Quantized Decentralized Stochastic Learning over Directed Graphs

Hossein Taheri [1] **Aryan Mokhtari** [2] **Hamed Hassani** [3] **Ramtin Pedarsani** [1]

Abstract

We consider a decentralized stochastic learning problem where data points are distributed among computing nodes communicating over a directed graph. As the model size gets large, decentralized learning faces a major bottleneck that is the heavy communication load due to each node transmitting large messages (model updates) to its neighbors. To tackle this bottleneck, we propose the quantized decentralized stochastic learning algorithm over directed graphs that is based on the push-sum algorithm in decentralized consensus optimization. More importantly, we prove that our algorithm achieves the same convergence rates of the decentralized stochastic learning algorithm with exact-communication for both convex and non-convex losses. Furthermore, our numerical results illustrate significant speed-up compared to the exact-communication methods.

1. Introduction

In modern machine learning applications we typically confront solving optimization problems with massive data sets which demands utilizing multiple computing agents to accelerate convergence. Furthermore, in some applications each processing agent has its own local data set. However, communicating data among different workers is often impractical from multiple aspects such as privacy and bandwidth utilization. In such settings, computing nodes rely on their own data to run (stochastic) gradient descent algorithm while exchanging parameters with other workers in each iteration to ensure converging to the optimal solution of the (global) objective. More precisely, the goal of decentralized learning is to optimize a function $f:\mathbb{R}^d\rightarrow\mathbb{R}$ defined as the average of functions $f_i(\cdot)$, $i\in[n]$ of all computing nodes, i.e.,

$$\widehat{\mathbf{x}} := \arg\min_{\mathbf{x}\in\mathbb{R}^d}\left\{f(\mathbf{x})=\frac{1}{n}\sum_{i=1}^{n}f_i(\mathbf{x})\right\}. \tag{1}$$

Here, each local function $f_i(\cdot)$ can be considered as the loss incurred over the local data-set of node i:

$$f_i(\mathbf{x}) := \frac{1}{m_i}\sum_{j=1}^{m_i}\ell(\mathbf{x},\zeta_j^i), \tag{2}$$

where $\ell:\mathbb{R}^d\times\mathbb{R}^{d'}\rightarrow\mathbb{R}$ is the loss function, m_i is the data-set size of node i, and ζ_j^i denotes the d'-dimensional local data points. Classical decentralized setting consists of two steps. First each computing node i runs (stochastic) gradient descent algorithm on its local function $f_i(\cdot)$ using its local parameter $\mathbf{x}_i(t)$ and local data-set. Then the local parameters are exchanged between neighbor workers to compute a weighted average of neighbor nodes' parameters. Local parameters of every node i at iteration $t+1$ is then obtained by a combination of the weighted average of neighboring nodes solutions and a negative descent direction based on local gradient direction. In particular, if we define $\mathbf{x}_i(t)$ as the decision variable of node i at step t, then its local update can be written as

$$\mathbf{x}_i(t+1)=\sum_{j=1}^{n}a_{ij}\,\mathbf{x}_j(t)-\alpha(t)\nabla F_i\Big(\mathbf{x}_i(t),\zeta_{i,t}\Big). \tag{3}$$

Here $\alpha(t)\geq 0$ is the stepsize and ∇F_i is the stochastic gradient of function f_i evaluated using a random sample of data-set of node i. Matrix A represents the weights used in the averaging step and in particular a_{ij} is the weight that node i assigns to node j. It can be shown that under some conditions for the weight matrix A, the iterates of the update in (3) converge to the optimal solution $\widehat{\mathbf{x}}$ when local functions $f_i(\cdot)$ are convex (Yuan et al., 2016) and to a first-order stationary point in the non-convex case (Lian et al., 2017). Most commonly in these settings, workers communicate over an *undirected* connected graph $\mathcal{G}=\{V,E\}$, and to derive these theoretical results the weight matrix A should have the following properties:

[1]Department of Electrical and Computer Engineering, University of California, Santa Barbara, Santa Barbara, CA, USA. [2]Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA. [3]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. Correspondence to: Hossein Taheri <hossein@ucsb.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

1. A is doubly stochastic, i.e., $\mathbf{1}^T A = \mathbf{1}^T;\ A\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ indicates an n-dimensional vector of all 1's;

2. A is symmetric: $A^T = A$;

3. The spectral gap of A is strictly positive, i.e, the second largest eigenvalue of A satisfies $\lambda_2(A) < 1$.

It can be shown that these assumptions guarantee that the t-th power of A, i.e., A^t, converges to the matrix $\frac{1}{n}\mathbf{1}\mathbf{1}^T$ at a linear rate (i.e., exponentially fast); see, e.g., Lemma 5 in (Lian et al., 2017). This ensures the consensus among different workers in estimating the optimum solution $\widehat{\mathbf{x}}$. However for *directed* graphs, satisfying the first and second constraints are not generally possible. Over the last few years there have been several works to tackle decentralized optimization over directed graphs, e.g., (Blondel et al., 2005) showed that for row-stochastic matrices with positive entries on the diagonal, the matrix A^t converges to $\mathbf{1}\boldsymbol{\phi}^T$ at a linear rate, where $\boldsymbol{\phi}$ is a stochastic vector. Based on this result (Nedić & Olshevsky, 2014) proposed the push-sum algorithm for decentralized optimization over (time-varying) directed graphs. The basic intuition is that the algorithm estimates the vector $\boldsymbol{\phi}$ by a vector $\mathbf{y}$ which is being updated among all workers in each iteration.

However, the mentioned algorithms for decentralized settings over directed graphs require exchanging the agents' model exactly and without any error in order to guarantee convergence to a desired solution. As the model size gets large, e.g., in deep learning algorithms, one can see that communication overhead of each iteration becomes the major bottleneck. Parameter quantization is the major approach to tackle this issue. Although this approach might increase the overall number of iterations, the reduction in the communication cost leads to an efficient algorithm for optimization or gossip problems with large model size. In this paper, we exploit the idea of compressing signals for communication to propose the first *quantized algorithms* for gossip and decentralized learning algorithm over *directed graphs*. The main contributions of this paper are summarized below:

- We propose the first algorithm for communication-efficient gossip type problems and decentralized stochastic learning over directed graphs. Importantly, our algorithm guarantees converging to the optimal solution.

- We prove that our proposed method converges at the same rate as push-sum with exact quantization. In particular for gossip type problems we show convergence in $O(\lambda^T)$. For stochastic learning problems with convex objectives over a directed network with n nodes, we show that the objective loss converges to the optimal solution with the rate $O(\frac{1}{\sqrt{nT}})$. For non-convex objectives we show that squared norm of the gradient converges with the rate $O(\frac{1}{\sqrt{nT}})$, suggesting convergence to a stationary point with the optimal rate.

- The proposed algorithms demonstrate significant speed-up for communication time compared to the exact-communication method for gossip and decentralized learning in our experiments.

Notation. We use boldface notation for vectors and small and large letters for scalars and matrices respectively. M^T denotes the transpose of the matrix M. Mean of rows of a matrix M, is denoted by $\overline{M}$. We use $[n]$ to denote the set of nodes $\{1, 2, \cdots, n\}$. $\|\cdot\|$ denotes the L_2 norm of a matrix or vector depending on its argument. The ith row of the matrix M is represented by $[M]_i$. The identity matrix of size $d \times d$ is denoted by $\mathbf{I}_d$ and the d dimensional zero vector is denoted by $\mathbf{0}_d$. To simplify notation we represent the n-dimensional vector of all 1's as $\mathbf{1}$, where n is the number of computing nodes.

1.1. Prior Work

Gossip and decentralized optimization. The consensus problems over graphs are generally called Gossip problems (Jadbabaie et al., 2003; Saber & Murray, 2003; Xiao & Boyd, 2004). In the gossip type problems, the goal of each node is to reach the average of initial values of all nodes over an undirected graph. Over the last few years there have been numerous research works which consider the convergence of decentralized optimization for undirected graphs (Nedic & Ozdaglar, 2009; Pu et al., 2020; Shi et al., 2015; Yuan et al., 2016). In particular, (Nedic & Ozdaglar, 2009; Yuan et al., 2016) prove the convergence of decentralized algorithm for convex and Lipschitz functions, while (Lian et al., 2017) prove the convergence of stochastic gradient descent for non-convex and smooth loss functions with the rate $O(\frac{1}{\sqrt{nT}})$. (Shi et al., 2015) propose the EXTRA algorithm which using gradient descent converges at a linear rate for strongly-convex losses.

Decentralized learning over directed graphs. The first study of push-sum scheme for gossip type problems in directed graphs is discussed in (Kempe et al., 2003). Authors in (Tsianos et al., 2012) extend this method to decentralized optimization problems and show the convergence of push-sum for convex loss functions. (Nedić & Olshevsky, 2014) extend these results to time-varying directed uniformly strongly connected graphs. Convergence of push-sum protocol for non-convex settings and for asynchronous communication are discussed in (Assran & Rabbat, 2020; Assran et al., 2018). General algorithms for reaching linear convergence rate in decentralized optimization e.g., EXTRA can be combined with the push-sum algorithm to obtain similar results for strongly-convex objective function in directed graphs. See the following works for interesting discussions regarding this topic (Nedic et al., 2017; Xi & Khan, 2015; 2017; Xi et al., 2017; 2018; Xin & Khan, 2018; Zeng & Yin, 2015).

Quantized decentralized learning. Over the last few years there has been a surge of interest in studying communication efficient algorithms for distributed/decentralized optimization (Alistarh et al., 2017; Bernstein et al., 2018; Doan et al., 2018; Karimireddy et al., 2019; Koloskova et al., 2019; Nedic et al., 2009; Reisizadeh et al., 2019a;b; Stich, 2018; Wang et al., 2019). In (Nedic et al., 2009) authors discuss the effect of quantization in decentralized gossip problems using quantization with constant variance. However they show that these algorithms converge to the average of the initial values of the agents within some error. Authors in (Koloskova et al., 2019) propose a quantized algorithm for decentralized optimization over undirected graphs. Here we employ the push-sum protocol to extend this quantized scheme for directed graphs.

2. Network Model

We consider a *directed* graph $\mathcal{G} = \{V, E\}$, where V is the set of nodes and E denotes the set of directed edges of the graph. We say there is a link from node i to node j when $(i, j) \in E$. Indeed, as the graph is directed this does not guarantee that there is also a link from j to i, i.e., $(j, i) \in E$. The sets of in-neighbors and out-neighbors of node i are defined as:

$$\mathcal{N}_i^{\text{in}} := \{j : (j, i) \in E\} \cup \{i\},$$
$$\mathcal{N}_i^{\text{out}} := \{j : (i, j) \in E\} \cup \{i\}.$$

We denote $d_i^{\text{out}} := |\mathcal{N}_i^{\text{out}}|$ to be the out-degree of node i. Throughout this paper we assume that $\mathcal{G}$ is strongly connected.

Assumption 1 (Graph structure)**.** *Graph $\mathcal{G}$ of workers is strongly connected.*

Additionally we assume that the weight matrix has non-negative entries and each node uses its own parameter as well as its in-neighbors. This implies that all vertices of graph $\mathcal{G}$ have self-loops. Also we assume that the weight matrix is column stochastic.

Assumption 2 (Weight matrix)**.** *Matrix A is column stochastic, all entries are non-negative and all entries on the diagonal are positive.*

Given the above assumptions, we next state a key result from (Nedić & Olshevsky, 2014; Zeng & Yin, 2015) that will be useful in our analysis.

Proposition 2.1. *Let Assumptions 1 and 2 hold and let A be the corresponding weight matrix of workers in a graph $\mathcal{G}$. Then, there exist a stochastic vector $\boldsymbol{\phi} \in \mathbb{R}^n$, and constants $0 < \lambda < 1$ and $C > 0$ such that for all $t \geq 0$:*

$$\left\| A^t - \boldsymbol{\phi}\mathbf{1}^T \right\| \leq C\lambda^t. \tag{4}$$

Moreover there exists a constant $\delta > 0$ such that for all $i \in [n]$ and $t \geq 1$

$$\left[A^t \mathbf{1}\right]_i \geq \delta. \tag{5}$$

Note that the column-stochastic property of the weight matrix is considerably weaker than double-stochastic property. As explained in the next example, each computing node i can use its own out-degree to form the i'th column of weight-matrix. Thus the weight matrix can be constructed in the decentralized setting without each node knowing n or the structure of the graph.

Example 1. *Consider a strongly connected network of n computing nodes where $a_{ij} = \frac{1}{d_j^{\text{out}}}$ for all $i, j \in [n]$, and each node has a self-loop. It is straight-forward to see that A is column stochastic and all entries on the diagonal are positive. Therefore the constructed weight matrix satisfies Assumption 2.*

3. Push-sum for Directed Graphs

Before explaining our main contributions on quantized decentralized learning, we discuss gossip or consensus algorithms over directed graphs. Consensus algorithms in the decentralized setting are denoted as gossip algorithms. In this problem, workers are exchanging their parameters $\mathbf{x}_i(t)$ at time t over a connected graph. The goal is to reach the average of initial values of all nodes, i.e., $\bar{X}(1)$, at every node, guaranteeing consensus among workers. The gossip algorithm is based on the weighted average of parameters of neighboring nodes, i.e.,$\mathbf{x}_i(t+1) = \sum_{j=1}^n a_{ij}\mathbf{x}_j(t)$. (Xiao & Boyd, 2004) showed that for doubly stochastic graphs with spectral gap smaller than one, the weight matrix A converges in linear iterations to the average matrix; thus, $X(t+1) = A^t X(1)$ asymptotically converges to $\frac{\mathbf{1}\mathbf{1}^T}{n}X(1)$, which guarantees convergence to the initial mean with linear rate. The condition on A being column-stochastic guarantees that average of workers is preserved in each iteration, i.e., $\bar{X}(t) = \bar{X}(t-1) = \cdots = \bar{X}(1)$. On the other hand, if the weight matrix A is not row-stochastic, $\mathbf{x}_i(t)$ converges to $\phi_i\bar{X}(1)$, where ϕ_i is the ith entry of the stochastic vector $\boldsymbol{\phi}$ with the property that $A^t \to \boldsymbol{\phi}\mathbf{1}^T$. The main approach to tackle consensus in directed networks or for non-doubly stochastic weight matrices is the push-sum protocol introduced for the first time in (Kempe et al., 2003). In the push-sum algorithm each worker i updates its auxiliary scalar variable $y_i(t)$ according to the following rule:

$$y_i(t+1) = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij} y_j(t).$$

Note that the matrix A is column stochastic but not necessarily row-stochastic, thus one can see that if the scalars y_i are initialized with $y_i(1) = 1$, for all $i \in [n]$, then $Y(t) = A^t Y(1) = A^t \mathbf{1}$. This implies that $Y(t) \xrightarrow{t \to \infty} n \cdot \boldsymbol{\phi}$. Thus for all $i \in [n]$,

$$\frac{\mathbf{x}_i(t)}{y_i(t)} \xrightarrow{t \to \infty} \frac{\phi_i \mathbf{1}^T \cdot X(1)}{n \cdot \phi_i} = \bar{X}(1).$$

This shows the asymptotic convergence of $\frac{\mathbf{x}_i}{y_i}$ to the initial mean. Since the parameters $\mathbf{x}_i(t)$ and $y_i(t)$ are kept locally at each node, in every iteration each node can obtain its variable $\mathbf{z}_i(t) := \frac{\mathbf{x}_i(t)}{y_i(t)}$ in the decentralized setting. Based on this approach, we present a communication-efficient algorithm for Gossip over directed networks which uses quantization for reducing communication time (Section 4.1). Moreover, we will show exact convergence with the same rate as the push-sum algorithm without quantization.

3.1. Extension to decentralized optimization

As studied in (Nedić & Olshevsky, 2014; Tsianos et al., 2012) the push-sum method for reaching consensus among nodes can be extended to decentralized convex optimization problems with some modifications. The push-sum algorithm for decentralized optimization with exact communication, can be summarized in the following steps:

$$\begin{cases} \mathbf{x}_i(t+1) & = \sum_{j=1}^{n} a_{ij}\, \mathbf{x}_j(t) - \alpha(t) \nabla f_i\left(\mathbf{z}_i(t)\right) \\ y_i(t+1) & = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij}\, y_j(t) \\ \mathbf{z}_i(t+1) & = \frac{\mathbf{x}_i(t+1)}{y_i(t+1)} \end{cases}$$

Here, local gradients ∇f_i are computed at the scaled parameters $\mathbf{z}_i(t)$, while the parameters $\mathbf{z}_i(t)$ are obtained similar to the gossip push-sum method. It is shown by (Nedić & Olshevsky, 2014) that for all nodes $i \in [n]$ and all $T \geq 1$ the iterates of gossip push-sum method satisfy

$$f\left(\widetilde{\mathbf{z}}_i(T)\right) - f\left(\mathbf{z}^\star\right) \leq O\left(\frac{1}{\sqrt{T}}\right),$$

for $\alpha(t) = O(1/t)$ and $\widetilde{\mathbf{z}}_i(T)$ denoting the weighted time average of $\mathbf{z}_i(t)$ for $t = 1, \cdots, T$. This result indicates that for column stochastic matrices, the push-sum protocol achieves the optimal solution at a sublinear rate of $O(1/\sqrt{T})$. In the following section, we show that one can obtain the similar complexity bound even for the case that nodes exchange quantized signals Section 4.2.

Algorithm 1 Quantized Push-sum for Gossip

for each node $i \in [n]$ and iteration $t \in [T]$ **do**
  $Q_i(t) = Q\left(\mathbf{x}_i(t) - \widehat{\mathbf{x}}_i(t)\right)$
  for all nodes $k \in \mathcal{N}_i^{\text{out}}$ and $j \in \mathcal{N}_i^{\text{in}}$ **do**
    send $Q_i(t)$ and $y_i(t)$ to k and receive $Q_j(t)$ and $y_j(t)$ from j.
    $\widehat{\mathbf{x}}_j(t+1) = \widehat{\mathbf{x}}_j(t) + Q_j(t)$
  end for
  $\mathbf{x}_i(t+1) = \mathbf{x}_i(t) - \widehat{\mathbf{x}}_i(t+1) + \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij} \widehat{\mathbf{x}}_j(t+1)$
  $y_i(t+1) = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij} y_j(t)$
  $\mathbf{z}_i(t+1) = \frac{\mathbf{x}_i(t+1)}{y_i(t+1)}$
end for

4. Quantized Push-sum for Directed Graphs

In this section, we propose two variants of the push-sum method with *quantization* for both gossip (consensus) and decentralized optimization over directed graphs.

4.1. Quantized push-sum for Gossip

We present a quantized gossip algorithm for the consensus problem in which nodes communicate quantized parameters over a directed graph. The steps of our proposed algorithm are described in Algorithm 1. Basically, Algorithm 1 consists of two parts: First, the "Quantization" step, in which each node computes

$$Q_i(t) := Q\left(\mathbf{x}_i(t) - \widehat{\mathbf{x}}_i(t)\right),$$

where $\widehat{\mathbf{x}}_i(t)$ is an auxiliary parameter stored locally at each node and is being updated at each iteration. Importantly every node i, communicates $Q_i(t)$ to its out-neighbors. Quantizing and communicating $\mathbf{x}_i(t) - \widehat{\mathbf{x}}_i(t)$ instead of $\mathbf{x}_i(t)$ is a crucial part of the algorithm as it guarantees that the quantization noise asymptotically vanishes. Second part of the proposed algorithm is the "Averaging" step, in which every node i updates in parallel its parameters $(\mathbf{x}_i(t), y_i(t), \mathbf{z}_i(t))$. The variables $\mathbf{z}_i(t)$ and $y_i(t)$ are updated similar to the push-sum algorithm. For updating $\mathbf{x}_i$, the algorithm uses estimates of the value of $\mathbf{x}_j(t)$ of its in-neighbors, denoted by $\widehat{\mathbf{x}}_j$. Each worker keeps track of the auxiliary parameters of its in-neighbors $\widehat{\mathbf{x}}_j(t)$, for all $j \in \mathcal{N}_i^{\text{in}}$ with updating it with $Q_j(t)$ received from them:

$$\widehat{\mathbf{x}}_j(t+1) = \widehat{\mathbf{x}}_j(t) + Q_j(t).$$

Using the same initialization for all $\widehat{\mathbf{x}}_j(t)$ kept locally in all out-neighbors of node j, one can see that $\widehat{\mathbf{x}}_j(t)$ remains the same among all out-neighbors of node j for all iterations t. Similar to the push-sum protocol with exact quantization, the role of $y_i(t)$ in Algorithm 1 is to scale the parameters $\mathbf{x}_i(t)$ of all nodes i in order to obtain $\mathbf{z}_i(t)$ which is the estimation of the average of initial values of nodes $\bar{X}(1)$.

Algorithm 2 Quantized Decentralized SGD over Directed Graphs

for each node $i \in [n]$ and iteration $t \in [T]$ **do**
  $Q_i(t) = Q\left(\mathbf{x}_i(t) - \widehat{\mathbf{x}}_i(t)\right)$
  for all nodes $k \in \mathcal{N}_i^{\text{out}}$ and $j \in \mathcal{N}_i^{\text{in}}$ **do**
    send $Q_i(t)$ and $y_i(t)$ to k and receive $Q_j(t)$ and $y_j(t)$ from j.
    $\widehat{\mathbf{x}}_j(t+1) = \widehat{\mathbf{x}}_j(t) + Q_j(t)$
  end for
  $\mathbf{w}_i(t+1) = \mathbf{x}_i(t) - \widehat{\mathbf{x}}_i(t+1) + \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij}\widehat{\mathbf{x}}_j(t+1)$
  $y_i(t+1) = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij} y_j(t)$
  $\mathbf{z}_i(t+1) = \frac{\mathbf{w}_i(t+1)}{y_i(t+1)}$
  $\mathbf{x}_i(t+1) = \mathbf{w}_i(t+1) - \alpha \nabla F_i(\mathbf{z}_i(t+1), \zeta_{i,t+1})$
end for

4.2. Quantized push-sum for decentralized optimization

Using the push-sum method for optimization problems we propose Algorithm 2 for communication efficient collaborative optimization over Directed Graphs. Similar to the method described in Algorithm 1, this method also has the Quantization and Averaging steps, with the difference that the update rule for $\mathbf{x}_i(t)$ is similar to one iteration of the stochastic gradient descent:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) - \widehat{\mathbf{x}}_i(t+1) + \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij}\widehat{\mathbf{x}}_j(t+1) \\ -\alpha \nabla F_i(\mathbf{z}_i(t+1), \zeta_{i,t+1}).$$

Importantly, we note that stochastic gradients are evaluated at the scaled values $\mathbf{z}_i(t) = \mathbf{w}_i(t)/y_i(t)$. One can observe that similar to Algorithm 1, here the role of $\mathbf{z}_i(t)$ is to *correct* the parameters $\mathbf{x}_i(t)$ through scaling with $y_i(t)$.

In Section 5, we show that the locally kept parameters $\mathbf{z}_i(t)$ will reach consensus at the rate $O(\frac{1}{\sqrt{T}})$ for $\alpha = O(\frac{1}{\sqrt{T}})$. Furthermore, with the same step size, the time average of the parameters $\mathbf{z}_i(t)$ for $t = 1, \cdots, T$ will converge to the optimal solution for convex losses and it will converge to a stationary point for non-convex losses with the same rate as Decentralized Stochastic Gradient Descent (DSGD) with exact communication. This reveals that quantization and structure of the graph (e.g, directed or undirected) have no effect on the rate of convergence under the proposed algorithm, however, these dependencies on quantization and the structure of graph appear in the terms of the upper bound.

5. Convergence Analysis

In this section, we study convergence properties of our proposed schemes for quantized gossip and decentralized stochastic learning with convex and non-convex objectives. To do so, we first assume the following conditions on the quantization scheme and loss functions are satisfied.

Assumption 3 (Quantization Scheme)**.** *The quantization function $Q : \mathbb{R}^d \to \mathbb{R}^d$ satisfies for all $\boldsymbol{x} \in \mathbb{R}^d$:*

$$\mathbb{E}\left[\left\|Q(\boldsymbol{x}) - \boldsymbol{x}\right\|^2\right] \le \omega^2 \left\|\boldsymbol{x}\right\|^2, \tag{6}$$

where $0 \le \omega < 1$.

In the following, we mention a specific quantization scheme and formally state its parameter ω.

Example 2 (Low-precision Quantizer)**.** *(Alistarh et al., 2017) The unbiased stochastic quantization assigns $\xi_i(\boldsymbol{x}, s)$ to each entry x_i in $\boldsymbol{x}$, where s is the number of levels used for encoding x_i and*

$$\xi_i(\boldsymbol{x}, s) = \begin{cases} (\ell+1)/s & w.p. \ \dfrac{|x_i|}{\|\boldsymbol{x}\|_2}s - \ell\,, \\ \ell/s & otherwise. \end{cases}$$

Here ℓ is the integer satisfying $0 \le \ell < s$ and $\frac{|x_i|}{\|\boldsymbol{x}\|_2} \in [\ell/s, (\ell+1)/s]$. The node at the receiving end, recovers the message according to :

$$Q\left(x_i\right) = \|\boldsymbol{x}\|_2 \cdot \operatorname{sign}\left(x_i\right) \cdot \xi_i(\boldsymbol{x}, s).$$

This quantization scheme satisfies Assumption 3 with

$$\omega^2 = \min\left(d/s^2, \sqrt{d}/s\right).$$

For convenience we assume that the parameters $\mathbf{x}_i$ and $\widehat{\mathbf{x}}_i$ of all nodes are initialized with zero vectors. This assumption is without loss of generality and is taken to simplify the resulting convergence bounds.

Assumption 4 (Initialization)**.** *The parameters $\mathbf{x}_i$ and $\widehat{\mathbf{x}}_i$ are initialized with $\mathbf{0}_d$ and $y_i(1) = 1$ for all $i \in [n]$.*

Additionally we make the following assumptions on the local objective function of each computing node and its stochastic gradient.

Assumption 5 (Lipschitz Local Gradients)**.** *Local functions $f_i(\cdot)$, have L-Lipschitz gradients i.e., for all $i \in [n]$*

$$\left\|\nabla f_i(\mathbf{y}) - \nabla f_i(\mathbf{x})\right\| \le L\left\|\mathbf{y} - \mathbf{x}\right\|, \ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d.$$

Assumption 6 (Bounded Stochastic Gradients)**.** *Local stochastic gradients $\nabla F_i(\mathbf{x}, \zeta_i)$ have bounded second moment i.e., for all $i \in [n]$*

$$\mathbb{E}_{\zeta_i \sim \mathcal{D}_i}\left\|\nabla F_i(\mathbf{x}, \zeta_i)\right\|^2 \le D^2, \ \forall \mathbf{x} \in \mathbb{R}^d.$$

Assumption 7 (Bounded Variance)**.** *Local stochastic gradients have bounded variance i.e., for all $i \in [n]$*

$$\mathbb{E}_{\zeta_i \sim \mathcal{D}_i} \left\| \nabla F_i(\mathbf{x}, \zeta_i) - \nabla f_i(\mathbf{x}) \right\|^2 \leq \sigma^2, \; \forall \mathbf{x} \in \mathbb{R}^d.$$

First, we show that in Algorithm 1, the parameters $\mathbf{z}_i$ of all nodes recover the exact value of initial mean in linear iterations. For convenience we denote by $\gamma := \|A - I\|$ and $\lambda_1 := \frac{1}{2\lambda^{-1/2} + 4C(\lambda - \lambda^{3/2})^{-1}}$.

Theorem 5.1 (Gossip)**.** *Recall the definitions of λ and δ in Proposition 2.1. Under Assumptions 1-3, the iterations of Algorithm 1 satisfy for $\omega \leq \frac{\lambda_1}{1+\gamma}$, $i \in [n]$ and all $t \geq 1$:*

$$\mathbb{E} \left\| \mathbf{z}_i(t+1) - \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i(1) \right\| \leq \frac{4C^2}{\delta} \cdot \frac{\omega \gamma \|X(1)\|}{\lambda - \lambda^{3/2}} \lambda^{t/2} + \frac{2C\|X(1)\|}{\delta} \lambda^t. \tag{7}$$

This bound signifies the effect of parameters related to the structure of graph and weight matrix, i.e. λ, C, δ and γ and the quantization parameter ω. In particular ω emerges as the coefficient of the larger term, and choosing $\omega = 0$ which corresponds to gossip with exact communication, results in convergence with rate $O(\lambda^t)$.

Next, we show that the quantization method for decentralized stochastic learning over directed graphs as described in Algorithm 2 converges to the optimal solution with optimal rates for convex objectives. In particular we show that global objective function evaluated at time average of $\mathbf{z}_i(t)$ converges to the optimal solution after T iterations with the rate $O(\frac{1}{\sqrt{nT}})$. The next theorem characterizes the convergence bound for Algorithm 2 with convex objectives. For compactness we define constants $\lambda_2 := (1 + \frac{6C^2}{(1-\lambda)^2})^{-1/2}$ and $\xi := 6nD^2(1+\gamma^2)(1 + \frac{6C^2}{(1-\lambda)^2})$.

Theorem 5.2 (Convex Objectives)**.** *Assume local functions $f_i(\cdot)$ for all $i \in [n]$ are convex, then under Assumptions 1-7 Algorithm 2 for $\omega \leq \frac{\lambda_2}{\sqrt{6(1+\gamma^2)}}$, $\alpha = \frac{\sqrt{n}}{8L\sqrt{T}}$ and $T \geq 1$ satisfies for all $i \in [n]$:*

$$\mathbb{E} f \left(\frac{1}{T} \sum_{t=1}^{T} \mathbf{z}_i(t+1) \right) - f(\mathbf{z}^\star) \leq \frac{8L(L+1)}{\sqrt{nT}} \|\mathbf{z}^\star\|^2 + \frac{\sigma^2(L+1)}{4\,L\sqrt{nT}} + \frac{nC^2\,(L+1)\left(\frac{L\sqrt{n}}{2\sqrt{T}} + L + 1\right)\left(\xi\omega^2 + 2nD^2\right)}{10\,\delta^2(1-\lambda)^2 L^2\, T}. \tag{8}$$

Remark 1. In the proof of the theorem we show that (see (37)) for fixed arbitrary α the error decays with the rate $O(\frac{1}{\alpha T}) + O(\alpha^2 n) + O(\frac{\alpha}{n})$. The inequality in the statement of the theorem follows by the specified choice of α. More importantly, we highlight that the largest term in (8), i.e., $\frac{1}{\sqrt{T}}$, is directly proportionate to $\frac{1}{\sqrt{n}}$ and σ^2 which emphasizes the impact of the number of workers and mini-batch size in accelerating the speed of convergence. Additionally parameters related to the structure of graph, i.e., λ, C, and δ and the quantization parameter ω, only appear in the terms of order $\frac{1}{T}$ and $\frac{1}{T\sqrt{T}}$ which are asymptotically negligible compared to $\frac{1}{\sqrt{T}}$.

In the next theorem we show convergence of Algorithm 2 with non-convex objectives. Importantly, we demonstrate that the gradient of global function converges to the zero vector with the same optimal rate as in decentralized optimization with exact-communication over undirected graphs(See (Lian et al., 2017)).

Theorem 5.3 (Non-convex Objectives)**.** *Under Assumptions 1-7, Algorithm 2 after sufficiently large iterations ($T \geq 4n$), and for $\omega \leq \frac{\lambda_2}{\sqrt{6(1+\gamma^2)}}$ and $\alpha = \frac{\sqrt{n}}{L\sqrt{T}}$ satisfies:*

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left\| \nabla f \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i(t) \right) \right\|^2 + \frac{1}{2T} \sum_{t=1}^{T} \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{z}_i(t+1)) \right\|^2 \leq \frac{\sigma^2 L}{\sqrt{nT}} + \frac{2L\left(f\left(0\right) - f^\star\right)}{\sqrt{nT}} + \frac{12C^2}{\delta^2(1-\lambda)^2 T}\left(\xi\omega^2 + 2nD^2\right). \tag{9}$$

Moreover for all $i \in [n]$ and $T \geq 1$, it holds that

$$\mathbb{E} \left\| \mathbf{z}_i(T+1) - \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i(T) \right\|^2 \leq \frac{6C^2 n}{\delta^2(1-\lambda)^2 L^2 T}\left(\xi\omega^2 + 2nD^2\right). \tag{10}$$

Remark 2. The inequality (9) implies convergence of the average of $\mathbf{x}_i(t)$ among workers to a stationary point as well as convergence of average of local gradients $\nabla f_i(\mathbf{z}_i(t))$ to zero with the optimal rate $O(\frac{1}{\sqrt{T}})$ for non-convex objectives. Interestingly similar to the convex-case discussed in Remark 1, the number of workers n and stochastic gradient variance σ^2 emerge in the dominant terms while the parameters related to weight matrix, graph structure and quantization appear in the term of order $O(\frac{1}{T})$.

Remark 3. As the proof of theorem shows, for arbitrary fixed step size α we derive the inequality in (44) and the desired result of theorem is concluded by the specified choice of α in the theorem. Importantly we note that the condition on the number of iterations ,i.e. $T \geq 4n$, is a direct result of the specified choice of α. Therefore one can get the same rate

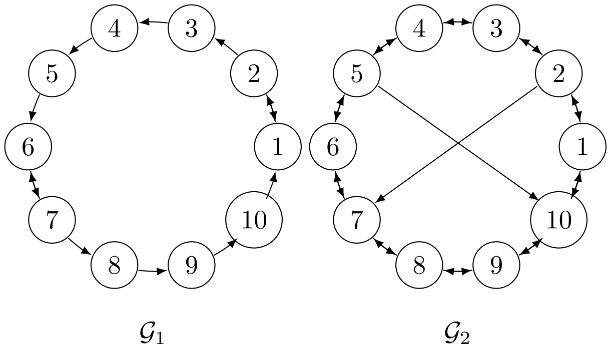

Figure 1. The experimented directed graphs representing communication between computing nodes

for convergence for all $T \geq 1$, with other choices for the step size. For example using the relation (44), by choosing $\alpha = \frac{L}{2\sqrt{T}}$ we achieve convergence for all $T \geq 1$.

Remark 4. Based on (10), the parameters $\mathbf{z}_i$ of nodes reach consensus with the rate $O(\frac{1}{T})$ for the specified value of α. For an arbitrary value of α, consensus is achieved with the rate $O(\alpha^2)$ (see (29)) which implies that smaller values of α will result in faster consensus among the nodes. However due to the term $O(\frac{1}{\alpha T})$ in the convergence of objective to the optimal solution or in the convergence to a stationary point, this fast consensus will be at the cost of slower convergence of objective function in both convex and non-convex cases.

6. Numerical Experiments

In this section, we compare the proposed methods for communication-efficient message passing over directed graphs, with the push-sum protocol using exact communication (e.g., as formulated in (Kempe et al., 2003; Tsianos et al., 2012) for gossip or optimization problems). Throughout the experiments we use the strongly-connected directed graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ with 10 vertices as illustrated in Figure 1. For each graph we form the column-stochastic weight matrix according to the method described in Example 1. In order to study the effect of graph structure we consider $\mathcal{G}_2$ to be more dense with more connection between nodes. For quantization, we use the method discussed in Example 2 with $B = log(s) + 1$ bits used for each entry of the vector (one bit is allocated for the sign). Moreover the norm of transmitted vector and the scalars y_i are transmitted without quantization. In the push-sum protocol with exact communication the entries of the vector $\mathbf{x}_i$ and the scalar y_i are each transmitted with 54 bits.

Gossip experiments. First, we evaluate performance of Algorithm 5.1 for gossip type problems. We initialize the parameters $\mathbf{x}_i(1) \in [0, 1]^{1024}$ of all nodes to be $i.i.d.$ uniformly distributed random variables. Moreover we initialize the auxiliary parameters $\widehat{\mathbf{x}}_i = \mathbf{0}$ and $y_i = 1$ for all $i \in [n]$. In Figure 2(Top) we compare the performance of Algorithm 1 with the push-sum protocol with exact-communication for both networks $\mathcal{G}_1$ and $\mathcal{G}_2$. While Algorithm 1 has almost the same performance as push-sum over $\mathcal{G}_1$, it is outperformed by push-sum over $\mathcal{G}_2$.

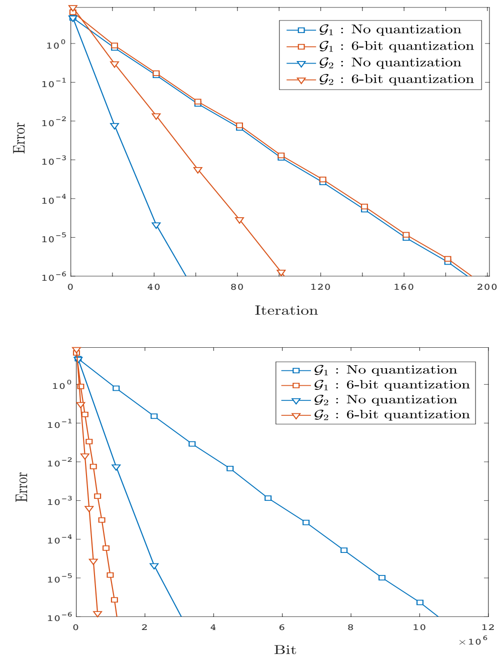

Figure 2. Comparison of the proposed algorithm for the gossip problem and the push-sum protocol using exact-communication based on iteration (Top) and total number of bits communicated between two neighbor nodes (Bottom) over the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$.

However the superiority of exact-communication methods compared in each iteration could be predicted. In order to compare the two methods based on time spent to reach a specific level of error, we compare their performances based on the number of bits that each worker communicates. In Figure 2 (Bottom) the number of bits required to reach a certain error performance is illustrated for both methods. For the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ we observe up to 10x and 6x reduction in the total number of bits, respectively.

Decentralized optimization experiments. Next, we study the performance of Algorithm 2 for decentralized stochastic optimization using convex and non-convex objective functions. First, we consider the objective

$$f(\mathbf{x}) = \frac{1}{2nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| \mathbf{x} - \zeta_j^i \right\|^2,$$

where we set $m = n = 10$ and $d = 256$. Thus each node i has access to its local data-set $\{\zeta_1^i, \zeta_2^i, \cdots, \zeta_{10}^i\}$ and is using one sample at random in each iteration to obtain the

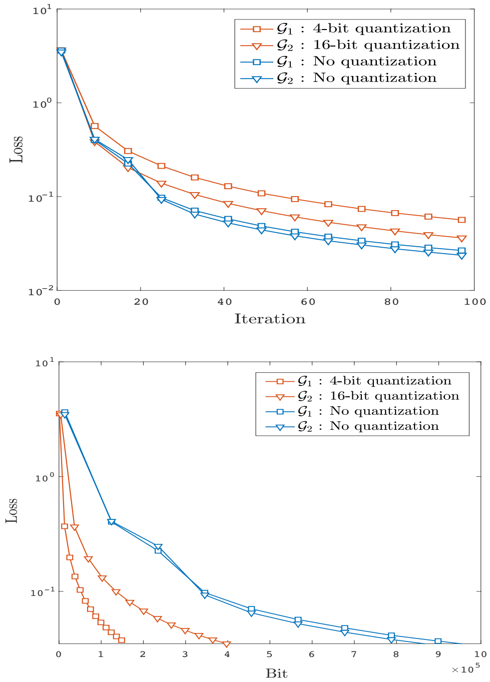

Figure 3. Comparison of the proposed method and exact-communication push-sum method using least-square as objective, based on the iteration number (Top) and total number of communicated bits over the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$.

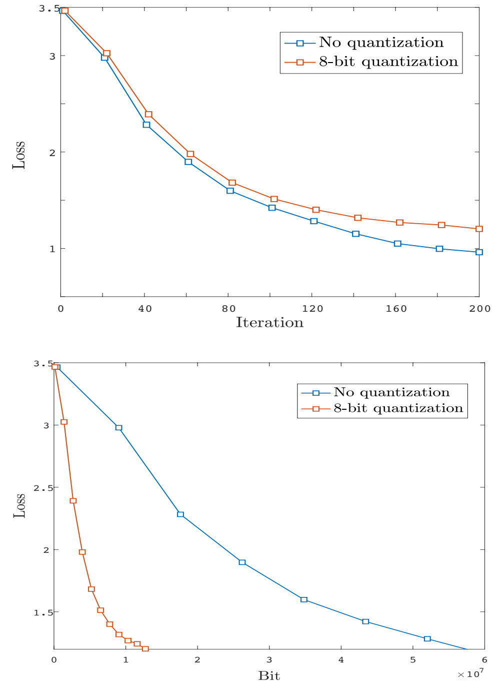

Figure 4. Comparison of the proposed method and exact-communication push-sum method in training a neural network with MNIST data-set, based on the iteration number (Top) and total number of communicated bits (Bottom).

stochastic gradient of its own local function. Here we use the data-set generated according to

$$\zeta_j^i \overset{i.i.d.}{\sim} \zeta^\star + \mathcal{N}\left(\mathbf{0}, \mathbf{I}_{256}\right), \quad \text{for all } i, j,$$

where $\zeta^\star$ is a fixed vector initialized as $\mathrm{Uniform}[0, 100]^{256}$. The step size α for each setting, is fine-tuned up to iteration 50 among 20 values in the interval $[0.01, 3]$. The Loss at iteration T is presented by $\frac{1}{d}\|\widetilde{\mathbf{z}}_1(T) - \zeta_{\text{opt}}\|$, where $\widetilde{\mathbf{z}}_1(T) := \frac{1}{T}\sum_{t=1}^{T} \widetilde{\mathbf{z}}_1(t)$ is the time-average of the model of worker 1 and ζ_{opt} is the optimal solution.

The results of this experiment are in Figure 3 (Top) which illustrates the convergence of Algorithm 2 based on the number of iterations for different levels of quantization and over the two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$. The non-quantized method outperforms the quantized methods based on iteration. This is due to the quantization noise injected in the flow of information over the graph which depends on the number of bits each node uses for encoding and the structure of graph. However this error asymptotically vanishes resulting in small overall quantization noise. This implies that with less quantization noise (i.e., using more bits to encode) the loss decay based on iteration number gets smaller. However as we observe in Figure 3 (Bottom), more quantization levels will result in larger number of bits required to achieve a certain level of loss. Consequently, the push-sum protocol with exact communication for optimization over directed graphs is not communication-efficient as we demonstrated that using smaller number of bits with Algorithm 2 results in 5x reduction in transmitted bits.

As we showed in Theorem 5.3 in Section 5, our proposed method guarantees convergence to a stationary point for non-convex and smooth objective functions. In order to illustrate this, we train a neural-network with 10 hidden units with sigmoid activation function to classify the MNIST data-set into 10 classes. We use the graph $\mathcal{G}_1$ with 10 nodes where each node has access to 1000 samples of data-set and uses a randomly selected mini-batch of size 10 for computing the local stochastic gradient descent. For each setting, the step-size α is fine-tuned up to iteration 200 and over 15 values in the interval $[0.1, 3]$. Figure 4 illustrates the results for training loss of two aforementioned methods based on number of iteration (Top) and total number of bits communicated between two neighbor nodes (Bottom). We note the close gap in each iteration between the loss decay of our proposed method with 8 bits quantization, and the push-sum with exact communication. However since our method uses significantly less bits in each iteration, it reaches the same training loss in fewer iterations. In particular Figure 4 (Bottom) demonstrates 5x reduction in total number of bits communicated using our proposed method.

7. Conclusion and Future Work

In this paper, we proposed a scheme for communication-efficient decentralized learning over directed graphs. We showed that our method converges at the same convergence rate as non-quantized methods for both gossip and decentralized optimization problems. As we demonstrated in Section 6, the proposed approach results in significant improvements in communication-time of the push-sum protocol. An interesting future direction is extending these results to algorithms that achieve linear convergence for strongly-convex problems (e.g., (Xi & Khan, 2017)). Another direction is extending our results to asynchronous decentralized optimization over directed graphs (e.g., as formulated in (Assran & Rabbat, 2020)).

Acknowledgements

This work was supported by National Science Foundation (NSF) under grant CCF-1909320 and UC Office of President under Grant LFR-18-548175.

References

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pp. 1709–1720, 2017.

Assran, M. and Rabbat, M. Asynchronous gradient-push. *IEEE Transactions on Automatic Control*, 2020.

Assran, M., Loizou, N., Ballas, N., and Rabbat, M. Stochastic gradient push for distributed deep learning. *arXiv preprint arXiv:1811.10792*, 2018.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. signsgd: Compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434*, 2018.

Blondel, V. D., Hendrickx, J. M., Olshevsky, A., and Tsitsiklis, J. N. Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 2996–3000. IEEE, 2005.

Doan, T. T., Maguluri, S. T., and Romberg, J. Accelerating the convergence rates of distributed subgradient methods with adaptive quantization. *arXiv preprint arXiv:1810.13245*, 2018.

Jadbabaie, A., Lin, J., and Morse, A. S. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001, 2003.

Karimireddy, S. P., Rebjock, Q., Stich, S. U., and Jaggi, M. Error feedback fixes signsgd and other gradient compression schemes. *arXiv preprint arXiv:1901.09847*, 2019.

Kempe, D., Dobra, A., and Gehrke, J. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pp. 482–491. IEEE, 2003.

Koloskova, A., Stich, S. U., and Jaggi, M. Decentralized stochastic optimization and gossip algorithms with compressed communication. *arXiv preprint arXiv:1902.00340*, 2019.

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

Nedić, A. and Olshevsky, A. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2014.

Nedic, A. and Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Nedic, A., Olshevsky, A., Ozdaglar, A., and Tsitsiklis, J. N. On distributed averaging algorithms and quantization effects. *IEEE Transactions on automatic control*, 54 (11):2506–2517, 2009.

Nedic, A., Olshevsky, A., and Shi, W. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27 (4):2597–2633, 2017.

Pu, S., Shi, W., Xu, J., and Nedic, A. Push-pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, 2020.

Reisizadeh, A., Mokhtari, A., Hassani, H., and Pedarsani, R. An exact quantized decentralized gradient descent algorithm. *IEEE Transactions on Signal Processing*, 67 (19):4934–4947, 2019a.

Reisizadeh, A., Taheri, H., Mokhtari, A., Hassani, H., and Pedarsani, R. Robust and communication-efficient collaborative learning. In *Advances in Neural Information Processing Systems*, pp. 8386–8397, 2019b.

Saber, R. O. and Murray, R. M. Consensus protocols for networks of dynamic agents. 2003.

Shi, W., Ling, Q., Wu, G., and Yin, W. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

Stich, S. U. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Tsianos, K. I., Lawlor, S., and Rabbat, M. G. Push-sum distributed dual averaging for convex optimization. In *2012 ieee 51st ieee conference on decision and control (cdc)*, pp. 5453–5458. IEEE, 2012.

Wang, J., Tantia, V., Ballas, N., and Rabbat, M. Slowmo: Improving communication-efficient distributed sgd with slow momentum. *arXiv preprint arXiv:1910.00643*, 2019.

Xi, C. and Khan, U. A. On the linear convergence of distributed optimization over directed graphs. *arXiv preprint arXiv:1510.02149*, 2015.

Xi, C. and Khan, U. A. Dextra: A fast algorithm for optimization over directed graphs. *IEEE Transactions on Automatic Control*, 62(10):4980–4993, 2017.

Xi, C., Wu, Q., and Khan, U. A. On the distributed optimization over directed networks. *Neurocomputing*, 267:508–515, 2017.

Xi, C., Mai, V. S., Xin, R., Abed, E. H., and Khan, U. A. Linear convergence in optimization over directed graphs with row-stochastic matrices. *IEEE Transactions on Automatic Control*, 63(10):3558–3565, 2018.

Xiao, L. and Boyd, S. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

Xin, R. and Khan, U. A. A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Systems Letters*, 2(3):315–320, 2018.

Yuan, K., Ling, Q., and Yin, W. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Zeng, J. and Yin, W. Extrapush for convex smooth decentralized optimization over directed networks. *arXiv preprint arXiv:1511.02942*, 2015.