Distributed empirical risk minimization over directed graphs

Ran Xin, Anit Kumar Sahu, Soummya Kar, and Usman A. Khan

***Abstract*— In this paper, we present stochastic optimization techniques for empirical minimization over directed graphs. Using a novel information fusion approach that utilizes both row- and column-stochastic weights simultaneously, we show that the proposed approach converges linearly to an error ball around the optimal solution with a constant step-size. Moreover, the algorithm converges to the optimal solution at $O(1/k)$, where k is the number of iterations, when decaying step-sizes are chosen. In cases where column-stochastic weights cannot be constructed, as they have stringent requirements, we present an algorithm that only utilizes row-stochastic weights but at the expense of eigenvector estimation. Finally, we illustrate the theoretical results with the help of experiments with real data.**

***Index Terms*— Stochastic optimization, Decentralized algorithms, multi-agent systems, directed graphs**

I. Introduction

In many signal processing, control, and machine learning problems of emerging interest, the data is often collected by geographically dispersed devices or is often stored on different machines, thus giving rise to the need of scalable learning and inference solutions that do not require communicating, storing, and processing all data at one single entity. In addition, to leverage modern computing platforms, advanced computational frameworks that use the communication and computation resources efficiently are particularly favorable. Various programming models and implementations of master-worker configurations have been proposed, such as MapReduce [1] and federated learning [2], that are tailored for specific computing needs and environments. Such architectures, although provide scalable solutions, may not be desirable in certain scenarios: (i) since the master is required to constantly push and pull information of very high dimensions, it could potentially become a communication bottleneck [3]; (ii) they are generally vulnerable to malicious attacks; and, (iii) when enormous data is generated in a local and streaming fashion from a large number of mobile, geographically dispersed, heterogeneous devices, e.g., in the Internet of Things (IoT), one needs a paradigm shift from the master-worker to a peer-to-peer network.

Peer-to-peer architectures, see Fig. 1 (Right), eliminate the need for specialized master nodes or coordinators and are based on flexible, non-deterministic, local communication, and thus naturally provides promising solutions to the aforementioned issues. Decentralized optimization applies to ad hoc peer-to-peer networks with limited communication and computation resources and has been an active topic of research in control and signal processing literature for the past several decades. Consider a sensor network or a network of robots, comprised of inexpensive, battery-operated, wireless devices, deployed in an arbitrary fashion, e.g., by an aircraft in a battlefield, mixed in concrete to monitor strength and damage in buildings, or buried under ground to monitor soil properties in a remote forest. These applications are giving rise to large-scale multi-agent machine learning problems where the application constraints do not allow for frequent battery recharges, expensive on-board computation, highly-sophisticated antennas, or manual re-installation/maintenance of the equipment. The communication is also not with the help of physical cables but is wireless, often in unfriendly/inaccessible territories, that is subject to packet drops, interference, fading, and equipment malfunction, in general. Such wireless networks only allow a minimal set of assumptions on connectivity, topology, synchrony, latency, etc., and thus peer-to-peer *ad hoc* and *unstructured* architectures are often used to model the underlying communication.

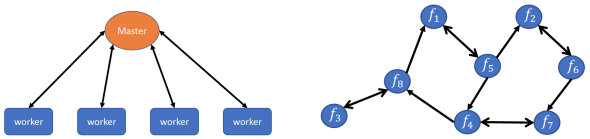

Fig. 1. (Left) A master-worker architecture. (Right) Decentralized consensus-based optimization.

This paper focuses on decentralized optimization where nodes collaborate to solve empirical risk minimization problems by communicating on proximity-based graphs, formed naturally by interactions among nearby agents. Having such ad hoc networks also paves the way to address imperfect cases, e.g., when the communication links are subject to noise and the data packets drop randomly. Much of the existing work on decentralized stochastic optimization, see the next paragraph, is restricted to undirected networks. In practice, however, it may not always be desirable to deploy bidirectional communication. The nodes (devices), for example, in an IoT setting, can be largely heterogeneous and of different physical nature. Some nodes may have limited broadcast power and are only capable of sending information to their nearby nodes; they are, however, still potentially able to receive information from nodes in a much wider range. Besides, when communication is relatively expensive compared with computation or when some nodes suffer from communication overload, a sparse communication topology

RX and SK are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213; {ranx, soummyak}@andrew.cmu.edu. AKS is with the Bosch Center for Artificial Intelligence, Pittsburgh, PA; anit.sahu@gmail.com. UAK is with the Department of Electrical and Computer Engineering, Tufts University, Medford, MA, 02155; khan@ece.tufts.edu. The work of RX and SK has been partially supported by NSF under grant # CCF-1513936. The work of UAK has been partially supported NSF under awards #1350264, #1903972, and #1935555.

is favorable that can be achieved by eliminating some communication links. These scenarios lead to *directed networks* that are much more flexible to both design and implement. In this context, the specific aim of this paper is to develop algorithms that are applicable to directed graphs.

Related Work: Finite-sum optimization has been a topic of significant research in the areas of signal processing, control, and machine learning, see e.g., [3]–[8]. Decentralized solutions require two key ingredients: (i) consensus, i.e., reaching agreement; and, (ii) optimality, i.e., agreement on the optimal. Naturally, consensus is used as the basic block of decentralized optimization on top of which a gradient correction (innovation) is added to steer the agreement to the optimal. Initial work thus closely follows the progress in consensus over undirected graphs [9]–[16]. Optimization over undirected graphs (built on doubly-stochsatic (DS) weights) can be found in [10], [17]–[22]. For unbalanced directed graphs, it is not possible to construct DS weights and thus optimization over digraphs [23]–[32] builds on consensus with non-DS weights [33]–[37]. Required now is a division involving the Perron eigenvector of the non-DS weight matrix, see e.g., [27], [29]–[31]. Decentralized stochastic optimization can be found in [38]–[41]. These methods converge sub-linearly and outperform their deterministic counterparts when local data batches are large.

We now describe the rest of this paper. Section II formulates the decentralized empirical risk minimization problem and provides necessary assumptions. Section III develops the corresponding algorithm that is applicable to directed graphs and provides intuitive analysis arguments and generalizations. Section IV provides numerical experiments while Section V concludes the paper.

II. Problem Formulation

In parametric learning problems [42], the goal of a typical machine learning system is to train a model $\boldsymbol{\theta} \in \mathbb{R}^p$, that maps an input data point, $\mathbf{x} \in \mathbb{R}^{d_{\mathbf{x}}}$, to its corresponding output, $\mathbf{y} \in \mathbb{R}^{d_{\mathbf{y}}}$. The setup requires defining a loss function, $l(\mathbf{x};(\boldsymbol{\theta},\mathbf{y}))$, which represents the loss incurred by the model $\boldsymbol{\theta}$ on the data $(\mathbf{x},\mathbf{y})$. In the setup of statistical machine learning, we assume that each data point $(\mathbf{x},\mathbf{y})$ belongs to a joint probability distribution $\mathcal{P}(\mathbf{x},\mathbf{y})$. Ideally, we would like to find the optimal model parameter $\widetilde{\boldsymbol{\theta}}^*$ by minimizing the following *risk (expected loss) function* $\widetilde{F}(\boldsymbol{\theta})$. However, the true distribution $\mathcal{P}(\mathbf{x},\mathbf{y})$ is often hidden or intractable in practice. In supervised machine learning, one usually has access to a large set of training data points $\{\mathbf{x}_i,\mathbf{y}_i\}_{i=1}^m$, which can be considered as the independent and identically distributed (i.i.d.) realizations from the corresponding data distribution. The average of the losses incurred by the model $\boldsymbol{\theta}$ on a finite set of the training data $\{\mathbf{x}_i,\mathbf{y}_i\}_{i=1}^m$, known as the *empirical risk*, serves as an appropriate surrogate objective function for the expected risk $\widetilde{F}(\boldsymbol{\theta})$.

When the data is further distributed, the empirical risk problem is formulated on a network of nodes. To this aim, consider n nodes, such as machines, devices, or decision-makers, interconnected over an arbitrary *graph*, $\mathcal{G}=(\mathcal{V},\mathcal{E})$, not necessarily undirected, where $\mathcal{V}=\{1,\cdots,n\}$ is the set of nodes, and $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ is the collection of *ordered* edges, $(i,j), i,j\in\mathcal{V}$, such that node j can send information to node i, i.e., $j\to i$. Each node i holds a private and local cost function $f_i:\mathbb{R}^p\to\mathbb{R}$, which is not available at any other node. Specifically, each node i is a computing resource and stores/collects a local batch of m_i training data samples that are possibly private and are not allowed to share with other nodes. The global objective is to find the best model $\boldsymbol{\theta}^*$ by leveraging all data in the entire network, formally written as the *decentralized empirical risk minimization* problem:

$$\text{P1:}\quad \boldsymbol{\theta}^*=\operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^p} F(\boldsymbol{\theta}),$$
$$F(\boldsymbol{\theta})=\frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{\theta})\triangleq\frac{1}{n}\sum_{i=1}^n\Big(\frac{1}{m_i}\sum_{j=1}^{m_i} f_{i,j}(\boldsymbol{\theta})\Big),$$

where $f_i(\boldsymbol{\theta})\triangleq\frac{1}{m_i}\sum_{j=1}^{m_i} f_{i,j}(\boldsymbol{\theta})$ is the local empirical risk corresponding to the m_i local training samples at node i.

A. Assumptions

Before we proceed, we provide the definitions and assumptions required to present the proposed algorithm.

Definition 1: A function $f_i:\mathbb{R}^p\to\mathbb{R}$ is called *l-smooth* if its gradient is Lipschitz-continuous, i.e., $\forall\boldsymbol{\theta}_1,\boldsymbol{\theta}_2\in\mathbb{R}^p$, we have, for some positive constant $l>0$,

$$\|\nabla f_i(\boldsymbol{\theta}_1)-\nabla f_i(\boldsymbol{\theta}_2)\|\le l\|\boldsymbol{\theta}_1-\boldsymbol{\theta}_2\|.$$

Definition 2: A function $f:\mathbb{R}^p\to\mathbb{R}$ is called *μ-strongly-convex* if $\forall\boldsymbol{\theta}_1,\boldsymbol{\theta}_2\in\mathbb{R}^p$, we have, for some positive constant $\mu>0$,

$$f(\boldsymbol{\theta}_2)\ge f(\boldsymbol{\theta}_1)+\nabla f(\boldsymbol{\theta}_1)^\top(\boldsymbol{\theta}_2-\boldsymbol{\theta}_1)+\frac{\mu}{2}\|\boldsymbol{\theta}_1-\boldsymbol{\theta}_2\|^2.$$

We define $\mathcal{S}_{\mu,l}$ as the class of functions that are l-smooth and μ-strongly-convex. It is important to note that for $f_i\in\mathcal{S}_{\mu,l}$, there exits a unique global minimizer of f_i; furthermore, if each $f_i\in\mathcal{S}_{\mu,l}$, then $F=\frac{1}{m}\sum_{i=1}^m f_i\in\mathcal{S}_{\mu,l}$. Various popular machine learning models belong to the class $\mathcal{S}_{\mu,l}$, such as regularized linear regression, logistic regression, and support vector machines. The algorithm provided in this paper makes the following assumptions.

Assumption 1: Each local cost function is μ-strongly-convex and l-smooth, i.e., each $f_{ij}\in\mathcal{S}_{\mu,l}$.

Under Assumption 1, $F\in\mathcal{S}_{\mu,l}$ and therefore has a unique global minimizer $\boldsymbol{\theta}^*$. We further assume that all mini-batch stochastic gradients have bounded variance, precisely written as follows.

Assumption 2: The following holds for all nodes $i\in\mathcal{V}$ and all time k:

$$\mathbb{E}\left[\left\|\nabla_k^i-\nabla f_i(\boldsymbol{\theta}_k^i)\right\|_2^2|\boldsymbol{\theta}_k^i\right]\le\sigma^2,$$

where $\sigma>0$ is some positive constant and $\|\cdot\|_2$ is the standard Euclidean norm.

Assumption 2 is standard in the stochastic optimization literature [43].

III. Stochastic Optimization in Directed Networks

We now describe a recently proposed $\mathcal{AB}$ algorithm [44], [45] and its stochastic variant $\mathcal{SAB}$ [46], both of which remove the need of eigenvector estimation in methods [**?**] based on the push-sum algorithm [**?**], while still being applicable to arbitrary strongly-connected directed graphs. The $\mathcal{AB}$ algorithm is based on the gradient-tracking technique [**?**] and a novel application of both row and column-stochastic weights. In the $\mathcal{AB}$ algorithm, each node i maintains two vectors $\boldsymbol{\theta}_k^i$, an estimate of $\boldsymbol{\theta}^*$, and $\mathbf{d}_k^i$, a gradient tracker, both in $\mathbb{R}^p$, iteratively updated as

$$\boldsymbol{\theta}_{k+1}^i = \sum_{j\in\mathcal{N}_i^{\text{in}}} a_{ij}\boldsymbol{\theta}_k^j - \alpha\mathbf{d}_k^i, \tag{1a}$$

$$\mathbf{d}_{k+1}^i = \sum_{j\in\mathcal{N}_i^{\text{in}}} b_{ij}\mathbf{d}_k^j + \nabla f_i\left(\boldsymbol{\theta}_{k+1}^i\right) - \nabla f_i\left(\boldsymbol{\theta}_k^i\right), \tag{1b}$$

where $\mathbf{d}_0^i = \nabla f_i\left(\boldsymbol{\theta}_0^i\right), \forall i$. In (1a) and (1b), $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are respectively the row[1] and column-stochastic weight matrices associated with the directed graph $\mathcal{G}$. Since doubly-stochastic weights are no longer required in both updates (1a) and (1b), $\mathcal{AB}$ is naturally applicable to arbitrary strongly-connected graphs, undirected and directed alike. It is shown in [44], [45] that $\mathcal{AB}$ converges linearly to the global minimizer of F, when each $f_i \in \mathcal{S}_{\mu,l}$.

A. Sketch of the Analysis

To explain the exact convergence of $\mathcal{AB}$, we first write it in a vector-matrix format as follows:

$$\boldsymbol{\theta}_{k+1} = A\boldsymbol{\theta}_k - \alpha\mathbf{d}_k, \tag{2a}$$

$$\mathbf{d}_{k+1} = B\mathbf{d}_k + \nabla\mathbf{f}(\boldsymbol{\theta}_{k+1}) - \nabla\mathbf{f}(\boldsymbol{\theta}_k), \tag{2b}$$

where $\boldsymbol{\theta}_k, \mathbf{d}_k, \nabla\mathbf{f}(\boldsymbol{\theta}_k)$ concatenate their corresponding local variables $\{\boldsymbol{\theta}_k\}, \{\mathbf{d}_k^i\}, \{\nabla f_i(\boldsymbol{\theta}_k^i)\}$. It can be shown that the row-stochastic weight matrix A in (2a) moves the estimates of all nodes towards their weighted average [**?**], i.e., $\boldsymbol{\theta}_k^i \to \widehat{\boldsymbol{\theta}}_k$, where $\widehat{\boldsymbol{\theta}}_k = \sum_{i=1}^n [\boldsymbol{\pi}_a]_i \boldsymbol{\theta}_k^i$ and $\boldsymbol{\pi}_a$ is the left Perron vector of A. We then multiply $\boldsymbol{\pi}_a^\top$ from both sides of (2a) to obtain the dynamics that governs the evolution of $\widehat{\boldsymbol{\theta}}_k$ as follows:

$$\widehat{\boldsymbol{\theta}}_{k+1} = \widehat{\boldsymbol{\theta}}_k - \alpha\sum_{i=1}^n [\boldsymbol{\pi}_a]_i \mathbf{d}_k^i. \tag{3}$$

From (2b), it can be verified that $\sum_{i=1}^n \mathbf{d}_k^i = \sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}_k^i)$, i.e., the sum of local gradient trackers preserves the sum of local gradients [44]. It can also be shown that [47]: $\forall i \in \mathcal{V}$,

$$\lim_{k\to\infty} \left\| \mathbf{d}_k^i - [\boldsymbol{\pi}_c]_i \frac{1}{n}\sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}_k^i) \right\|_2 = 0. \tag{4}$$

Combining (3) and (4), we obtain the (approximate) gradient corrections on $\widehat{\boldsymbol{\theta}}_k$ (when $k \gg 1$):

$$\widehat{\boldsymbol{\theta}}_{k+1} = \widehat{\boldsymbol{\theta}}_k - \alpha\left(\sum_{i=1}^n [\boldsymbol{\pi}_a]_i [\boldsymbol{\pi}_c]_i\right)\left(\frac{1}{n}\sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}_k^i)\right), \tag{5}$$

[1]Row-stochastic weights are easy to construct as each node can arbitrarily assign weights to its incoming information.

which becomes the full-batch centralized gradient descent as $\boldsymbol{\theta}_k^i \to \widehat{\boldsymbol{\theta}}_k$, and thus leads to the exact geometric convergence of $\mathcal{AB}$. Therefore, it can be concluded that *in $\mathcal{AB}$, the row-stochasticity of the weight matrix A guarantees the agreement* while *the column-stochasticity of weight matrix B guarantees the optimality*. This is consistent with our previous discussion.

B. Stochastic $\mathcal{AB}$

Finally, the stochastic gradient variant of $\mathcal{AB}$, termed as $\mathcal{SAB}$, is presented in Algorithm 1.

Algorithm 1 $\mathcal{SAB}$ at each node i

Require: $\boldsymbol{\theta}_i^0 \in \mathbb{R}^p$, $\alpha > 0$, row-stochastic weights $A = \{a_{ij}\}$ and column-stochastic weights $B = \{b_{ij}\}$ associated with $\mathcal{G}$, $\mathbf{d}_0^i = \nabla_0^i$.

1: **for** $k = 0, 1, 2, \cdots$ **do**
2: **Sample** (without replacement) a mini-batch $\mathcal{T}_k^i \subseteq \mathcal{T}^i = \{1, \cdots, m_i\}$
3: **Compute** the local stochastic gradient $\nabla_k^i \triangleq \frac{1}{|\mathcal{T}_k^i|}\sum_{j\in\mathcal{T}_k^i} \nabla f_{i,j}\left(\boldsymbol{\theta}_k^i\right)$.
4: **Update**: $\boldsymbol{\theta}_{k+1}^i = \sum_{j\in\mathcal{N}_i} a_{ij}\boldsymbol{\theta}_k^j - \alpha\mathbf{d}_k^i$
5: **Update**: $\mathbf{d}_{k+1}^i = \sum_{j\in\mathcal{N}_i} b_{ij}\mathbf{d}_k^j + \nabla_{k+1}^i - \nabla_k^i$
6: **end for**

Under the Assumptions 1 and 2 and a sufficiently small constant step-size α, the convergence of $\mathcal{SAB}$ is given as the following [46]:

$$\limsup_{k\to\infty} \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\left\|\boldsymbol{\theta}_k^i - \boldsymbol{\theta}^*\right\|_2^2\right] = \left(1 - \mathcal{O}\left(\mu\alpha\right)\right)^k + \mathcal{O}\left(\sigma^2\right). \tag{6}$$

C. Generalization of $\mathcal{AB}$ and $\mathcal{SAB}$

The $\mathcal{AB}$ algorithm provides a fundamental insight by unifying various gradient-tracking based algorithms. First, it is straightforward to obtain the algorithm in [**?**] DOGT (**??**) from $\mathcal{AB}$ by replacing both the row-stochastic A and the column-stochastic B with doubly-stochastic weights. To derive the relationships between $\mathcal{AB}$ and gradient-tracking based approaches over directed graphs [**?**], we define a state transformation with the help of the left Perron vector $\boldsymbol{\pi}_a$ of A. Let Π_a be a diagonal matrix with $\boldsymbol{\pi}_a$ on its main diagonal. We then obtain a transformed $\mathcal{AB}$ with $\mathbf{z}_k \triangleq \Pi_a\boldsymbol{\theta}_k$:

$$\mathbf{z}_{k+1} = \widetilde{B}\mathbf{z}_k - \alpha\Pi_a\mathbf{d}_k, \tag{7a}$$

$$\boldsymbol{\theta}_k = \Pi_a^{-1}\mathbf{z}_k, \tag{7b}$$

$$\mathbf{d}_{k+1} = B\mathbf{d}_k + \nabla\mathbf{f}(\boldsymbol{\theta}_{k+1}) - \nabla\mathbf{f}(\boldsymbol{\theta}_k), \tag{7c}$$

where $\widetilde{B} \triangleq \Pi_a A \Pi_a^{-1}$. It can be shown that $\widetilde{B}$ is in fact column-stochastic and $\widetilde{B}\boldsymbol{\pi}_a = \boldsymbol{\pi}_a$. The transformed $\mathcal{AB}$ (7) has two weight matrices, B and $\widetilde{B}$, that are both column-stochastic and associated with the directed graph $\mathcal{G}$. Note however that the decentralized implementation of (7) requires the right Perron vector $\boldsymbol{\pi}_a$ of a column-stochastic matrix $\widetilde{B}$, which is global information and not locally known to any node. Thus, one can use local iterative eigenvector estimators

to replace to the corresponding divisions in Π_a^{-1}, similar to the procedure used in SGP [**?**]. The resulting algorithm is well-known as ADD-OPT and Push-DIGing [**?**], [48] in the literature and is a gradient-tracking extension of SGP. A similar state transformation on the $\mathbf{d}_{k+1}$-update in (2b) leads to another gradient-tracking based algorithm with only row-stochastic weights; details on such procedures can be found in [49].

IV. Numerical Experiments

In this section, we present numerical experiments to illustrate the convergence properties of the consensus-based optimization algorithms in the context of decentralized training of a regularized logistic regression model [42] to classify the hand-written digits $\{3, 8\}$ from the MNIST dataset. Each digit image is represented by a vector in $\mathbb{R}^{784}$. We generate a connected undirected graph, $\mathcal{G}_{un}$, and a strongly-connected directed graph, $\mathcal{G}$, using nearest-neighbor rules, both of which have $n = 50$ nodes. Doubly-stochastic weights are generated using the Metroplis method [50], while row-stochastic weights $A = \{a_{ij}\}$ and column-stochastic weights $B = \{b_{ij}\}$ are generated with the uniform weighting strategy: $a_{ij} = 1/|\mathcal{N}_i^{\text{in}}|, b_{ij} = 1/|\mathcal{N}_j^{\text{out}}|, \forall i, j$. We note that both weighting strategies are applicable to undirected graphs but only the uniform strategy can be used over directed graphs. In our setting, each node i has access to $m_i = 20$ training data, $\{\mathbf{x}_{i,j}, y_{i,j}\}_{j=1}^{m_i} \subseteq \mathbb{R}^{784} \times \{-1, +1\}$, where $\mathbf{x}_{ij}$ is the feature vector and y_{ij} is the corresponding binary label. The nodes cooperatively solve the following smooth and strongly-convex optimization problem:

$$\min_{\mathbf{b}\in\mathbb{R}^{784},\, c\in\mathbb{R}} F(\mathbf{b}, c) = \sum_{i=1}^{n} \frac{1}{m_i} \sum_{j=1}^{m_i} \ln\left[1 + \exp\left\{-(\mathbf{b}^\top \mathbf{x}_{ij} + c) y_{ij}\right\}\right] + \frac{\lambda}{2}\|\mathbf{b}\|_2^2.$$

To compare applicable algorithms, we plot the average residual $\frac{1}{n}\sum_{i=1}^n \|\boldsymbol{\theta}_k^i - \boldsymbol{\theta}^*\|_2^2$ across all nodes and the test error rate versus the number of local epochs (number of effective passes of local data batch). Over the undirected graph $\mathcal{G}_{un}$, we compare the performance of DSGD, DSOGT, and $\mathcal{SAB}$. Over the directed graph $\mathcal{G}$, we compare the performance of SGP and $\mathcal{SAB}$. The step-sizes for all methods are fixed as 0.02. The experimental results are shown in Fig. 2 and Fig. 3. It can observed that DSGD, DSOGT, SGP and $\mathcal{SAB}$

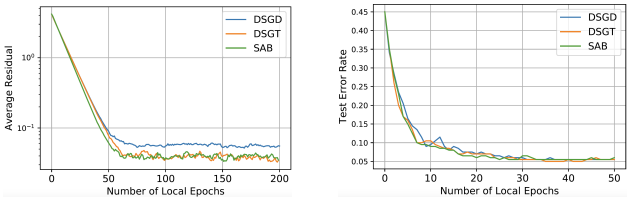

Fig. 2. Undirected graphs: average residual (left) and test error rate (right) versus number of local epochs.

are all effective methods for training the logistic regression classifier. DSGD and SGP have larger steady-state residuals due to their inherent bias, compared with gradient-tracking methods, DSOGT and $\mathcal{SAB}$. Moreover, the convergence of

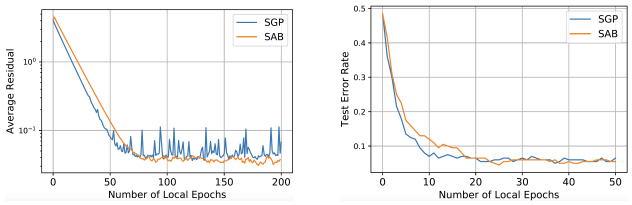

Fig. 3. Directed graphs: average residual (left) and test error rate (right) versus number of local epochs.

SGP is less stable compared with $\mathcal{SAB}$ because of the nonlinearity of the push-sum update in SGP. These results are consistent with our previous discussions.

V. Conclusions

In this paper, we describe a decentralized solution to empirical risk minimization problems when the data samples are distributed over a network of arbitrarily-connected nodes. Our particular focus is on directed graphs, i.e., when the nodes in the network may not be able to engage in bidirectional communication. To this aim, we discuss the $\mathcal{SAB}$ algorithm that utilizes a novel application of row and column-stochastic weights. Simulation results illustrate the discussion.

References

[1] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, Apr. 2017, vol. 54, pp. 1273–1282.

[3] X. Lian, C. Zhang, H. Zhang, C. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 5330–5340.

[4] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. on Automatic Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[5] Francesco Bullo, Jorge Cortes, and Sonia Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*, vol. 27, Princeton University Press, 2009.

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

[7] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "A decentralized second-order method with exact linear convergence rate for consensus optimization," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 507–522, 2016.

[8] H.-T. Wai, Z. Yang, Z. Wang, and M. Hong, "Multi-agent reinforcement learning via double averaging primal-dual optimization," *arXiv preprint arXiv:1806.00877*, 2018.

[9] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed subgradient projection algorithm for convex optimization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 3653–3656.

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

[11] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011.

[12] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, no. May, pp. 1663–1707, 2010.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundation and Trends in Maching Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[14] S. Lee and A. Nedić, "Distributed random projection algorithm for convex optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 221–229, Apr. 2013.

[15] S. Safavi and U. A. Khan, "Revisiting finite-time distributed algorithms via successive nulling of eigenvalues," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 54–57, Jan. 2015.

[16] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[17] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, Sep. 2016.

[18] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *IEEE 54th Annual Conference on Decision and Control*, 2015, pp. 2055–2060.

[19] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. on Control of Network Systems*, Apr. 2017.

[20] W. Shi, Q. Ling, G. Wu, and W Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[21] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning—part i: Algorithm development," *arXiv preprint arXiv:1702.05122*, 2017.

[22] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning—part ii: Convergence analysis," *arXiv preprint arXiv:1702.05142*, 2017.

[23] K. I. Tsianos, *The role of the Network in Distributed Optimization Algorithms: Convergence Rates, Scalability, Communication/Computation Tradeoffs and Communication Delays*, Ph.D. thesis, Dept. Elect. Comp. Eng. McGill University, 2013.

[24] A. Makhdoumi and A. Ozdaglar, "Graph balancing for distributed subgradient methods over directed graphs," *to appear in 54th IEEE Annual Conference on Decision and Control*, 2015.

[25] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. on Automatic Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.

[26] C. Xi, Q. Wu, and U. A. Khan, "On the distributed optimization over directed networks," *Neurocomputing*, vol. 267, pp. 508–515, Dec. 2017.

[27] C. Xi and U. A. Khan, "Distributed subgradient projection algorithm over directed graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3986–3992, Oct. 2016.

[28] C. Xi and U. A. Khan, "DEXTRA: A fast algorithm for optimization over directed graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4980–4993, Oct. 2017.

[29] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal of Optimization*, vol. 27, no. 4, pp. 2597–2633, Dec. 2017.

[30] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1329–1339, May 2018.

[31] C. Xi, V. S. Mai, R. Xin, E. Abed, and U. A. Khan, "Linear convergence in optimization over directed graphs with row-stochastic matrices," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3558–3565, Oct. 2018.

[32] R. Xin, C. Xi, and U. A. Khan, "FROST – Fast row-stochastic optimization with uncoordinated step-sizes," *EURASIP Journal on Advances in Signal Processing–Special Issue on Optimization, Learning, and Adaptation over Networks*, Jan. 2019.

[33] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science*, Oct. 2003, pp. 482–491.

[34] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *IEEE International Symposium on Information Theory*, Jun. 2010, pp. 1753–1757.

[35] K. Cai and H. Ishii, "Average consensus on general strongly connected digraphs," *Automatica*, vol. 48, no. 11, pp. 2750 – 2761, 2012.

[36] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Trans. on Automatic Control*, vol. 59, no. 3, pp. 781–786, March 2014.

[37] T. Charalambous, M. G. Rabbat, M. Johansson, and C. N. Hadjicostis, "Distributed finite-time computation of digraph parameters: Left-eigenvector, out-degree and spectrum," *IEEE Trans. on Control of Network Systems*, vol. 3, no. 2, pp. 137–148, June 2016.

[38] S. S. Ram, A. Nedich, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.

[39] A. Nedich and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, 2016.

[40] U. A. Khan R. Xin, A. K. Sahu and S. Kar, "Distributed stochastic optimization with gradient tracking over strongly-connected networks," in *58th IEEE Conference of Decision and Control*, Nice, France, Dec. 2019, to appear.

[41] B. Ying and A. H. Sayed, "Performance limits of stochastic sub-gradient learning, part II: Multi-agent case," *Signal Processing*, vol. 144, pp. 253–264, Mar. 2018.

[42] C. M. Bishop, *Pattern recognition and machine learning*, springer, 2006.

[43] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[44] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 315–320, 2018.

[45] S. Pu, W. Shi, J. Xu, and A. Nedić, "A push-pull gradient method for distributed optimization in networks," in *57th IEEE Annual Conference on Decision and Control*, Dec. 2018.

[46] R. Xin, A. K. Sahu, U. A. Khan, and S. Kar, "Distributed stochastic optimization with gradient tracking over strongly-connected networks," in *IEEE Conference on Decision and Control, accepted for publication, arXiv:1903.07266*, 2019.

[47] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 40–72, 2019.

[48] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Trans. on Automatic Control*, Aug. 2017, *in press*.

[49] R. Xin and U. A. Khan, "Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking," *arXiv preprint arXiv:1808.02942*, 2018.

[50] A. Nedić, A. Olshevsky, and M. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.