{k,i-1})\right\} \tag{8}$$

From the optimality condition for Eq. (7), namely that the subgradient set at the minimizer contains the zero-vector, it follows that [17, 19]:

$$w_{k,i} \in w_{k,i-1} - \mu\nabla_w J_k(w_{k,i-1}) - \mu\, \partial_w R_k\,(w_{k,i}) \tag{9}$$

where $\partial_w R_k\,(w_{k,i})$ denotes the set of subgradients of $R_k(w)$ at $w_{k,i}$. The proximal operation (8) returns a particular subgradient vector, which we denote by $\widehat{\partial_w R_k}(w_{k,i})$. In this way, the resulting iterate from (9) can be written as

$$w_{k,i} = w_{k,i-1} - \mu\nabla_w J_k(w_{k,i-1}) - \mu\,\widehat{\partial_w R_k}\,(w_{k,i}) \tag{10}$$

Observe from (9) and (10) that $\nabla_w J_k(\cdot)$ is evaluated at $w_{k,i-1}$, whereas $\partial_w R_k(\cdot)$ is evaluated at $w_{k,i}$. This property sometimes motivates the alternative designation "forward-backward" operator for the proximal gradient step. Proximal gradient descent is of particular interest when (7) can be evaluated efficiently or even in closed form – see [14] for an overview of closed form solutions of (7) for particular $R_k(\cdot)$. In the case of the ℓ_1-norm, for example, the proximal operator reduces to soft-thresholding [18, 20].

Some recent studies examine the performance of *inexact* proximal methods for particular sources of uncertainties in the gradient information. For example, the work in [21] considers inexact proximal gradient descent where the errors in the computation of the gradient and/or proximal operator are assumed to be deterministic and decay to zero. The work [22] builds on this analysis and develops a fast distributed implementation that enforces agreement among agents by embedding i communication steps between iterations i and $i+1$ and letting $i \to \infty$. This construction can be reasonable in the deterministic context, where a given accuracy can be tolerated after finite time i, but is infeasible in the context of continuous adaptation and learning from streaming data since it will require the number of communication steps to grow unbounded. In [23] regret bounds for stochastic proximal subgradient descent are derived under the assumption of Lipschitz continuous costs; the bounds there were limited to a single-agent implementation.

Returning to (1a)–(1b), the above discussion motivates us to introduce the following proximal implementation of diffusion:

$$\boldsymbol{\phi}_{k,i} = \text{prox}_{\mu R_k}\left\{\boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1})\right\} \tag{11a}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^{N} a_{\ell k}\boldsymbol{\phi}_{\ell,i} \tag{11b}$$

where a proximal step has been added to (1a) as shown by (11a). This adjustment is meant to address the presence of the regularization term added in (6). Observe that (11a)–(11b) responds immediately to streaming data; it does not require repeated iterations between two successive time instants. We will further see that this implementation does also not require the gradient noise to be deterministic or to decay to zero.

The analysis in the subsequent sections will establish the following facts about the stochastic implementation (11a)–(11b):

- In Section 4.1, it will be shown that, when the true gradient vectors are employed in (11a), then each agent in the diffusion strategy will converge to a unique fixed point, denoted by $w_{k,\infty}$.
- In Section 4.2, we will relate $w_{k,\infty}$ to the global minimizer w^o of (4) and show that $\|w^o - w_{k,\infty}\|^2 \leq O(\mu^{2\nu}) + O(\mu^2)$.
- In Section 4.3, we will conclude that, for $\nu \geq 1/2$, recursion (11a)–(11b) *with* gradient noise converges to w^o within $O(\mu)$ in the mean-square-error sense.

Due to space limitations, proofs and derivations are omitted. We focus on highlighting the conclusions and their interpretations. The following two assumptions are needed in establishing the results – see [5] for explanations and motivation.

Assumption 1 (Bounded Hessian)**.** *For any k, the Hessian matrix function, $H_k(w) = \nabla_w^2 J_k(w)$, is assumed to be uniformly bounded from below and from above:*

$$0 < \lambda_{\min} I_N \leq H_k(w) \leq \lambda_{\max} I_N \tag{12}$$

Assumption 2 (Gradient Noise Process)**.** *For any k, the gradient noise process is defined as*

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) = \widehat{\nabla_w J}_k(\boldsymbol{w}_{k,i-1}) - \nabla_w J_k(\boldsymbol{w}_{k,i-1}) \tag{13}$$

and satisfies

$$\mathbb{E}\left[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})|\boldsymbol{\mathcal{F}}_{i-1}\right] = 0 \tag{14a}$$

$$\mathbb{E}\left[\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^2|\boldsymbol{\mathcal{F}}_{i-1}\right] \leq \beta^2\|\boldsymbol{w}_{k,i-1}\|^2 + \sigma_s^2 \tag{14b}$$

for some non-negative constants $\{\beta^2, \sigma_s^2\}$, and where $\boldsymbol{\mathcal{F}}_{i-1}$ denotes the filtration generated by the random processes $\{\boldsymbol{w}_{\ell,j}\}$ for all $\ell = 1, 2, \ldots, N$ and $j \leq i-1$, i.e., $\boldsymbol{\mathcal{F}}_{i-1}$ represents the information that is available about the random processes $\{\boldsymbol{w}_{\ell,j}\}$ up to time $i-1$.

3. OPERATOR REPRESENTATION OF PROXIMAL DIFFUSION

We first show that the proximal diffusion strategy (11a)–(11b) can be represented as the concatenation of three operators, in a manner that extends the representation developed in [7] for the conventional diffusion iteration without proximal steps. We subsequently show that this mapping is contractive and invoke Banach's fixed-point theorem [24] to conclude that the proximal diffusion mapping has a unique fixed-point. We first introduce some notation and definitions. Thus, let

$$x = \mathrm{col}\{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{MN} \tag{15}$$

denote an $N \times 1$ block-column vector, where each x_k is $M \times 1$.

Definition 1. (Combination Operator) The combination operator $T_A : \mathbb{R}^{MN} \to \mathbb{R}^{MN}$ is defined as the linear mapping:

$$T_A(x) \triangleq (A^{\mathsf{T}} \otimes I_M)x = \mathrm{col}\left\{\sum_{\ell=1}^{N} a_{\ell k} x_\ell\right\} \tag{16}$$

where $A = [a_{\ell k}]$ is an $N \times N$ left-stochastic matrix and $\otimes$ denotes the Kronecker product operation. □

Definition 2. (Block Gradient Descent Operator) The block gradient descent operator $T_G : \mathbb{R}^{MN} \to \mathbb{R}^{MN}$ is defined as the non-linear mapping:

$$T_G(x) \triangleq \begin{bmatrix} x_1 - \mu\nabla_w J_1(x_1) \\ \vdots \\ x_N - \mu\nabla_w J_N(x_N) \end{bmatrix} \tag{17}$$

□

Definition 3. (Stochastic Block Gradient Descent Operator) The stochastic block gradient descent operator $\widehat{\boldsymbol{T}}_G : \mathbb{R}^{MN} \to \mathbb{R}^{MN}$ is defined as the non-linear mapping:

$$\widehat{\boldsymbol{T}}_G(\boldsymbol{x}) \triangleq \begin{bmatrix} \boldsymbol{x}_1 - \mu\widehat{\nabla_w J}_1(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{x}_N - \mu\widehat{\nabla_w J}_N(\boldsymbol{x}_N) \end{bmatrix} = T_G(\boldsymbol{x}) + \mu\boldsymbol{s}(\boldsymbol{x}) \tag{18}$$

where

$$\boldsymbol{s}(\boldsymbol{x}) \triangleq \mathrm{col}\{\boldsymbol{s}_1(\boldsymbol{x}_1), \ldots, \boldsymbol{s}_N(\boldsymbol{x}_N)\} \tag{19}$$

is the (block) gradient noise vector. □

Definition 4. (Block Proximal Operator) The block proximal operator $T_P : \mathbb{R}^{MN} \to \mathbb{R}^{MN}$ is defined as the non-linear mapping:

$$T_P(x) \triangleq \begin{bmatrix} \mathrm{prox}_{\mu R_1}(x_1) \\ \vdots \\ \mathrm{prox}_{\mu R_N}(x_N) \end{bmatrix} \tag{20}$$

□

Using these operators, we can then rewrite the proximal diffusion algorithm (11a)–(11b) more compactly as the following concatenation of operators in terms of the network vector $\boldsymbol{w}_i = \mathrm{col}\{\boldsymbol{w}_{1,i}, \ldots, \boldsymbol{w}_{N,i}\}$:

$$\boldsymbol{w_i} = \widehat{\boldsymbol{T}}_{\mathrm{pd}}(\boldsymbol{w_{i-1}}) \triangleq T_A \circ T_P \circ \widehat{\boldsymbol{T}}_G(\boldsymbol{w_{i-1}}) \tag{21}$$

Without gradient noise, this relation reduces to:

$$w_i = T_{\mathrm{pd}}(w_{i-1}) \triangleq T_A \circ T_P \circ T_G(w_{i-1}) \tag{22}$$

Fig. 3 displays the stochastic proximal diffusion implementation as a cascade of operators.

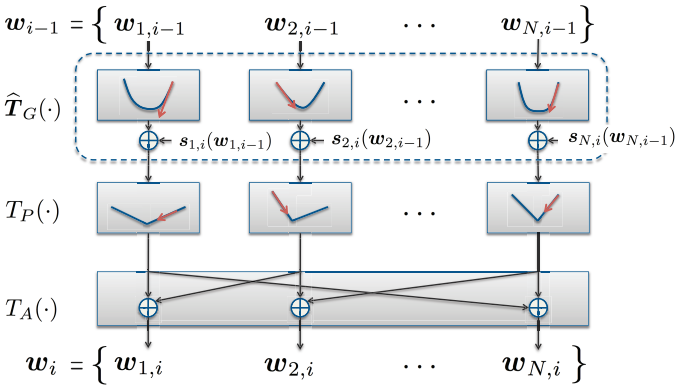

Fig. 1. Proximal diffusion as a cascade of operators.

4. MAIN RESULTS

4.1. Fixed-Point of Deterministic Recursion

Lemma 1 (Contractive Mapping)**.** *The deterministic proximal diffusion operator $T_{\mathrm{pd}}(\cdot)$ defined in (22) satisfies*

$$\|T_{\mathrm{pd}}(x) - T_{\mathrm{pd}}(y)\|_{b,\infty} \leq \gamma \cdot \|x - y\|_{b,\infty} \tag{23}$$

with $\gamma^2 \triangleq 1 - 2\mu\lambda_{\min} + \mu^2\lambda_{\max}^2$, and where $\|\cdot\|_{b,\infty}$ denotes the block maximum norm [5]. The condition on μ to guarantee $\gamma^2 < 1$ is:

$$0 < \mu < \frac{2\lambda_{\min}}{\lambda_{\max}^2} \tag{24}$$

Proof. Omitted for brevity. We only note that the argument exploits the following property of the proximal operator [17]:

$$\|\mathrm{prox}_{\mu R}(x) - \mathrm{prox}_{\mu R}(y)\| \leq \|x - y\| \tag{25}$$

□

It then follows from Banach's fixed point theorem [24,25] that $w_i = T_{\mathrm{pd}}(w_{i-1})$ converges to a unique fixed-point, w_∞, geometrically.

4.2. Bias Analysis

Now we analyze how far this fixed point w_∞ is from the desired global solution, w^o, to problem (4). In steady-state, the deterministic fixed-point equation (22) can be unfolded as follows:

$$\phi_{k,\infty} = \mathrm{prox}_{\mu R_k}\{w_{k,\infty} - \mu\nabla_w J_k(w_{k,\infty})\} \tag{26a}$$

$$w_{k,\infty} = \sum_{\ell=1}^{N} a_{\ell k}\phi_{\ell,\infty} \tag{26b}$$

To proceed, we introduce an assumption of bounded subgradients, which is common in the subgradient [3,23] and distributed proximal gradient [22] literature, namely, that for every agent k, the set of subdifferentials $\partial_w R_k^{\mathrm{org}}(w)$ is uniformly bounded, i.e. for all w:

$$\|\partial_w R_k^{\mathrm{org}}(w)\| \leq \eta_k^{\mathrm{org}} \tag{27}$$

for some non-negative constant η_k^{org}. For convex functions, the statement is equivalent to requiring $R_k^{\mathrm{org}}(w)$ to be Lipschitz continuous with constant η_k^{org}. For the scaled costs $R_k(w) \triangleq \delta\mu^\nu R_k^{\mathrm{org}}(w)$, condition (27) translates to:

$$\|\partial_w R_k(w)\| \leq \delta\mu^\nu\eta_k^{\mathrm{org}} \triangleq \eta_k = O(\mu^\nu) \tag{28}$$

Now we subtract Eqs. (26a) and (26b) from w^o and define the error variables $\widetilde{w}_{k,\infty} = w^o - w_{k,\infty}$. This leads to the error recursion:

$$\widetilde{\phi}_{k,\infty} = \widetilde{w}_{k,\infty} + \mu\nabla_w J_k(w_{k,\infty}) + \mu\widehat{\partial_w R}_k(\phi_{k,\infty}) \quad (29a)$$

$$\widetilde{w}_{k,\infty} = \sum_{\ell=1}^{N} a_{\ell k}\widetilde{\phi}_{\ell,\infty} \quad (29b)$$

Using the mean-value theorem [5, 26], we can write:

$$\nabla_w J_k(w_{k,\infty}) = \nabla_w J_k(w^o) - H_{k,\infty}\widetilde{w}_{k,\infty} \quad (30)$$

where $H_{k,\infty}$ denotes the Hessian of $J_k(w)$ at $w_{k,\infty}$. We get

$$\widetilde{\phi}_{k,\infty} = (I_M - \mu H_{k,\infty})\,\widetilde{w}_{k,\infty} + \mu\nabla_w J_k(w^o) + \mu\widehat{\partial_w R}_k(\phi_{k,\infty}) \quad (31a)$$

$$\widetilde{w}_{k,\infty} = \sum_{\ell=1}^{N} a_{\ell k}\widetilde{\phi}_{\ell,\infty} \quad (31b)$$

We next introduce the following extended vectors and matrices:

$$\widetilde{w}_\infty \triangleq \mathrm{col}\{\widetilde{w}_{1,\infty},\ldots,\widetilde{w}_{N,\infty}\} \quad (32)$$

$$\mathcal{A} \triangleq A \otimes I_M \quad (33)$$

$$\mathcal{H}_\infty \triangleq \mathrm{diag}\{H_{1,\infty},\ldots,H_{N,\infty}\} \quad (34)$$

$$\mathcal{B}_\infty \triangleq \mathcal{A}^{\mathsf{T}}(I_{MN} - \mu\mathcal{H}_\infty) \quad (35)$$

$$g^o \triangleq \mathrm{col}\{\nabla_w J_1(w^o),\ldots,\nabla_w J_N(w^o)\} \quad (36)$$

$$r_\infty \triangleq \mathrm{col}\left\{\widehat{\partial_w R}_1(\phi_{1,\infty}),\ldots,\widehat{\partial_w R}_N(\phi_{N,\infty})\right\} \quad (37)$$

With these quantities, relations (31a)–(31b) lead to:

$$\widetilde{w}_\infty = \mathcal{B}_\infty\widetilde{w}_\infty + \mu\mathcal{A}^{\mathsf{T}}(g^o + r_\infty). \quad (38)$$

Because A is a left-stochastic and primitive matrix, it admits a Jordan decomposition of the form $A = V_\epsilon J V_\epsilon^{-1}$ with

$$V_\epsilon = \left[\begin{array}{c|c} p & V_R \end{array}\right], \quad J = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & J_\epsilon \end{array}\right], \quad V_\epsilon^{-1} = \left[\begin{array}{c} \mathbb{1}^{\mathsf{T}} \\ \hline V_L^{\mathsf{T}} \end{array}\right] \quad (39)$$

where all diagonal entries of J_ϵ are inside the unit circle and J_ϵ consists of Jordan blocks with the value ϵ on the first lower diagonal instead of ones [5, 8]. Pre-multiplying both sides of (38) by $\mathcal{V}_\epsilon^{\mathsf{T}} = V_\epsilon^{\mathsf{T}} \otimes I_M$ gives:

$$\overline{w}_\infty = \overline{\mathcal{B}}_\infty\overline{w}_\infty + \mu\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}^{\mathsf{T}}(g^o + r_\infty) \quad (40)$$

where $\overline{w}_\infty = \mathcal{V}_\epsilon^{\mathsf{T}}\widetilde{w}_\infty$ and $\overline{\mathcal{B}}_\infty = \mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{B}_\infty(\mathcal{V}_\epsilon^{-1})^{\mathsf{T}}$. It follows that

$$\overline{w}_\infty = \mu\left(I_{MN} - \overline{\mathcal{B}}_\infty\right)^{-1}\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}^{\mathsf{T}}(g^o + r_\infty). \quad (41)$$

It was shown in [5, p. 541, Lemma 9.4] that, for sufficiently small step-sizes, it holds that

$$\left(I_{MN} - \overline{\mathcal{B}}_\infty\right)^{-1} = \left[\begin{array}{c|c} O(1/\mu) & O(1) \\ \hline O(1) & O(1) \end{array}\right] \quad (42)$$

where the leading $(1,1)$ block has dimensions $M \times M$. It can further be verified from the decomposition of V_ϵ in (39), that

$$\mathcal{V}_\epsilon^{\mathsf{T}}\mathcal{A}^{\mathsf{T}}(g^o + r_\infty) = \left[\begin{array}{c} \sum_{\ell=1}^{N} p_\ell\widehat{\partial_w R}_\ell(\phi_{\ell,\infty}) \\ \hline O(1) + \mathcal{V}_R^{\mathsf{T}}\mathcal{A}^{\mathsf{T}}r_\infty \end{array}\right] \quad (43)$$

Theorem 1. *Under assumption (27) and for small μ, the steady-state bias of the deterministic proximal diffusion recursion is bounded as:*

$$\|w^o - w_{k,\infty}\|^2 \le O\left(\mu^{2\nu}\right) + O(\mu^2) \quad (44)$$

Proof. The result follows from (28) and (42)–(43). □

4.3. Evolution of Stochastic Recursion

We now examine how close the stochastic recursion $\boldsymbol{w}_i = \widehat{\boldsymbol{T}}_{\mathrm{pd}}(\boldsymbol{w}_{i-1})$ approaches w^o. For this purpose, we introduce the mean-square perturbation vector at time i relative to w_∞:

$$\mathrm{MSP}_i \triangleq \mathrm{col}\left\{\mathbb{E}\|\boldsymbol{w}_{k,i} - w_{k,\infty}\|^2\right\} \in \mathbb{R}^N \quad (45)$$

Lemma 2. *The* MSP *at time i can be recursively bounded as:*

$$\mathrm{MSP}_i \preceq \left(\gamma^2 + 2\mu^2\beta^2\right)A^{\mathsf{T}}\mathrm{MSP}_{i-1} + \mu^2 d \quad (46)$$

where $d = O(1)$. A sufficient condition on μ for stability of (46) is:

$$0 < \mu < \frac{2\lambda_{\min}}{\lambda_{\max}^2 + 2\beta^2} \quad (47)$$

It follows that

$$\limsup_{i\to\infty} \|\mathrm{MSP}_i\|_\infty = O(\mu). \quad (48)$$

Proof. Omitted for brevity. □

The following theorem ties all results together.

Theorem 2. *For sufficiently small step-sizes and $\nu \ge 1/2$, the steady-state* MSD *of the proximal diffusion algorithm (11a)–(11b) is*

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu) \quad (49)$$

Proof. The result follows from (44) and (48). □

5. NUMERICAL RESULTS

Consider a network of $N = 10$ agents and $M = 20$. The network topology is shown in Fig. 2. Observations $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ for each agent k are generated according to the linear regression model $\boldsymbol{d}_k = \boldsymbol{u}_k w^o + \boldsymbol{v}_k$, where $\boldsymbol{u}_{k,i}$ and $\boldsymbol{v}_k(i)$ are zero-mean Gaussian random variables with power shown in Fig. 3. The true w^o is sparse with only one non-zero element. For the special case with $J_k(w) = \mathbb{E}\|\boldsymbol{d}_k - \boldsymbol{u}_k w\|^2$ and $R_k^{\mathrm{org}}(w) = \|w\|_1$, we compare the performance of the regularized proximal diffusion implementation (11a)–(11b) and the unregularized diffusion implementation (1a)–(1b). Fig. 5 displays the steady-state MSD for different choices of the step-size parameter. Note that, for small step-sizes, the MSD of the proximal diffusion implementation decreases linearly with μ. This is consistent with the theoretically derived expression (49).

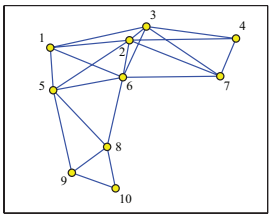

Fig. 2. Network topology.

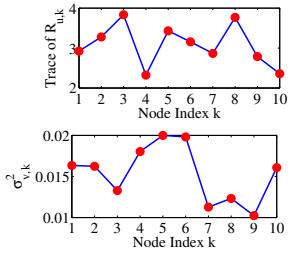

Fig. 3. Data statistics.

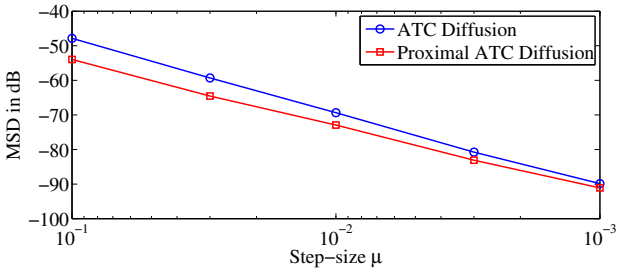

Fig. 4. Performance comparison for $\delta = 0.1, \nu = 1$.

6. REFERENCES

[1] J. N. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Trans. Automatic Control*, vol. 29, no. 1, pp. 42–50, Jan 1984.

[2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[3] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan 2009.

[4] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[5] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, July 2014.

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

[7] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, April 2013.

[8] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2003.

[9] S. U. Pillai, T. Suel, and S. Cha, "The Perron-Frobenius theorem: Some of its applications," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 62–75, March 2005.

[10] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Sparse diffusion LMS for distributed adaptive estimation," in *Proc. IEEE ICASSP*, Kyoto, Japan, March 2012, pp. 3281–3284.

[11] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, March 2013.

[12] 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.

[13] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4480–4485, Aug 2012.

[14] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, pp. 185–221, Springer, NY, 2011.

[15] P. L. Combettes, "Solving monotone inclusions via compositions of nonexpansive averaged operators," *Optimization*, vol. 53, no. 5-6, pp. 475–504, 2004.

[16] P. L. Combettes and V. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[17] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2013.

[18] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[19] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proc. IEEE ICASSP*, Dallas, USA, March 2010, pp. 3734–3737.

[20] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 1001–1016, February 2015.

[21] M. Schmidt, N. L. Roux, and F. R. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," in *Proc. Advances in Neural Information Processing Systems 24*, Granada, Spain, 2011, pp. 1458–1466.

[22] A. I. Chen and A. Ozdaglar, "A fast distributed proximal-gradient method," in *Proc. Annual Allerton Conference on Communication, Control, and Computing*, Allerton, USA, Oct. 2012, pp. 601–608.

[23] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *Journal of Machine Learning Research*, vol. 10, pp. 2899–2934, 2009.

[24] E. Kreyszig, *Introductory Functional Analysis with Applications*, John Wiley & Sons, 1989.

[25] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 1997.

[26] B. T. Polyak, *Introduction to Optimization*, Optimization Software, 1997.