ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9413600

A DECENTRALIZED VARIANCE-REDUCED METHOD FOR STOCHASTIC OPTIMIZATION OVER DIRECTED GRAPHS

Muhammad I. Qureshi[y], Ran Xin[z], Soummya Kar[z], Usman A. Khan[y]

[y]Tufts University, Medford, MA, USA, [z]Carnegie Mellon University, Pittsburgh, PA, USA

ABSTRACT

In this paper, we propose a decentralized first-order stochastic optimization method **Push-SAGA** for finite-sum minimization over a strongly connected directed graph. This method features local variance reduction to remove the uncertainty caused by random sampling of the local gradients, global gradient tracking to address the distributed nature of the data, and push-sum consensus to handle the imbalance caused by the directed nature of the underlying graph. We show that, for a sufficiently small step-size, **Push-SAGA** linearly converges to the optimal solution for smooth and strongly convex problems, making it the first linearly-convergent stochastic algorithm over arbitrary strongly-connected directed graphs. We illustrate the behavior and convergence properties of **Push-SAGA** with the help of numerical experiments for strongly convex and non-convex problems.

Index Terms— Stochastic optimization, first-order methods, variance reduction, decentralized algorithms, directed graphs

1. INTRODUCTION AND RELATED WORK

Decentralized optimization has gained a significant interest recently because of its relevance to modern signal processing and machine learning tasks. In many such problems, the size of the data is very large and is available at nodes that are geographically distributed. Bringing the entire dataset to a central processor for learning an inference is computationally prohibitive and further requires extensive communication. It is therefore essential to design algorithms that are computationally efficient and requires minimal communication.

In this paper, we consider decentralized finite-sum minimization over a directed network of n nodes where each node i possesses a local batch of data, i.e.,

$$P: \quad \min_{z \in R^p} F(z) := \frac{1}{n} \sum_{i=1}^{n} f_i(z); \qquad f_i(z) := \frac{1}{m_i} \sum_{j=1}^{m_i} f_{i;j}(z);$$

where each local cost function $f_i : R^p \to R$ is decomposed into m_i component functions $f_{i;j}$, and j indexes the local data batch at node i; this setup is illustrated in Fig.1. Clearly, an algorithm based on the local batch gradient ∇f_i requires m_i gradient evaluations which could be cost prohibitive. Our focus thus is on stochastic methods that randomly sample one data point per iteration.

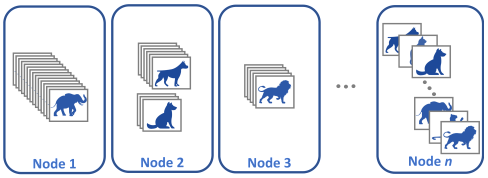

Fig. 1. Offline/Batch optimization problems.

A promising solution for Problem P is decentralized stochastic gradient descent (**DSGD**) [1, 2] that is further extended to directed graphs by **SGP** [3, 4] with the help of push-sum consensus [5,6]. It is well known that **DSGD** (and **SGP**) incur a steady state error due to the variance of the stochastic gradients, and the difference in the local gradient ∇f_i and the global gradient ∇F [7]. The steady state error due to the local vs. global functions' dissimilarity is eliminated in [8] with the help of gradient tracking [9–14], but the performance still suffers from the variance of stochastic gradients. This variance is eliminated in [15] with the help of variance reduction [16, 17]. All these methods however require doubly stochastic network weight matrices and thus are not applicable to arbitrary directed graphs. Of relevance here are also **SADDOPT** [18] that adds gradient tracking to **SGP** and is a stochastic extension of the methods described in [13, 14].

In this paper, we propose **Push-SAGA** that systematically eliminates the steady state errors in **SGP** and **SADDOPT** with a combination of three ingredients: local variance reduction, global gradient tracking, and push-sum consensus. As a result, **Push-SAGA** converges linearly to the optimal solution for smooth and strongly convex problems over arbitrary directed graphs. To the best of our knowledge, **Push-SAGA** is the first stochastic method that provides a linear convergence guarantee over directed graphs. We now describe the rest of this paper. Section 2 devel-ops and formally describes **Push-SAGA** while Section 3 provides the main result and convergence analysis. Finally, Section 4 presents numerical experiments on strongly convex

The work of MIQ and UAK has been partially supported by NSF under awards #1903972 and #1935555. The work of SK and RX has been partially supported by NSF under award #1513936.

Authorized licensed use limited to: TUFTS UNIV. Downloaded on August 07,2023 at 16:23:53 UTC from IEEE Xplore. Restrictions apply.

and non-convex problems.

Basic Notation: We use lowercase bold letters to represent vectors and uppercase letters for matrices. The column vector of n ones is denoted by $\mathbf{1}_n$ and I_n denotes the $n \times n$ identity matrix. For a primitive column stochastic matrix $B \in R^{n \times n}$, from Perron Frobenius theorem [19], we have that $\lim_{k \to \infty} B^k = \mathbf{1} v^\top$, where $v = \{v_i\}$ is the right eigenvector corresponding to the unique eigen-value 1. We use $\rho(B)$ to denote the spectral radius of B. We denote $\|\cdot\|$ and $|||\cdot|||$ as the Euclidean vector norm and the spectral (matrix) norm, respectively. Since $\rho(B - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) < 1$, it can be shown that there exists a matrix norm $|||\cdot|||_\alpha$, formally defined in [20], such that $\beta := |||B - \frac{1}{n}\mathbf{1}\mathbf{1}^\top|||_\alpha < 1$.

2. ALGORITHM DEVELOPMENT

Recall Problem P and the fact that the underlying communication graph is not necessarily undirected. **SGP** proposed in [3, 4] is a decentralized optimization algorithm that can be implemented over arbitrary directed graphs since it only requires a column stochastic weight matrix $B = \{b_{ir}\} \in R^{n \times n}$. **SGP** is formally described as follows: at each node i,

$$y_i^{k+1} = \sum_{r=1}^{n} b_{ir} y_i^k;$$

$$x_i^{k+1} = \sum_{r=1}^{n} b_{ir} x_i^k - \nabla f_{i;s_i^{k+1}}(z_i^k);$$

$$z_i^{k+1} = \frac{x_i^{k+1}}{y_i^{k+1}};$$

where x_i^k and z_i^k, both in R^p, estimate the global optimizer z at node i and time k, s_i^{k+1} is drawn uniformly at random from the index set $\{1, \ldots, m_i\}$, $\nabla f_{i;s_i^{k+1}}(z_i)^k$ is the gradient of a randomly drawn component function from the local batch, and $y_i^k \in R$ estimates the right eigenvector of B. At each node i, **SGP** is initialized with an arbitrary $x_i^0 \in R^p$, $z_i^0 = x_i^0$, and $y_i^0 = 1$. It can be shown [7] that the **SGP** iterate z_k^i, at each node i, converges to an error ball that is a function of two components, i.e., the variance introduced by stochastic gradient $\nabla f_{i;s_i^{k+1}}$ and the difference between the local batch gradient ∇f_i and the global gradient ∇F.

The proposed algorithm **Push-SAGA** eliminates the variance of the stochastic gradient, introduced due to the sampling of the local batch at each node, with the help of a variance reduction technique known as **SAGA** [16,17]. In particular, each node i maintains an iterate g_k^i that estimates the entire local batch gradient ∇f_i from randomly drawn samples of the local batch, and thus, loosely speaking, $g_k^i \to \nabla f_i(z_i^k)$. The **SAGA** estimator g_i^k requires a gradient table at each node i and thus results in extra storage costs. At each iteration of the algorithm, the s_i^{k+1}-th element of this gradient table is updated with the component gradient $\nabla f_{i;s_i^{k+1}}$ evaluated at the current iterate z_i^{k+1}. Subsequently, these variance-reduced estimators $\{g_i^k\}$'s are used to update the gradient tracking iterate w_i^k such that, loosely speaking, $w_i^k \to \sum_i g_i^k$, and thus the descent direction w_i^k asymptotically tracks the global gradient ∇F. The gradient tracking update comes from the dynamic average consensus protocol [21].

Algorithm 1 **Push-SAGA** at each node i

Require: $x_i^0 \in R^p$; $z_i^0 = x_i^0$; $w_i^0 = g_i^0 = \nabla f_i(z_i^0)$; $v_{i;j}^1 = z_i^0$; $\forall j \in \{1, \ldots, m_i\}$; $y_i^0 = 1$; $\gamma > 0$; $\{b_{ir}\}_{r=1}^n$; Gradient table: $\{\nabla f_{i;j}(v_{i;j})\}_{j=1}^{m_i}$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $x_i^{k+1} \leftarrow \sum_{r=1}^{n} b_{ir} x_r^k - \gamma w_i^k$;
$y_i^{k+1} \leftarrow \sum_{r=1}^{n} b_{ir} y_r^k$
4: $z_i^{k+1} \leftarrow x_i^{k+1} / y_i^{k+1}$
5: Select s_i^{k+1} uniformly at random from $\{1, \ldots, m_i\}$
6: $g_i^{k+1} \leftarrow \nabla f_{i;s_i^{k+1}}(z_i^{k+1}) - \nabla f_{i;s_i^{k+1}}(v_{i;s_i^{k+1}}^{k+1}) + \frac{1}{m_i} \sum_{j=1}^{m_i} \nabla f_{i;j}(v_{i;j}^{k+1})$
7: Replace $\nabla f_{i;s_i^{k+1}}(v_{i;s_i^{k+1}}^{k+1})$ by $\nabla f_{i;s_i^{k+1}}(z_i^{k+1})$ in the gradient table
8: $w_i^{k+1} \leftarrow \sum_{r=1}^{n} b_{ir} w_r^k + g_i^{k+1} - g_i^k$
9: **if** $j = s_i^{k+1}$, **then** $v_{i;j}^{k+2} \leftarrow z_i^{k+1}$, **else** $v_{i;j}^{k+2} \leftarrow v_{i;j}^{k+1}$
10: **end if**
11: **end for**

---

3. LINEAR CONVERGENCE OF PUSH-SAGA

This section formally describes the convergence analysis of **Push-SAGA**. We start with the following assumptions.

Assumption 1 (Strongly connected directed graphs). Each node communicates over a strongly connected directed graph. The underlying weight matrix $B = \{b_{ir}\}$ is such that it follows the sparsity of the graph. In addition, B is primitive and column stochastic, i.e., $\mathbf{1}_n^\top B = \mathbf{1}_n^\top$ and $B \mathbf{1}_n = \mathbf{1}_n$.

Assumption 2 (Smooth and strongly convex cost functions). Each local cost function f_i is μ-strongly convex and each component cost $f_{i;j}$ is L-smooth.

The above assumptions are standard in the literature. Assumption 1 for example can be fulfilled when each node chooses a suitable weight for its outgoing neighbors [13]. A valid choice at node i is $b_{ri} = 1/d_i^{out}$, for each outgoing neighbor r, that requires each node to know its out-degree denoted by d_i^{out}. Following Assumption 2, it can be verified that the global cost F is L-smooth and μ-strongly convex and thus have a unique global minimizer that is denoted by z.

Based on the above assumptions and **Push-SAGA** in Algorithm 1, we now provide the main result of this paper.

Theorem 1. Consider Problem P under Assumptions 1 and 2. For a sufficiently small step-size $\gamma > 0$, **Push-SAGA** linearly converges to the optimal solution z at each node.

Authorized licensed use limited to: TUFTS UNIV. Downloaded on August 07,2023 at 16:23:53 UTC from IEEE Xplore. Restrictions apply.

In the following, we provide the convergence analysis and the auxiliary results that are needed to prove Theorem 1. To this aim, we write **Push-SAGA** in a vector-matrix format. Let $x^k; z^k; g^k; w^k$, all in R^{pn} be the global vectors that stack the local variables $x_i^k; z_i^k; g_i^k; w_i^k \in R^p$, respectively, and $y^k \in R^n$ concatenate y_i^k's. Similarly, the global matrices can be defined as $B := \underline{B} \otimes I_p$ and $Y_k := \mathrm{diag}(y^k) \otimes I_p$. It can be verified that **Push-SAGA** in Algorithm 1 can be compactly written as

$$x^{k+1} = Bx^k - w^k; \tag{1a}$$

$$y^{k+1} = \underline{B}y^k; \tag{1b}$$

$$z^{k+1} = Y_k^{-1}x^{k+1}; \tag{1c}$$

$$w^{k+1} = Bw^k + g^{k+1} - g^k. \tag{1d}$$

We next introduce four error quantities that will aid in the convergence analysis:

(i) Network agreement error: $E\|x^k - B^\infty x^k\|^2$;
(ii) Optimality gap: $E\|\bar{x}^k - z^k\|^2$;
(iii) Mean auxiliary gap: $E[t^k]$;
(iv) Gradient tracking error: $E\|w^k - B^\infty w^k\|^2$;

where $\bar{x}^k := \frac{1}{n}\sum_i x_i^k$ and

$$t^k := \sum_{i=1}^n \Big(\frac{1}{m_i}\sum_{j=1}^{m_i} \|v_{i;j}^k - z^k\|^2\Big).$$

In order to show that **Push-SAGA** converges linearly to the global minimum, we establish that each of these four errors converges to zero linearly, which subsequently leads to $z_i^k \to z; \forall i$. These properties are derived with the help of an LTI system that captures the evolution of these error quantities. We start with a standard result that comes from the literature on directed graphs.

Lemma 1. [22] Let Assumption 1 hold and $Y^\infty := \lim_{k\to\infty} Y_k$, then $\|Y_k - Y^\infty\|_2 \le \sigma^k; \forall k$, where $\sigma := \max\{\lambda_2, \lambda_n\}$, $\lambda_2 := \max_i |\lambda_i|$ and $\lambda_n = \min_i |\lambda_i|$.

Lemma 2. Consider **Push-SAGA** under Assumptions 1, 2, and for all $k \ge 0$, define $u^k; s^k \in R^4$ and $G; H_k \in R^{4\times 4}$ as

$$u^k := \begin{bmatrix} E[\|x^k - B^\infty x^k\|^2] \\ E[n\|\bar{x}^k - z^k\|^2] \\ E[t^k] \\ E[L^2\|w^k - B^\infty w^k\|^2] \end{bmatrix}; \quad s^k := \begin{bmatrix} E[\|x^k\|_2^2] \\ 0 \\ 0 \\ 0 \end{bmatrix};$$

$$G := \begin{bmatrix} \frac{1+\sigma^2}{2} & 0 & 0 & \frac{\sigma}{2} \\ \frac{2L^2\sigma^2}{(1-\sigma^2)^2} & \frac{2}{1-\sigma^2} & \frac{m}{5M} & 0 \\ \frac{2}{m} & \frac{2}{169m(1-\sigma^2)} & 1 - \frac{1}{38M} & 0 \\ \frac{188\sigma^2}{(1-\sigma^2)^2} & \frac{169}{(1-\sigma^2)^2} & \frac{38}{(1-\sigma^2)^2} & \frac{3+\sigma^2}{4} \end{bmatrix};$$

$$H_k := \begin{bmatrix} 2 & 0 & 0 & 0 \\ \frac{2L^2}{(1-\sigma^2)^2} & 0 & 0 & 0 \\ \frac{2}{m} & 0 & 0 & 0 \\ \frac{188\sigma^2}{(1-\sigma^2)^2} & 0 & 0 & 0 \end{bmatrix} T^k;$$

where $m := \min_i m_i$; $M := \max_i m_i$; $\gamma := \sup_k \|Y_k\|_2$; $\gamma := \sup_k \|Y_k^{-1}\|_2$, and $\Gamma := \gamma\gamma^2(1+T)h$. Then, for $\alpha := L/\tau$ and the step-size $0 < \tau < \frac{1-\sigma^2}{28L}$, we have

$$u^k \le G u^{k-1} + H_{k-1} s^{k-1}. \tag{2}$$

The proof of Lemma 2 can be found in [23] and follows the procedure in [15, 18]. Next, we show that $\|u^k\|_2$ goes to zero and further characterize its convergence rate. To this aim, note that H_k linearly decays to a zero matrix at $O(\sigma^k)$. The evolution of (2) further requires the spectral radius $\rho(G)$ of G. We characterize this spectral radius in Lemma 4 with the help of the matrix perturbation argument in Lemma 3.

Lemma 3. [19] Consider $X; Y \in R^{n\times n}$. Let λ be simple eigenvalue of X, and w^* and v be left and right eigenvectors of X corresponding to λ. Let $\lambda(t)$ be an eigenvalue of $X + tY; \forall t \in R$. Then $\lambda(t)$ is unique and differentiable at $t = 0$, and

$$\frac{d\lambda(t)}{dt}\Big|_{t=0} = \frac{w^H Y v}{w^H v}.$$

Lemma 4. Consider G defined in Lemma 2. If the stepsize τ is sufficiently small, then $\rho(G) < 1$.

Proof. We start by noting that G can be decomposed into two components: one that is independent of the step-size τ and another perturbation term that is controlled by τ. In particular, we have that $G := G_0 + P$; such that

$$G_0 := \begin{bmatrix} \frac{1+\sigma^2}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{2}{m} & \frac{2}{169m(1-\sigma^2)} & 1 - \frac{1}{38M} & 0 \\ \frac{188\sigma^2}{(1-\sigma^2)^2} & \frac{169}{(1-\sigma^2)^2} & \frac{38}{(1-\sigma^2)^2} & \frac{3+\sigma^2}{4} \end{bmatrix};$$

$$P := \begin{bmatrix} 0 & 0 & 0 & \frac{\sigma}{2} \\ \frac{2L^2\sigma^2}{(1-\sigma^2)^2} & \frac{2}{1-\sigma^2} & \frac{m}{5M} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since G_0 is lower triangular, its eigenvalues appear on the diagonal and it can be verified that $\rho(G_0) = 1$ since $\sigma^2 \in [0; 1)$ and $M \ge 1$. We next note that the left eigenvector of G_0, corresponding to the eigenvalue 1, is $w^* = [0; 1; 0; 0]$, whereas, by choosing $r = 169/M + 76/m$, it can be shown that the corresponding right eigenvector is

$$v = \Big[0; \frac{(1-\sigma^2)^2}{4Mr}; \frac{(1-\sigma^2)^2}{2mr}; 1\Big]^\top.$$

Denoting the eigenvalues of G by $\lambda(\tau)$, Lemma 3 leads to $\frac{d\lambda(\tau)}{d\tau}\big|_{\tau=0} = \frac{w^H P v}{w^H v} = -\frac{1}{10} < 0$;

because 1 is a simple eigenvalue of G_0. Finally, since eigenvalues are continuous functions of the parameters of a matrix, we have that $\rho(G)$ decreases from 1 as τ slightly increases from zero and the proof follows. □

The following corollary derives the linear convergence of u_k.

Authorized licensed use limited to: TUFTS UNIV. Downloaded on August 07,2023 at 16:23:53 UTC from IEEE Xplore. Restrictions apply.

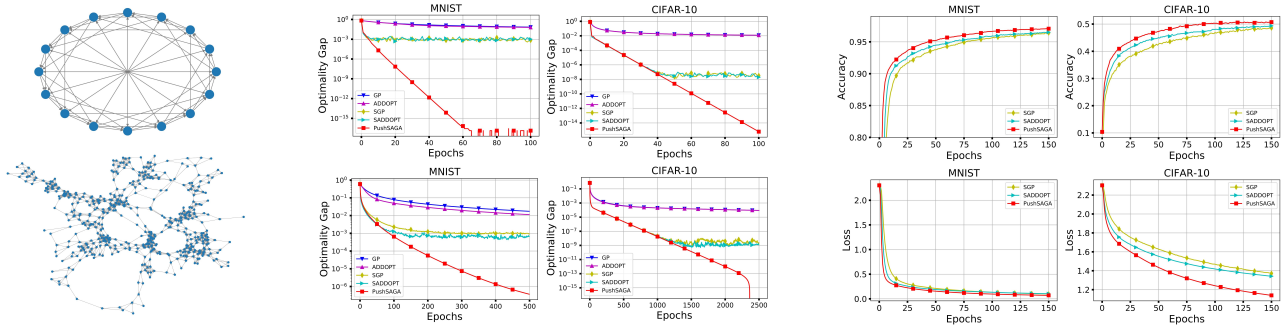

Fig. 2. (Left) Directed exponential graph (top) with n = 16 nodes and directed geometric graph (bottom) with n = 500 nodes. (Center) Optimality gap for logistic regression classifier trained over directed exponential graph (top) and directed geometric graph (bottom); (right) Test accuracy and training loss for neural networks trained over a geometric graph.

Corollary 1. Let := max{; (G)g, then $\|u^k\|_2$ decays linearly to zero i.e., there exists a constant > 0; such that $\|u^k\|_2$ (+)^k; 8k; where > 0 is arbitrarily small.

Proof. We expand (2) and take the Euclidean norm on both sides. Then for some positive constants $_1$ and $_2$,

$$\|u^k\|_2 \quad \|Gu^{0k}\|_2 + \sum_{r=0}^{k} \|G^{k-r-1} H_r s^r\|_2; \quad {}_1$$
$$+ {}_2 \sum_{r=0}^{k-1} \|s^r\|_2 \quad {}^k;$$

It can be verified that $\|s^r\|_2$ $6(y^2 +)\|u^r\|_2 + 3y^2 n\|z\|^2$. $_2$ Then for b := 6 $(y^2 +)$ and c := 3 $y^2 n\|z\|^2$, we have

$$\|u^k\|_2 \quad {}_1 + kc + b \sum_{r=0}^{k} {}^{k-r-1} \|u^r\|_2:$$

We denote $c_k := (_1 + kc)^k$; $w_k := \sum_{r=0}^{k-1} {}^{k-r-1} \|u^r\|_2$, and for $b_k := b^k$, we can rewrite the above equation as

$$\|u^k\|_2 = w_{k+1} \quad w_k \quad (_1 + kc + bw_k)^k:$$

We note that 8k; w_{k+1} $(1 + b_k)w_k + c_k$, whereas the se-ries $\sum_{k=0}^{1} b_k < 1$ and $\sum_{k=0}^{1} c_k < 1$. Thus from [24] we have that the sequence $\{w_k\}$ converges. Hence, it is bounded and 8 2 (; 1),

$$\lim_{k \to 1} \frac{\|u^k\|_2}{k} \quad \lim_{k \to 1} \frac{(_1 + kc + bw_k)^k}{k} = 0; \text{ and}$$

the corollary follows. □

3.1. Proof of Theorem 1

Proof. We find a bound on $E[\|z^k \quad (1_n \quad {}_2$ $z)\|^2]$ with the help of Corollary 1. We have

$$E[\|z^k \quad (1_n \quad {}_2 \quad {}_2$$
$$z)\|^2] \quad 3y^2 E[\|x^k \quad B^{-1} x^k\|^2] + 3ny^2 \quad {}_2$$
$$y^2 E[\|x^{k-} \quad z\|^2] + 3n(y \quad T^k)^2 kzk^2$$
$$6y^2 (y^2 +)\|u^k\|_2 + 3ny^2 T^{2k} kzk^2 \quad {}_2$$
$$6y^2 (y^2 +)(\quad +)^k + 3ny^2 T^2 (+)^k kzk^2 \quad 6y^2 \,!\, (+ {}_2$$
$$)^k;$$

with ! := $(y^2 +)\quad + nT^2 kzk^2$ and the result follows. □

4. NUMERICAL EXPERIMENTS

In this section, we provide numerical experiments to illustrate **Push-SAGA** for strongly convex and non-convex problems, and compare its performance with related methods.

Logistic Regression: We first train a binary classifier on N = 12;000 images from two classes in MNIST and CIFAR-10 datasets with the help of logistic regression, with a strongly convex regularizer. These images are divided among n nodes that communicate over directed exponential and geometric graphs, see Fig. 2 (left). We compare **Push-SAGA** with **SGP** and **SADDOPT**, along with their deterministic counterparts **GP** and **ADDOPT**, and plot the optimality gap $F(\bar{z}^k) \quad F(z)$, where $z^{k-} := \frac{1}{n} \sum_i z_i^k$, over epochs (one epoch represents m_i gradient computations per node). Fig. 2 (center) shows that **Push-SAGA** is much faster than **GP**, **ADDOPT** and, unlike the stochastic algorithms (**SGP**, **SADDOPT**), converges linearly to the optimal solution.

Neural Networks: Finally, we train a multi-class classifier using N = 60;000 images from the MNIST and CIFAR-10 datasets, with 6;000 images per class. We use this data to train decentralized neural networks distributed over a directed geometric graph, where each node possess a custom two-layered neural network comprising of a hidden layer of 64 neurons and a fully connected output layer. We compare the stochastic algorithms: **SGP**, **SADDOPT** and **Push-SAGA**, in Fig. 2 (right) that plots the loss $F(\bar{z}^k)$ and test accuracy. It can be seen that **Push-SAGA** is not only faster than the other two algorithms but also has smaller loss and better accuracy.

5. CONCLUSION

This paper describes **Push-SAGA**, a decentralized stochastic method applicable to arbitrary directed graphs. We show that **Push-SAGA** linearly converges to the optimal solution for smooth and strongly convex problems with a sufficiently small step-size. Numerical experiments illustrate the performance for both strongly convex and non-convex problems.

Authorized licensed use limited to: TUFTS UNIV. Downloaded on August 07,2023 at 16:23:53 UTC from IEEE Xplore. Restrictions apply.

6. REFERENCES

[1] S. S. Ram, A. Nedić, 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.

[2] 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.

[3] A. Nedić 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.

[4] M. Assran, N. Loizou, N. Ballas, and M. G. Rabbat, "Stochastic gradient-push for distributed deep learning," in 36th International Conference on Machine Learning, Jun. 2019, vol. 97, pp. 344–353.

[5] 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.

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

[7] R. Xin, S. Kar, and U. A. Khan, "Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence," IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 102–113, 2020.

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

[9] 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 Conference on Decision and Control, 2015, pp. 2055–2060.

[10] P. D. Lorenzo and G. Scutari, "NEXT: in-network nonconvex optimization," IEEE Transactions on Signal and Information Processing over Networks, vol. 2, no. 2, pp. 120–136, 2016.

[11] Y. Tian, Y. Sun, and G. Scutari, "Achieving linear convergence in distributed asynchronous multi-agent optimization," IEEE Transactions on Automatic Control, pp. 1–1, 2020.

[12] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," IEEE Transactions on Control of Network Systems, vol. 5, no. 3, pp. 1245–1260, 2017.

[13] 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, 2017.

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

[15] R. Xin, U. A. Khan, and S. Kar, "Variance-reduced decentralized stochastic optimization with accelerated convergence," arXiv:1912.04230, Dec. 2019.

[16] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," Mathematical Programming, vol. 162, no. 1-2, pp. 83–112, 2017.

[17] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in Advances in Neural Information Processing Systems, 2014, pp. 1646–1654.

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

[19] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 2nd edition, 2012.

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

[21] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," Automatica, vol. 46, no. 2, pp. 322–329, 2010.

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

[23] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, "Push-SAGA: A decentralized stochastic algorithm with variance reduction over directed graphs," arXiv:2008.06082, 2020.

[24] B. Polyak, Introduction to optimization, Optimization Software, 1987.

Authorized licensed use limited to: TUFTS UNIV. Downloaded on August 07,2023 at 16:23:53 UTC from IEEE Xplore. Restrictions apply.