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/ICASSP39728.2021.9415052

DECENTRALIZED OPTIMIZATION ON TIME-VARYING DIRECTED GRAPHS UNDER COMMUNICATION CONSTRAINTS

Yiyue Chen[†], *Abolfazl Hashemi*[‡], *and Haris Vikalo*[†]

[†]Department of Electrical and Computer Engineering, UT Austin, TX, USA
[‡]Oden Institute for Computational Engineering and Sciences, UT Austin, TX, USA

ABSTRACT

We consider the problem of decentralized optimization where a collection of agents, each having access to a local cost function, communicate over a time-varying directed network and aim to minimize the sum of those functions. In practice, the amount of information that can be exchanged between the agents is limited due to communication constraints. We propose a communication-efficient algorithm for decentralized convex optimization that rely on sparsification of local updates exchanged between neighboring agents in the network. In directed networks, message sparsification alters column-stochasticity – a property that plays an important role in establishing convergence of decentralized learning tasks. We propose a decentralized optimization scheme that relies on local modification of mixing matrices, and show that it achieves $\mathcal{O}(\frac{\ln T}{\sqrt{T}})$ convergence rate in the considered settings. Experiments validate theoretical results and demonstrate efficacy of the proposed algorithm.

Index Terms— decentralized optimization, convex programming

1. INTRODUCTION

In recent years, decentralized optimization has attracted considerable interest from the machine learning, signal processing, and control communities [1, 2, 3, 4]. We consider the setting where a collection of agents attempts to minimize an objective that consists of functions distributed among the agents; each agent evaluates one of the functions on its local data. Formally, this optimization task can be stated as

$$\min_{\mathbf{x}\in\mathbb{R}^d}\left[f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})\right], \qquad (1)$$

where n is the number of agents and $f_i : \mathbb{R}^d \to \mathbb{R}$ is the function assigned to the i^{th} node, $i \in [n] := \{1, ..., n\}$. The agents collaborate by exchanging information over a network modeled by a time-varying directed graph $\mathcal{G}(t) = (|n|, \mathcal{E}(t))$, where $\mathcal{E}(t)$ denotes the set of edges at time t; agent i can send a message to agent j at time t if there exist an edge from i to j at t, i.e., if $\{i, j\} \in \mathcal{E}(t)$.

The described setting has been a subject of extensive studies over the last decade, leading to a number of seminal results [5, 6, 7, 8, 9, 10, 11]. Majority of prior work assumes symmetry in the agents' communication capabilities, i.e., models the problem using undirected graphs. However, the assumption of symmetry is often violated and the graph that captures properties of the communication network should be directed. Providing provably convergent decentralized convex optimization schemes over directed graphs is challenging; technically, this stems from the fact that unlike in undirected graphs, the so-called mixing matrix of a directed graph is not doubly stochastic. The existing prior work in the directed graph settings includes the grad-push algorithm [12, 3], which compensates for the imbalance in a column-stochastic mixing matrix by relying on local normalization scalars, and the directed distributed gradient descent (D-DGD) scheme [13] which carefully tracks link changes over time and their impact on the mixing matrices. Assuming convex local function, both of these methods achieve $\mathcal{O}(\frac{\ln T}{\sqrt{T}})$ convergence rate.

In practice, communication bandwidth is often limited and thus the amount of information that can be exchanged between the agents is restricted. This motivates design of decentralized optimization schemes capable of operating under communication constraints; none of the aforementioned methods considers such settings. Recently, techniques that address communication constraints in decentralized optimization by quantizing or sparsifying messages exchanged between participating agents have been proposed in literature [14, 15, 10]. Such schemes have been deployed in the context of decentralized convex optimization over undirected networks [11] as well as in *fixed* directed networks [16]. However, there has been no prior work on communication-constrained decentralized learning over time-varying directed networks.

In this paper we propose, to our knowledge the first, communication-sparsifying scheme for decentralized convex optimization over *directed* networks, and provide formal guarantees of its convergence; in particular, we show that the proposed method achieves $\mathcal{O}(\frac{\ln T}{\sqrt{T}})$ convergence rate. Experiments demonstrate efficacy of the proposed scheme.

This work was supported in part by NSF grant 1809327.

2. PROBLEM SETTING

Assume that a collection of agents aims to collaboratively find the unique solution to decentralized convex optimization (1); let us denote this solution by $\mathbf{x}^*$ and assume, for simplicity, that $\mathcal{X} = \mathbb{R}^d$. The agents, represented by nodes of a directed time-varying graph, are allowed to exchange sparsified messages. In the following, we do not assume smoothness or strong convexity of the objective; however, our analysis can be extended to such settings.

Let W_{in}^t (row-stochastic) and W_{out}^t (column-stochastic) denote the in-neighbor and out-neighbor connectivity matrix at time t, respectively. Moreover, let $\mathcal{N}_{in,i}^t$ be the set of nodes that can send information to node i (including i), and $\mathcal{N}_{out,j}^t$ the set of nodes that can receive information from node j (including j) at time t. We assume that both $\mathcal{N}_{in,i}^t$ and $\mathcal{N}_{out,i}^t$ are known to node i. A simple policy for designing W_{in}^t and W_{out}^t is to set

$$[W_{in}^t]_{ij} = 1/|\mathcal{N}_{in,i}^t|, \quad [W_{out}^t]_{ij} = 1/|\mathcal{N}_{out,j}^t|. \tag{2}$$

We assume that the constructed mixing matrices have non-zero spectral gaps; this is readily satisfied in a variety of settings including when the union graph is jointly-connected. Matrices W_{in}^t and W_{out}^t can be used to synthesize the mixing matrix, as formally stated in Section 3 (see Definition 1).

To reduce the size of the messages exchanged between agents in a network, we perform *sparsification*. In particular, each node uniformly at random selects and communicates k out of d entries of a d-dimensional message. To formalize this, we introduce a sparsification operator $Q : \mathbb{R}^d \to \mathbb{R}^d$. The operator Q is biased, i.e., $\mathbb{E}[Q(\mathbf{x})] \neq \mathbf{x}$, and has variance that depends on the norm of its argument, $\mathbb{E}[\|Q(\mathbf{x}) - \mathbf{x}\|^2] \propto \|\mathbf{x}\|^2$. Biased compression operators have previously been considered in the context of time-invariant networks [10, 11, 17, 16] but are not encountered in time-varying network settings.

3. COMPRESSED TIME-VARYING DECENTRALIZED OPTIMIZATION

A common strategy to solving decentralized optimization problems is to orchestrate exchange of messages between agents such that each message consists of a combination of compressed messages from neighboring nodes and a gradient noise term. The gradient term is rendered vanishing by adopting a decreasing stepsize scheme; this ensures that the agents in the network reach a consensus state which is the optimal solution to the optimization problem.

To meet communication constraints, messages may be sparsified; however, simplistic introduction of sparsification to the existing methods, e.g., [12, 18, 19, 3], may have adverse effect on their convergence – indeed, modified schemes may only converge to a neighborhood of the optimal solution or even end up diverging. This is caused by the non-vanishing error due to the bias and variance of the sparsification operator. We note that the impact of sparsification on the entries of a state vector in the network can be interpreted as that of link failures; this motivates us to account for it in the structure of the connectivity matrices. Specifically, we split the vector-valued decentralized problem into d individual scalar-valued sub-problems with the coordinate in-neighbor and out-neighbor connectivity matrices, $\{W_{in,m}^t\}_{m=1}^d$ and $\{W_{out,m}^t\}_{m=1}^d$, specified for each time t. If an entry is sparsified at time t (i.e., set to zero and not communicated), the corresponding coordinate connectivity matrices are no longer stochastic. To handle this issue, we *re-normalize* the connectivity matrices $\{W_{in,m}^t\}_{m=1}^d$ and $\{W_{out,m}^t\}_{m=1}^d$, ensuring their row stochasticity and column stochasticity, respectively; node i performs re-normalization of the i^{th} row of $\{W_{in,m}^t\}_{m=1}^d$ and i^{th} column of $\{W_{out,m}^t\}_{m=1}^d$ locally. We denote by $\{A_m^t\}_{m=1}^d$ and $\{B_m^t\}_{m=1}^d$ the weight matrices resulting from the re-normalization of $\{W_{in,m}^t\}_{m=1}^d$ and $\{W_{out,m}^t\}_{m=1}^d$, respectively.

Following the work of [18] on average consensus, we introduce an auxiliary vector $\mathbf{y}_i \in \mathbb{R}^d$ for each node. Referred to as the surplus vector, $\mathbf{y}_i \in \mathbb{R}^d$ records variations of the state vectors over time and is used to help ensure the state vectors approach the consensus state. At time step t, node i compresses $\mathbf{x}_i^t$ and $\mathbf{y}_i^t$ and sends both to the current out-neighbors. To allow succinct expression of the update rule, we introduce $\mathbf{z}_i^t \in \mathbb{R}^d$ defined as

$$\mathbf{z}_i^t = \begin{cases} \mathbf{x}_i^t, & i \in \{1, ..., n\} \\ \mathbf{y}_{i-n}^t, & i \in \{n+1, ..., 2n\}. \end{cases} \tag{3}$$

The sparsification operator $Q(\cdot)$ is applied to $\mathbf{z}_i^t$, resulting in $Q(\mathbf{z}_i^t)$; we denote the m^{th} entry of the sparsified vector by $[Q(\mathbf{z}_i^t)]_m$. The aforementioned weight matrix A_m^t is formed as

$$[A_m^t]_{ij} = \begin{cases} \frac{[W_{in,m}^t]_{ij}}{\sum_{j \in \mathcal{S}_m^t(i,j)} [W_{in,m}^t]_{ij}} & \text{if } j \in \mathcal{S}_m^t(i,j) \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\mathcal{S}_m^t(i,j) := \{j | j \in \mathcal{N}_{in,i}^t, [Q(\mathbf{z}_j^t)]_m \neq 0\} \cup \{i\}$. Likewise, B_m^t is defined as

$$[B_m^t]_{ij} = \begin{cases} \frac{[W_{out,m}^t]_{ij}}{\sum_{i \in \mathcal{T}_m^t(i,j)} [W_{out,m}^t]_{ij}} & \text{if } i \in \mathcal{T}_m^t(i,j) \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where $\mathcal{T}_m^t(i,j) := \{i | i \in \mathcal{N}_{out,j}^t, [Q(\mathbf{z}_i^t)]_m \neq 0\} \cup \{j\}$.

To obtain the update rule for the optimization algorithm, we first need to define the *mixing matrix* of a directed network with sparsified messages.

Definition 1. *At time t, the m^{th} mixing matrix of a time-varying directed network deploying sparsified messages, $\bar{M}_m^t \in \mathbb{R}^{2n \times 2n}$, is a matrix with eigenvalues $1 = |\lambda_1(\bar{M}_m^t)| = |\lambda_2(\bar{M}_m^t)| \geq |\lambda_3(\bar{M}_m^t)| \geq \cdots |\lambda_{2n}(\bar{M}_m^t)|$ that is constructed from the current network topology as*

$$\bar{M}_m^t = \begin{bmatrix} A_m^t & \mathbf{0} \\ I - A_m^t & B_m^t \end{bmatrix}, \tag{6}$$

Algorithm 1 Communication-Sparsifying Jointly-Connected Gradient Descent

Input: T, ϵ, $\mathbf{x}^0$, $\mathbf{y}^0 = \mathbf{0}$,
set $\mathbf{z}^0 = [\mathbf{x}^0; \mathbf{y}^0]$
for each $t \in [0, 1, ..., T]$ **do**
  generate non-negative matrices W_{in}^t, W_{out}^t
  for each $m \in [1, ..., d]$ **do**
    construct a row-stochastic A_m^t and a column-stochastic B_m^t according to (4) and (5)
    construct $\bar{M}_m^t$ according to (6)
    for each $i \in \{1, ..., 2n\}$ **do**
      Update z_{im}^{t+1} according to (7)
    end for
  end for
end for

where A_m^t and B_m^t represent the m^{th} normalized in-neighbor and out-neighbor connectivity matrices at time t, respectively.

With $\mathbf{z}_i^t$ and $\bar{M}_m^t$ defined in (3) and (6), respectively, node i updates the m^{th} component of its message according to

$$z_{im}^{t+1} = \sum_{j=1}^{2n} [\bar{M}_m^t]_{ij} [Q(\mathbf{z}_j^t)]_m + \mathbb{1}_{\{t \bmod \mathcal{B} = \mathcal{B}-1\}} \epsilon [F]_{ij} z_{jm}^{\mathcal{B}\lfloor t/\mathcal{B} \rfloor} - \mathbb{1}_{\{t \bmod \mathcal{B} = \mathcal{B}-1\}} \alpha_{\lfloor t/\mathcal{B} \rfloor} g_{im}^{\mathcal{B}\lfloor t/\mathcal{B} \rfloor}, \tag{7}$$

where g_{im}^t denotes the m^{th} entry of the gradient vector $\mathbf{g}_i^t$ constructed as

$$\mathbf{g}_i^t = \begin{cases} \nabla f_i(\mathbf{x}_i^t), & i \in \{1, ..., n\} \\ \mathbf{0}, & i \in \{n+1, ..., 2n\}. \end{cases} \tag{8}$$

Moreover, $F = \begin{bmatrix} \mathbf{0} & I \\ \mathbf{0} & -I \end{bmatrix}$, and α_t is the stepsize at time t.

In (7), the update of vectors $\mathbf{z}_i^t$ consists of a mixture of the compressed state vectors and surplus vectors, and includes a vanishing gradient computed from history. The mixture of compressed messages can be interpreted as obtained by sparsification and multiplication with the mixing matrix from the previous time steps, except for the times when

$$t \mod \mathcal{B} = \mathcal{B} - 1. \tag{9}$$

When t satisfies (9), the update of $\mathbf{z}_i^t$ incorporates stored vectors $\mathbf{z}_i^{\mathcal{B}\lfloor t/\mathcal{B} \rfloor}$. Note that $\mathbf{z}_i^{\mathcal{B}\lfloor t/\mathcal{B} \rfloor}$ is multiplied by ϵF, where the perturbation parameter ϵ determines the extent F affects the update. One can show that ϵF, in combination with the mixing matrix $\bar{M}_m^t$, guarantees non-zero spectral gap of the product matrix over $\mathcal{B}$ consecutive time steps starting from $t = k\mathcal{B}$. Similarly, gradient term $\alpha_{\lfloor t/\mathcal{B} \rfloor} g_{im}^{\mathcal{B}\lfloor t/\mathcal{B} \rfloor}$, computed using state vectors $\mathbf{x}_i^{t-(\mathcal{B}-1)}$, participates in the update when (9) holds. We formalize the proposed procedure as Algorithm 1.

Remark. It is worth pointing out that in Algorithm 1 each node needs to store local messages of size $4d$ (four d-dimensional vectors: the current state and surplus vectors, past surplus vector, and local gradient vector). Only the two current vectors may be communicated to the neighboring nodes while the other two vectors are used locally when (9) holds. Note that $\bar{M}_m^t$ has column sum equal to one but it is not column-stochastic due to having negative entries. Finally, note that when $\mathcal{B} = 1$, the network is strongly connected at all times.

3.1. Convergence Analysis

Let $\bar{M}_m(T : s) = \bar{M}_m^T \bar{M}_m^{T-1} \cdots \bar{M}_m^s$ denote the product of a sequence of consecutive mixing matrices from time s to T, with the superscript indicating the time and the subscript indicating the entry position. The perturbed product, $M_m((k+1)\mathcal{B} - 1 : k\mathcal{B})$, is obtained from adding the perturbation term ϵF to the product of mixing matrices as

$$M_m((k+1)\mathcal{B}-1 : k\mathcal{B}) = \bar{M}_m((k+1)\mathcal{B}-1 : k\mathcal{B}) + \epsilon F. \tag{10}$$

To proceed, we require the following assumptions.

Assumption 1. *The mixing matrices, stepsizes, and the local objectives satisfy:*

(i) $\forall k \geq 0, 1 \leq m \leq d$, *there exists some* $0 < \epsilon_0 < 1$ *such that the perturbed product,* $M_m((k+1)\mathcal{B} - 1 : k\mathcal{B})$ *has a non-zero spectral gap* $\forall \epsilon$ *such that* $0 < \epsilon < \epsilon_0$.

(ii) For a fixed $\epsilon \in (0, 1)$, *the set of all possible mixing matrices* $\{\bar{M}_m^t\}$ *is a finite set.*

(iii) The sequence of stepsizes, $\{\alpha_t\}$, *is non-negative and satisfies* $\sum_{t=0}^{\infty} \alpha_t = \infty$, $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$.

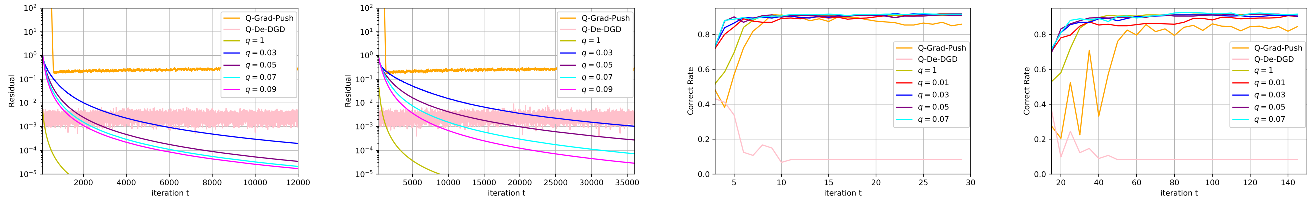
(iv) $\forall 1 \leq i \leq n, 1 \leq m \leq d, t \geq 0$, *there exists some* $D > 0$ *such that* $|g_{im}^t| < D$.

Given the weight matrices scheme in (2), assumptions (i) and (ii) hold for a variety of network structures. Assumptions (iii) and (iv) are common in decentralized optimization [5, 3, 13]) and help guide nodes in the network to a consensus that approaches the global optimal solution. We formalize our main theoretical results in Theorem 1, which establishes convergence of Algorithm 1 to the optimal solution. Proof of the theorem is omitted for brevity (please see [20] for details).

Theorem 1. *Suppose Assumption 1 holds. Let* $\mathbf{x}^*$ *be the unique optimal solution and* $f^* = f(\mathbf{x}^*)$. *Then*

$$2 \sum_{k=0}^{\infty} \alpha_k (f(\bar{\mathbf{z}}^{k\mathcal{B}}) - f^*) \leq n \|\bar{\mathbf{z}}^0 - \mathbf{x}^*\| + nD'^2 \sum_{k=0}^{\infty} \alpha_k^2 + \frac{4D'}{n} \sum_{i=1}^{n} \sum_{k=0}^{\infty} \alpha_k \|\mathbf{z}_i^{k\mathcal{B}} - \bar{\mathbf{z}}^{k\mathcal{B}}\|, \tag{11}$$

where $D' = \sqrt{d} D$ *and* $\bar{\mathbf{z}}^t = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^t + \frac{1}{n} \sum_{i=1}^{n} \mathbf{y}_i^t$.



(a) Residual: $\mathcal{B} = 1$ (b) Residual: $\mathcal{B} = 3$ (c) Correct rate: $\mathcal{B} = 1$ (d) Correct rate: $\mathcal{B} = 5$

Fig. 1. Linear regression on a jointly connected network with $\mathcal{B} = 1, 3, \epsilon = 0.05$, see (a), (b); logistic regression on a jointly connected network with $\mathcal{B} = 1, 5, \epsilon = 0.01$, see (c), (d).

Note that since $\sum_{t=0}^{\infty} \alpha_t = \infty$, it is straightforward to see that Theorem 1 implies $\lim_{t\to\infty} f(\mathbf{z}_i^t) = f^*$ for every agent i, thereby establishing convergence of Algorithm 1 to the global minimum of (1). Additionally, for the stepsize $\alpha_t = \mathcal{O}(1/\sqrt{t})$, Algorithm 1 attains the convergence rate $\mathcal{O}(\frac{\ln T}{\sqrt{T}})$.

4. NUMERICAL SIMULATIONS

We test Algorithm 1 in applications to linear and logistic regression, and compare the results to Q-Grad-Push, obtained by applying simple quantization to the push-sum scheme [3], and Q-De-DGD [16]. Neither of these two schemes was developed with communication-constrained optimization over time-varying directed networks in mind – the former was originally proposed for unconstrainted communication, while the latter is concerned with static networks. However, since there is no prior work on decentralized optimization over time-varying directed networks under communication constraints, we adopt them for the purpose of benchmarking.

We use Erdős–Rényi model to generate strongly connected instances of a graph with 10 nodes and edge appearance probability 0.9. Two uni-directional edges are dropped randomly from each such graph while still preserving strong connectivity. We then remove in-going and out-going edges of randomly selected nodes to create a scenario where an almost-surely strongly connected network is formed only after taking a union of graphs over $\mathcal{B}$ time instances (see Assumption 1). Finally, recall that q denotes the fraction of entries that nodes communicate to their neighbors (small q implies high compression).

Decentralized linear regression. First, consider the optimization problem $\min_{\mathbf{x}} \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{y}_i - D_i\mathbf{x}_i\|^2$, where $D_i \in \mathbb{R}^{200\times 128}$ is a local data matrix with 200 data points of size $d = 128$ at node i, and $\mathbf{y}_i \in \mathbb{R}^{200}$ represents the local measurement vector at node i. We generate $\mathbf{x}^*$ from a normal distribution, and set up the measurement model as $\mathbf{y}_i = M_i\mathbf{x}^* + \eta_i$, where M_i is randomly generated from the standard normal distribution; finally, the rows of the data matrix are normalized to sum to one. The local additive noise η_i is generated from a zero-mean Gaussian distribution with variance 0.01. In Algorithm 1 and Q-Grad-Push, local vectors are initialized randomly to $\mathbf{x}_i^0$; Q-De-DGD is initialized with an all-zero vector. The quantization level of the benchmarking algorithms is selected to ensure that the number of bits those algorithms communicate is equal to that of Algorithm 1 when $q = 0.09$. All algorithms are run with stepsize $\alpha_t = \frac{0.2}{t}$. The performance of different schemes is quantified by the residuals $\frac{\|\mathbf{x}^t - \bar{\mathbf{x}}\|}{\|\mathbf{x}^0 - \bar{\mathbf{x}}\|}$. The results are shown in Fig.1 (a), (b). As shown in the subplots, for all the considered sparsification rates Algorithm 1 converges with rate proportional to q, while the benchmarking algorithms do not converge to the optimal solution.

Decentralized logistic regression. Next, we consider a multi-class classification task on the MNIST dataset [21]. The logistic regression problem is formulated as $\min_{\mathbf{x}} \left\{ \frac{\mu}{2}\|\mathbf{x}\|^2 + \sum_{i=1}^{n}\sum_{j=1}^{N} \ln(1 + \exp(-(\mathbf{m}_{ij}^T\mathbf{x}_i)y_{ij})) \right\}$. The data is distributed across the network such that each node i has access to $N = 120$ training samples $(\mathbf{m}_{ij}, y_{ij}) \in \mathbb{R}^{64} \times \{0, \cdots, 9\}$, where $\mathbf{m}_{ij}$ denotes a vectorized image with size $d = 64$ and y_{ij} denotes the corresponding digit label. Performance of Algorithm 1 is again compared with Q-Grad-Push and Q-De-DGD; all algorithms are initialized with zero vectors. The quantization level of the benchmarking algorithms is selected such that the number of bits they communicate is equal to that of Algorithm 1 for $q = 0.07$. The experiment is run using the stepsize $\alpha_t = \frac{0.02}{t}$; we set $\mu = 10^{-5}$. Fig. 1 (c), (d) show the classification correct rate of Algorithm 1 for different sparsification and connectivity levels. As can be seen there, all sparsified schemes achieve the same level of the classification correct rate. The schemes communicating fewer messages in less connected networks converge slower, while the two benchmarking algorithms converge only to a neighborhood of the optimal solution.

5. CONCLUSION

We considered the problem of decentralized learning over time-varying directed graphs where, due to communication constraints, nodes communicate sparsified messages. We proposed a communication-efficient algorithm that achieves $\mathcal{O}(\frac{\ln T}{\sqrt{T}})$ convergence rate for general decentralized convex optimization tasks. As part of the future work, it is of interest to study reduction of the computational cost of the optimization procedure by extending the results to the setting where network agents rely on stochastic gradients.

6. REFERENCES

[1] Wei Ren and Randal W Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[2] Wei Ren, Randal W Beard, and Ella M Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

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

[4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

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

[6] Björn Johansson, Maben Rabi, and Mikael Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1157–1170, 2010.

[7] Ermin Wei and Asuman Ozdaglar, "Distributed alternating direction method of multipliers," in *2012 IEEE 51st IEEE Conference on Decision and Control*. IEEE, 2012, pp. 5445–5450.

[8] John C Duchi, Alekh Agarwal, and Martin J Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2011.

[9] Angelia Nedić, Soomin Lee, and Maxim Raginsky, "Decentralized online optimization with global objectives and local communication," in *2015 American Control Conference*. IEEE, 2015, pp. 4497–4503.

[10] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi, "Sparsified sgd with memory," in *Advances in Neural Information Processing Systems*, 2018, pp. 4447–4458.

[11] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Int'l Conf. on Machine Learning*, 2019, pp. 3478–3487.

[12] David Kempe, Alin Dobra, and Johannes Gehrke, "Gossip-based computation of aggregate information," in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings* . IEEE, 2003, pp. 482–491.

[13] Chenguang Xi, Qiong Wu, and Usman A Khan, "On the distributed optimization over directed networks," *Neurocomputing*, vol. 267, pp. 508–515, 2017.

[14] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 1509–1519.

[15] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang, "Zipml: Training linear models with end-to-end low precision, and a little bit of deep learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 4035–4043.

[16] Hossein Taheri, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani, "Quantized decentralized stochastic learning over directed graphs," in *International Conference on Machine Learning*, 2020.

[17] Anastasia Koloskova, Tao Lin, Sebastian U Stich, and Martin Jaggi, "Decentralized deep learning with arbitrary communication compression," *arXiv preprint arXiv:1907.09356*, 2019.

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

[19] Kai Cai and Hideaki Ishii, "Average consensus on arbitrary strongly connected digraphs with time-varying topologies," *IEEE Trans. on Automatic Control*, vol. 59, no. 4, pp. 1066–1071, 2014.

[20] Yiuye Chen, Abolfazl Hashemi, and Haris Vikalo, "Communication-efficient decentralized optimization over time-varying directed graphs," *arXiv preprint arXiv:2005.13189*, 2020.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.