STOCHASTIC FIRST-ORDER METHODS OVER DISTRIBUTED DATA

Muhammad I. Qureshi and Usman A. Khan

Tufts University, Medford, MA

2022 IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM) | 978-1-6654-0633-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SAM53842.2022.9827792

ABSTRACT

In this paper, we study the problem of learning from data available over a network of geographically distributed nodes. Each node possess a private local cost function and the goal is to minimize the global cost defined as the average of all local costs. Assuming that the cost functions are smooth and strongly-convex, and that the information exchange among the nodes can be asymmetric, we propose two first-order stochastic optimization methods **PushSVRG** and **AB−SVRG** converging to the global minimum. Both methods use network level gradient tracking to eliminate the dissimilarity among heterogeneous data distribution and node level variance reduction to mitigate the variance caused by imperfect (local) gradient information. To eliminate the asymmetry of information exchange caused by the communication, **PushSVRG** uses column-stochastic weights and push-sum consensus while **AB−SVRG** uses both row and column stochastic weights and does not require the extra push-sum iterations. We compare the proposed methods with related work on first-order stochastic optimization using extensive numerical experiments and highlight the practical aspects of different variance reduction techniques.

***Index Terms*—** Distributed optimization, stochastic first-order methods, variance reduction.

1. INTRODUCTION

Stochastic first-order methods lie at the heart of many modern signal processing and machine learning tasks, see e.g., [1–5]. In many realistic scenarios, data is often collected and stored at various geographically distributed nodes. The nodes are connected over a strongly-connected graph and can send and receive information from their neighbours. Each node has a local cost function and the goal is to minimize the average of the local costs available at n nodes. Mathematically, we consider the following problem:

$$\mathbf{P1}: \min_{\mathbf{x}\in\mathbb{R}^p} F(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^p} \tfrac{1}{n}\textstyle\sum_{i=1}^n f_i(\mathbf{x}),$$

where each local cost $f_i(\mathbf{x})$ is private to node i. A special case of this problem is when each local f_i is further decomposable into a finite sum of m_i component cost functions leading to

$$\mathbf{P2}: \min_{\mathbf{x}\in\mathbb{R}^p} F(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^p} \tfrac{1}{n}\textstyle\sum_{i=1}^n \tfrac{1}{m_i}\sum_{j=1}^{m_i} f_{i,j}(\mathbf{x}).$$

Much existing work over distributed data assumes bi-directional communication over agents. However, many modern IoT or robotic networks may only allow one-way communication because of diverse battery and charging capabilities. We thus seek algorithms that are able to deal with directed graphs, i.e., one directional communication where a node may not be able to directly receive information from a node to which it can send information. The directed graph however is assumed to be strongly-connected, i.e., there exists a path between every two nodes in the network. In stochastic problems, another source of noise comes from imperfect data. In particular, we are not able to compute a gradient ∇f_i at any node i and can either compute a noisy gradient $\mathbf{g}_i$, in the context of Problem $\mathbf{P1}$, or randomly sample a component cost function from the local data, in the context of Problem $\mathbf{P2}$. In a more general form, a distributed stochastic first-order method thus has to address the following challenges: (i) *Information asymmetry* caused by the directed nature of the communication; (ii) *Data dissimilarity* across the nodes; (iii) *Imperfect gradient* information. The current research thus has addressed various facets of these challenges. We provide a brief literature survey next.

Early work on distributed optimization includes **DGD** that requires a bidirectional communication [6], which was then extended in **GP** [7] to directed graphs with the help of push-sum to address the asymmetry caused by the directed communication [8]. Both **DGD** and **GP** incur a steady-state error due to the data dissimilarity across the nodes, which was removed with the help of gradient tracking [9]. Gradient tracking based methods over undirected graphs were introduced in [10, 11] and then extended to directed graphs with the help of push-sum [12, 13]. A novel approach was introduced in **AB** [14], where push-sum was removed with the help of data fusion that utilizes both row and column stochastic matrices; a detailed survey can be found in [9]. The work we covered so far assumes full local gradients ∇f_i's available at each individual node. Stochastic extensions to **DGD** and **GP** can be found in **DSGD** [15] and **SGP** [16, 17], while [18, 19]

The authors are with the Electrical and Computer Engineering Department at Tufts University; muhammad.qureshi@tufts.edu and khan@ece.tufts.edu. This work has been partially supported by NSF under awards #1903972 and #1935555.

and **S-ADDOPT** [20] further use gradient tracking. Another line of work **S-AB** [21] removes push-sum consensus from **S-ADDOPT** and is the stochastic extension of **AB**.

In this paper, we introduce two new methods **PushSVRG** and **AB-SVRG** that use a variance reduction technique inspired by the **SVRG** method [22] to address distributed stochastic optimization over arbitrary strongly-connected graphs. During this process, we recap the recent work on stochastic first-order methods and describe their theoretical results. We further discuss the practical aspects of the corresponding methods and describe their limitations. We now describe the rest of the paper. Section 2 states the assumptions necessary for convergence and motivates the proposed methods. The algorithm developments are discussed in Section 3. Section 4 highlights the implementation of **SVRG**-based methods along with their comparison with other techniques. We provide extensive numerical experiments to show linear convergence of both **PushSVRG** and **AB-SVRG** and compare with other related methods in Section 5. Finally, Section 6 concludes the paper.

Notation: We use uppercase letters to denote matrices, lowercase letters to denote scalars and lowercase bold letters vectors. We define the Euclidean vector norm by $\|\cdot\|$.

2. ASSUMPTIONS AND MOTIVATION

We next describe the assumptions to guarantee linear convergence of **PushSVRG** and **AB-SVRG** to the unique minimum $\mathbf{x}^*$ of the global cost function F.

Assumption 1 (Stochastic First Oracle)**.** *Each agent i has access to a stochastic first-order oracle (SFO) that returns a stochastic gradient $\mathbf{g}_i(\mathbf{x}_i^k, \xi_i^k), \forall \mathbf{x}_i^k \in \mathbb{R}^p$, such that*

$$\mathbb{E}_{\xi_i^k}\left[\mathbf{g}_i(\mathbf{x}_i^k, \xi_i^k)|\mathbf{x}_i^k\right] = \nabla f_i(\mathbf{x}_i^k),$$
$$\mathbb{E}_{\xi_i^k}\left[\|\mathbf{g}_i(\mathbf{x}_i^k, \xi_i^k) - \nabla f_i(\mathbf{x}_i^k)\|^2|\mathbf{x}_i^k\right] \leq \sigma^2;$$

where we assume that the set of random vectors $\left\{\xi_i^k\right\}_{i=\{1,\ldots,n\}}^{k\geq 0}$ are independent of each other.

Assumption 2 (Smoothness and strong convexity)**.** *Each local f_i is L-smooth and the global cost F is μ-strongly convex.*

Assumption 3 (Strong Connectivity)**.** *The nodes communicate over a strongly-connected graph.*

These assumptions are standard in literature on distributed optimization. Assumption 1 ensures that each node can access it's stochastic gradient and that the second moment is bounded by a positive number. Assumption 2 guarantees that the f_i's are differentiable and that there exists a global minimizer of F, which is also unique. We now recap a few standard distributed stochastic first-order methods.

DSGD is one of the earliest first-order stochastic optimization method proposed in [15, 23] to solve $\mathbf{P1}$. Let $\alpha > 0$ be the step-size, $\mathbf{x}^*$ be the unique global minimum of $\mathbf{P1}$, and $\mathbf{x}_i^k \in \mathbb{R}^p$ be the estimate of the minimum at node i and iteration k. **DSGD** updates the estimate at each node as

$$\mathbf{x}_i^{k+1} = \textstyle\sum_{r=1}^n w_{ir}\mathbf{x}_i^k - \alpha \cdot \mathbf{g}_i(\mathbf{x}_i^k, \xi_i^k), \qquad \forall k \geq 0, \quad (1)$$

where $W = \{w_{ir}\} \in \mathbb{R}^{n\times n}$ is a doubly stochastic weight matrix restricting the network communication graph to be bi-directional (undirected). For a constant step-size α, the error $\mathbf{e}_i^k := \mathbb{E}\|\mathbf{x}_i^k - \mathbf{x}^*\|^2$ at each node decays linearly such that

$$\limsup_{k\to\infty} \tfrac{1}{n}\textstyle\sum_{i=1}^n \mathbf{e}_i^k = \mathcal{O}\left(\frac{\alpha}{n\mu}\sigma^2 + \frac{\alpha^2\kappa^2}{1-\lambda}\sigma^2 + \frac{\alpha^2\kappa^2}{(1-\lambda)^2}\eta\right),$$

where $\kappa = L/\mu$ is the condition number, $(1-\lambda)$ is the spectral gap of W, and $\eta = \frac{1}{n}\sum_{i=1}^n \|\nabla F(\mathbf{x}^*) - \nabla f_i(\mathbf{x}^*)\|^2$ is the error due to global versus local cost gap. The third term was removed in **DSGT** [18, 19] using gradient tracking where $\mathbf{g}_i(\mathbf{x}_i^k, \xi_i^k)$ in (1) is replaced by the following equation:

$$\mathbf{w}_i^k = \textstyle\sum_{r=1}^n w_{ir}(\mathbf{w}_r^k + \mathbf{g}_i(\mathbf{x}_i^{k+1}, \xi_i^{k+1}) - \mathbf{g}_i(\mathbf{x}_i^k, \xi_i^k)),$$

such that $\mathbf{w}_i^k$ mimics the stochastic gradient of global cost F. For a constant step-size, the resulting asymptotic error at each node decays linearly to an error ball given by [18]

$$\limsup_{k\to\infty} \tfrac{1}{n}\textstyle\sum_{i=1}^n \mathbf{e}_i^k = \mathcal{O}\left(\frac{\alpha}{n\mu}\sigma^2 + \frac{\alpha^2\kappa^2}{(1-\lambda)^3}\sigma^2\right).$$

It can be observed that the error in **DSGT** is a function of the variance σ^2 of the stochastic gradient evaluated by each node. **DSGT** converges to the global minimum $\mathbf{x}^*$ using decaying step-size but the rate of decay becomes sub-linear. To keep the convergence linear, variance reduction is often employed. Furthermore, we observe that **DSGD** and **DSGT** require bi-directional communication and therefore do not guarantee convergence for asymmetric information exchange. To deal with these challenges, we discuss some recent advancements on stochastic optimization in the following section.

3. ALGORITHM DEVELOPMENT

S-ADDOPT proposed in [20] is based only on column stochastic weights $B = \{b_{ir}\} \in \mathbb{R}^{n\times n}$ characterizing the (one directional) communication of the underlying network of nodes. In addition to gradient tracking, it uses push-sum [8] that requires a division with an eigenvector estimate of the underlying communication graph. The method is formally described below:

$$\mathbf{x}_i^{k+1} = \sum_{r=1}^n b_{ir}\mathbf{x}_i^k - \alpha\mathbf{w}_i^k,$$
$$y_i^{k+1} = \sum_{j=1}^n b_{ij}y_j^k, \qquad \mathbf{z}_i^{k+1} = \mathbf{x}_{k+1}^i/y_i^{k+1},$$
$$\mathbf{w}_i^{k+1} = \sum_{j=1}^n b_{ij}\mathbf{w}_j^k + \nabla\mathbf{g}_i(\mathbf{z}_i^{k+1}, \xi_i^{k+1}) - \nabla\mathbf{g}_i(\mathbf{z}_i^k, \xi_i^k),$$

where we initialize $y_i^k = 1, \forall i$. Using push-sum correction, it can be shown [9] that the scaled iterates $\mathbf{z}_i^k$ converge to the average of the initial states of the nodes, i.e., $\mathbf{z}_i^k \to \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^0$, $\forall i$. This iterative procedure of eigenvalue estimation slows down the convergence of **S-ADDOPT** especially when the network of nodes is not well connected. **S-AB** proposed in [21] uses row and column stochastic weights instead of using push-sum consensus and converges linearly to an error ball around the global minimum $\mathbf{x}^*$. This error is dependent on the variance caused by stochastic gradients evaluated at each node.

PushSAGA [24] and **AB-VR** [25] are two techniques built upon the **SAGA**-based variance reduction, applicable to Problem $\mathbf{P2}$. Each node evaluates the gradient of a randomly sampled component cost $\nabla f_{i,j}$ and estimates the gradient of the full local batch ∇f_i. **PushSAGA** uses push-sum consensus to handle the asymmetric information exchange and achieves ϵ-optimal solution in

$$\mathcal{O}\left(\max\left\{M, \frac{M}{m}\frac{\kappa^2\psi}{(1-\lambda)^2}\right\}\log\frac{1}{\epsilon}\right),$$

component gradient computations where $\psi \geq 1$ is the directivity constant, $M = \max_i m_i$ and $m = \min_i m_i$. **AB-VR** uses row and column stochastic weight to handle one directional communication between the nodes. It is similar to **S-AB** with the addition of variance reduction. **SAGA**-based optimization is generally faster than other first-order methods but it requires storing the most recent component gradients at each node [5].

4. SVRG BASED IMPLEMENTATION

Another well-known variance reduction method is **SVRG**. **SVRG** can be thought of as a double loop method, which achieves variance reduction by evaluating the local gradients periodically (with a period of T). At every outer-loop update $\{\mathbf{x}^{lT}\}_{l\geq 0}$, over l, each node computes local full gradient $\nabla f_i(\mathbf{x}^{lT})$, which is used in subsequent inner-loop iterations to update the local gradient estimator $\mathbf{v}_i^k$. For $k \in [lT, (l+1)T-1]$,

$$\mathbf{v}_i^k = \nabla f_{i,s_i^k}(\mathbf{x}_i^k) - \nabla f_{i,s_i^k}(\mathbf{x}_i^{lT}) + \nabla f_i(\mathbf{x}_i^{lT}).$$

We propose two **SVRG**-based methods formally described in Algorithm 1 and 2. **PushSVRG** (described in Algorithm 1) uses column stochastic weights B along with the *push-sum* iterations, to deal with asymmetric communication among the nodes, *gradient tracking*, to eliminate the local versus global cost gap, and ***SVRG*-based gradient estimation**, to eliminate the variance caused by inexact gradient evaluation at each iteration. **AB-SVRG** (described in Algorithm 2) uses row stochastic weights $A = \{a_{ir}\} \in \mathbb{R}^{n\times n}$ and column stochastic weights B to address the asymmetric communication among the nodes instead of the push-sum protocol. Avoiding push-sum potentially leads to faster convergence especially when the connectivity among the nodes is weak. We show numerically that both **PushSVRG** and **AB-SVRG** converge linearly to the global minimum. Detailed numerical studies on real data are provided in the next section. Next we highlight some features of the different methods provided in this paper and discuss several practical aspects.

Algorithm 1 **PushSVRG** at each node i

Require: $\mathbf{x}_i^0 \in \mathbb{R}^p, \mathbf{d}_i^0 = \mathbf{x}_i^0,\ \mathbf{w}_i^0 = \mathbf{v}_i^0 = \nabla f_i(\mathbf{x}_i^0)$, $\mathbf{y}_i^0 = 1,\ \alpha > 0,\ \{b_{ir}\}_{r=1}^n$,

for $k = 0, 1, 2, \ldots$ **do**
  $\mathbf{x}_i^{k+1} \leftarrow \sum_{r=1}^n b_{ir}(\mathbf{x}_r^k - \alpha \cdot \mathbf{w}_i^k)$
  $\mathbf{y}_i^{k+1} \leftarrow \sum_{r=1}^n b_{ir}\mathbf{y}_r^k$
  $\mathbf{z}_i^{k+1} \leftarrow \mathbf{x}_i^{k+1}/\mathbf{y}_i^{k+1}$
  Select s_i^{k+1} uniformly at random from $\{1, \ldots, m_i\}$
  if $\mod(k+1, T) = 0$, **then** $\mathbf{d}_i^{k+1} \leftarrow \mathbf{z}_i^{k+1}$, **else** $\mathbf{d}_i^{k+1} \leftarrow \mathbf{d}_i^k$
  end if
  $\mathbf{v}_i^{k+1} \leftarrow \nabla f_{i,s_i^{k+1}}(\mathbf{z}_i^{k+1}) - \nabla f_{i,s_i^{k+1}}(\mathbf{d}_i^{k+1}) + \nabla f_i(\mathbf{d}_i^{k+1})$
  $\mathbf{w}_i^{k+1} \leftarrow \sum_{r=1}^n b_{ir}(\mathbf{w}_r^k + \mathbf{v}_i^{k+1} - \mathbf{v}_i^k)$
end for

Remark 1 (Class of problems)**.** ***DSGT***, ***S-ADDOPT***, *and* ***S-AB*** *are applicable to a broader class of problems often referred to as streaming or online, where the local costs $f_i, \forall i$, are not necessarily decomposable. The variance reduction methods however assume that each f_i is decomposable into component costs as described in Problem* $\mathbf{P2}$ *and therefore* ***SAGA*** *and* ***SVRG****-based implementations are not directly applicable to Problem* $\mathbf{P1}$.

Remark 2 (Convergence to the global minimum)**.** *The stochastic optimization methods that don't use variance reduction converge to the global minimum sub-linearly with a decaying step-size.* ***PushSAGA***, ***AB-VR***, $PSVRG$,

Algorithm 2 **AB-SVRG** at each node i

Require: $\mathbf{x}_i^0 \in \mathbb{R}^p, \mathbf{d}_i^0 = \mathbf{x}_i^0,\ \mathbf{w}_i^0 = \mathbf{v}_i^0 = \nabla f_i(\mathbf{x}_i^0)$, $\alpha > 0,\ \{a_{ir}\}_{r=1}^n,\ \{b_{ir}\}_{r=1}^n$,

for $k = 0, 1, 2, \ldots$ **do**
  $\mathbf{x}_i^{k+1} \leftarrow \sum_{r=1}^n a_{ir}(\mathbf{x}_r^k - \alpha \cdot \mathbf{w}_i^k)$
  Select s_i^{k+1} uniformly at random from $\{1, \ldots, m_i\}$
  if $\mod(k+1, T) = 0$, **then** $\mathbf{d}_i^{k+1} \leftarrow \mathbf{x}_i^{k+1}$, **else** $\mathbf{d}_i^{k+1} \leftarrow \mathbf{d}_i^k$
  end if
  $\mathbf{v}_i^{k+1} \leftarrow \nabla f_{i,s_i^{k+1}}(\mathbf{x}_i^{k+1}) - \nabla f_{i,s_i^{k+1}}(\mathbf{d}_i^{k+1}) + \nabla f_i(\mathbf{d}_i^{k+1})$
  $\mathbf{w}_i^{k+1} \leftarrow \sum_{r=1}^n b_{ir}(\mathbf{w}_r^k + \mathbf{v}_i^{k+1} - \mathbf{v}_i^k)$
end for

and ***AB−SVRG*** *achieve faster (linear) convergence with the help of variance reduction.*

Remark 3 (Synchronization versus storage)**.** ***PushSVRG*** *and* ***AB−SVRG*** *require network synchrony as all nodes are required to evaluate their full local gradients every T iterations. This requirements results in idle time for nodes with smaller datasets as they complete their gradient computation faster than nodes with larger datasets.* ***PushSAGA*** *and* ***AB−VR*** *required extra storage as they need to keep a record of gradient table but do not need any network synchrony.*

5. NUMERICAL EXPERIMENTS

In this section, we illustrate the performance of **PushSVRG** and **AB−SVRG** for strongly-convex problems and compare it with related work. We randomly select two classes from MNIST and CIFAR-10 datasets and classify them using logistic regression with a strongly-convex regularizer. We use $N = 12{,}000$ labeled images to train, which are divided among n nodes. Each node possess a batch of m_i training samples and the local cost function can be described as

$$f_i = \frac{1}{m_i}\sum_{j=1}^{m_i} \ln\left[1+\exp\left\{-(\mathbf{b}^\top \mathbf{x}_{i,j}+c)y_{i,j}\right\}\right] + \frac{\lambda}{2}\|\mathbf{b}\|_2^2,$$

where $\mathbf{b}$ and c define the hyperplane separating the two classes. The network aims to solve for

$$\min_{\{\mathbf{b},c\}} F(\mathbf{b},c) = \frac{1}{n}\sum_{i=1}^{n} f_i.$$

To model variety in network connectivity, we use exponential graph for structured applications and geometric graph for ad-hoc training setups. Next we provide the performance comparison for different graphs.

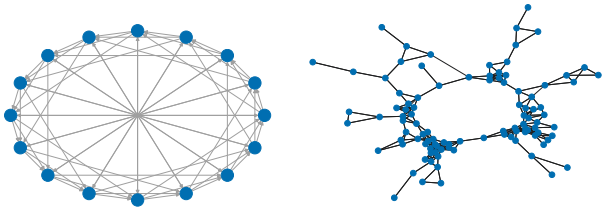

Fig. 1. (Left) Directed exponential graph with $n = 16$ nodes. (Right) geometric graph with $n = 100$ nodes

Structured training setup–Data-centers: We choose an exponential graph consisting of $n = 16$ nodes (see Fig. 1, left) to model a highly structured training setup, e.g., a data center. Each node possesses the same number of data samples, $m_i = 750, \forall i$, and we plot the optimality gap $F(\overline{\mathbf{x}}^k) - F(\mathbf{x}^*)$ such that $\overline{\mathbf{x}}^k = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^k$. Fig 2 shows the results for MNIST (left) and CIFAR-10 (right) classification. It can be seen that **S−ADDOPT** converges to an error ball around $\mathbf{x}^*$ but the methods that use variance reduction converge to the global minimum. Moreover, **PushSVRG** and **AB−SVRG** are

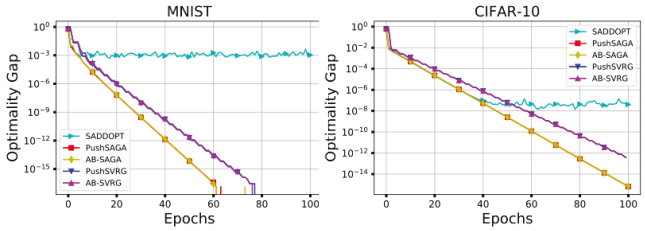

Fig. 2. Balanced data: performance comparison over exponential graph of $n = 16$ nodes.

comparatively slower that **PushSAGA** and **AB−VR** but are effected by network synchronization issues (see Remark 3).

Ad hoc training setup–Multi-agent networks: Next we use a geometric graph that, e.g., models ad hoc IoT-type training environments where the nodes possess different sizes of local batches depending on their location and resources. For a network with $n = 100$ nodes, we classify the images from MNIST and CIFAR-10 datasets and plot the optimality gap, see Fig. 3. It can be observed that **AB−SVRG** converges much faster than **PushSVRG**. This aspect is highlighted in the case of a geometric graph because it takes more time for **PushSVRG** to estimate the right eigenvector using push-sum.

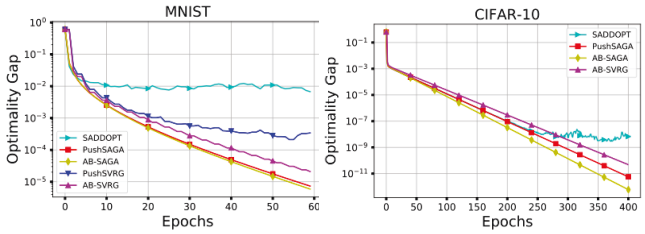

Fig. 3. Unbalanced data: performance comparison over geometric graph of $n = 100$ nodes.

6. CONCLUSIONS

In this paper, we proposed two first-order methods, **PushSVRG** and **AB−SVRG**, to minimize a sum of functions available over a geographically distributed network of nodes. Both methods are based on the **SVRG**-based variance reduction technique, to estimate local batch gradients from component gradients, and *gradient tracking*, to eliminate local versus global cost gap. To deal with the potential asymmetry in the underlying communication, **PushSVRG** uses column stochastic weights along with the push-sum correction, while **AB−SVRG** uses both row and column stochastic weights. For smooth local f_i and strongly-convex global cost F, we numerically showed that both algorithms converge linearly to the global minimum. We compared the performance with the related methods and further highlighted their practical aspects.

7. REFERENCES

[1] S. Safavi, U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed localization: A linear theory," *Proc. of the IEEE*, vol. 106, no. 7, pp. 1204–1223, Jul. 2018.

[2] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. of Machine Learning Research*, vol. 11, pp. 1663–1707, 2010.

[3] H. Raja and W. U. Bajwa, "Cloud *K*-SVD: A collaborative dictionary learning algorithm for big, distributed data," *IEEE Trans. on Sig. Process.*, vol. 64, no. 1, pp. 173–188, Jan. 2016.

[4] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278 – 305, 2019.

[5] 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.

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

[7] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[8] C. N. Hadjicostis and T. Charalambous, "Average consensus in the presence of delays in directed graph topologies," *IEEE Trans. on Automatic Control*, vol. 59, no. 3, pp. 763–768, 2014.

[9] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," *Proc. of the IEEE*, vol. 108, no. 11, pp. 1869–1889, 2020.

[10] P. D. Lorenzo and G. Scutari, "NEXT: in-network nonconvex optimization," *IEEE Trans. on Sig. and Inf. Process. over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

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

[12] C. Xi, R. Xin, and U. A. Khan, "ADD-OPT: Accelerated distributed directed optimization," *IEEE Trans. on Automatic Control*, vol. 63, no. 5, pp. 1329–1339, 2017.

[13] R. Xin, C. Xi, and U. A. Khan, "FROST—Fast row-stochastic optimization with uncoordinated step-sizes," *EURASIP J. on Advances in Sig. Process.*, Jan. 2019.

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

[15] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

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

[17] A. Spiridonoff, A. Olshevsky, and I. C. Paschalidis, "Robust asynchronous stochastic gradient-push: Asymptotically optimal and network-independent performance for strongly convex functions," *J. of Machine Learning Research*, vol. 21, no. 58, pp. 1–47, 2020.

[18] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Mathematical Programming*, 2020.

[19] R. Xin, U. A. Khan, and S. Kar, "An improved convergence analysis for decentralized online stochastic nonconvex optimization," *IEEE Trans. on Sig. Process.*, vol. 69, pp. 1842–1858, 2021.

[20] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "S-ADDOPT: Decentralized stochastic first-order optimization over directed graphs," *IEEE Con. Sys. Letters*, vol. 5, no. 3, pp. 953–958, 2021.

[21] R. Xin, A. K. Sahu, U. A. Khan, and S. Kar, "Distributed stochastic optimization with gradient tracking over strongly-connected networks," in *58th IEEE Conf. on Decision and Control*, Dec. 2019, pp. 8353–8358.

[22] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Adv. in Neural Inf. Process. Systems*, 2013, pp. 315–323.

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

[24] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "Push-SAGA: A decentralized stochastic algorithm with variance reduction over directed graphs," *IEEE Con. Sys. Letters*, vol. 6, pp. 1202–1207, 2022.

[25] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "Variance reduced stochastic optimization over directed graphs with row and column stochastic weights," 2022, arXiv: 2202.03346.