PLIABLE DATA SHUFFLING FOR ON-DEVICE DISTRIBUTED LEARNING

Tao Jiang, Kai Yang, and Yuanming Shi

School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
E-mail: {jiangtao1, yangkai, shiym}@shanghaitech.edu.cn

ABSTRACT

Dataset reshuffling across mobile devices allows for speeding up on-device distributed machine learning, which however requires significant communication bandwidth. In this paper, we propose a *pliable data shuffling* approach to significantly reduce the communication cost for on-device distributed learning via joint data placement and transmission design. This is achieved by establishing the novel interference alignment conditions and diversity constraints for data shuffling to improve the statistical learning performance. Unfortunately, the presented pliable data shuffling problem is a highly intractable mixed combinatorial optimization problem, for which a novel sparse and low-rank framework is developed, supported by the computationally efficient difference-of-convex (DC) algorithm. Numerical results demonstrate that the proposed pliable data shuffling is able to significantly reduce the communication bandwidth while achieving desirable learning performance.

***Index Terms*—** Distributed learning, data shuffling, wireless, pliable index coding, sparse and low rank.

1. INTRODUCTION

With the explosive growth in the volume of data for machine learning, deploying algorithms on distributed workers has become the prevalent choice in practice due to their substantial speedups gains [1], supported by the state-of-art distributed machine learning frameworks (e.g., Tensorflow [2]). In a typical distributed machine learning setting, i.e., master-worker architecture, the master aggregates local model parameters from all the workers and then sends the aggregation results to each worker periodically, it is shown that reshuffling the training data across the workers allows for superior convergence performance and better prediction accuracy [3, 4]. Furthermore, the rapid development of mobile edge computing [5] with the growing computation and storage power of devices, provides opportunities for training a machine learning model distributed among a large number of mobile devices. This also allows for privacy preservation [6] since we do not need to send all the collected data to a centralized cloud center. However, wireless network bandwidth becomes the significant bottleneck for distributed training on mobile devices with low throughput, high latency, and poor network connections. In this paper, we shall investigate a novel data shuffling strategy for on-device distributed machine learning inspired by the wireless pliable index coding [7], thereby significantly reducing the communication cost.

Although data shuffling among devices brings statistical benefits in terms of convergence rate and prediction accuracy, this performance improvements come at a cost. Specifically, for each data shuffling procedure, the entire dataset is communicated from the master to each worker, which results in a huge communication overhead. To tackle this issue, index coding has been proposed to improve the communication efficiency of data shuffling [3], which is achieved by designing efficient data transmission schemes given the data placement rules. In this scenario, the specific data point needs to be delivered successfully for data shuffling based on index coding [8]. However, in many distributed learning tasks, e.g., classification, the statistical learning performance can be improved as long as each worker are refreshed with a new data point [9], instead of delivering the specific data points like [3]. This key observation offers the possibility to further reduce the communication cost for data shuffling based on the principles of pliable index coding [10]. However, the finite field pliable index coding approaches [10, 9] may not be generalized to the wireless communication scenarios for on-device distributed learning, where the transceivers are normally operated in complex field [7]. In this paper, we instead propose a pliable index coding approach for joint data placement and transmission scheme design for wireless on-device distributed learning. This is achieved by establishing a novel pliable interference alignment condition to deliver data, as well as deriving a diversity constraint to avoid similarity among the shuffled data, thereby achieving high communication efficiency with comparable statistical performance.

However, the pliable data shuffling problem turns out to be a highly intractable mixed combinatorial optimization problem. Inspired by the recent success of generalized sparse and low-rank models for wireless index coding problem [11, 12] and wireless pliable index coding problem [7], we shall propose a novel sparse and low rank optimization framework to solve the mixed combinational optimization problem, which is able to assist efficient algorithm design. However, the proposed sparse and low-rank optimization framework raises a unique challenge due to the non-convex constraints including the sparsity constraints coupled with a low-rank constraint. Although ℓ_1-norm and nuclear norm relaxation is widely used as a convex surrogate for ℓ_0-norm and the low-rank function, respectively, these convex relaxation approaches can not ensure exact sparsity and low-rank constraints. To address this issue, we further develop novel difference-of-convex (DC) representations for the ℓ_0-norm and the rank function, followed by developing the efficient DC algorithm with convergence guarantees for solving the pliable data shuffling problem. Furthermore, simulation results on real datasets illustrate that the proposed pliable data shuffling strategy achieves comparable statistical performance for on-device distributed learning compared to index coding based data shuffling scheme, while significantly reducing the communication cost.

2. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we present the on-device distributed learning system, followed by the communication-efficient pliable data shuffling scheme to improve the learning performance.

This work was supported in part by the National Nature Science Foundation of China under Grant 61601290 and in part by the Shanghai Sailing Program under Grant 16YF1407700.

2.1. On-Device Distributed Learning System

Consider the on-device distributed learning system, as shown in Fig. 1, which consists of a single-antenna access point (AP) and K single-antenna mobile devices [13]. The AP has access to the entire dataset $\{W_1, \cdots, W_M\}$ of size M data points. Each mobile device is equipped with a cache of size S. We assume that the dataset can not be fully stored at each device, i.e. $S < M$, otherwise there is no need for data shuffling. The K devices collaboratively perform a distributed learning task to learn a model parameterized by $\boldsymbol{z}$. This is achieved by iteratively performing local updates at each device and then updating the global model at the AP. Specifically, let $\mathcal{V}_k \subset \{1, \cdots, M\}$ be the index set of data points cached at mobile device k, and then the available data points set can be represented as $W_{[\mathcal{V}_k]} = \{W_j, j \in \mathcal{V}_k\}$. At epoch t, initialized by the global parameter $\boldsymbol{z}^t$ from AP, the k-th device carries out local computation based on its cached data points $W_{[\mathcal{V}_k]}$ and uploads its local outcome $\boldsymbol{z}_k^t$ to AP. The AP aggregates all the local outcomes to obtain a new global model parameter via the aggregation function $\boldsymbol{z}^{t+1} = g(\boldsymbol{z}_1^t, \cdots, \boldsymbol{z}_K^t)$, and then broadcasts it to each device as the initial value for the $(t+1)$-th epoch.

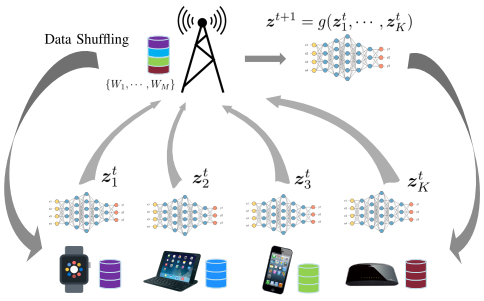


Fig. 1: On-device distributed learning system.

To improve the statistical learning performance, including the faster convergence rate and higher prediction accuracy, the dataset normally needs to be reshuffled across the devices [3, 4]. However, the statistical benefits come at the cost of heavy communication overheads between the AP and mobile devices, since the entire dataset is communicated from AP to each mobile device during the dataset shuffling. To address this communication challenge, index coding has been proposed to improve the communication efficiency of data shuffling [3]. Specifically, the index coding based data shuffling approach is achieved by: (i) randomly selecting data points for each device, i.e., data placement; and (ii) sending the corresponding data points to each device by the index coding approach [8], i.e., data transmission. To further reduce the communication costs, in this work, we shall propose a novel pliable data shuffling approach for joint data placement and data transmission design based on the principles of pliable index coding [7, 10]. This is based on the key observation that, in many distributed learning problems (e.g., classification), the statistical learning performance can be improved as long as the data points at each device are refreshed with a new data point [9], instead of the specific data points.

2.2. Pliable Data Shuffling

Let $h_k \in \mathbb{C}$ denote the channel coefficient between the AP and mobile device k. We encode data points W_i into a vector $\boldsymbol{x} \in \mathbb{C}^r$ of length r. Therefore, the received signal at the k-th mobile device over the r channel uses is given by

$$\boldsymbol{y}_k = \sum_{i=1}^{M} h_k \boldsymbol{x}_i + \boldsymbol{n}_k, \tag{1}$$

where $\boldsymbol{n}_k \sim \mathcal{CN}(\boldsymbol{0}, \sigma_k^2 \boldsymbol{I}_r)$ is the additive isotropic white Gaussian noise at device k. Here, we consider a quasi-static fading channel model in which channel coefficients remain unchanged over r time slots [11].

In this paper, we restrict the encoding and decoding schemes to be linear and leverage the side information at each mobile device to help transceiver design. Without loss of generality, let $s_i \in \mathbb{C}$ be the representative scalar for data point W_i. The transmitted signal of the data point W_i after linear precoding is thus $\boldsymbol{x}_i = \boldsymbol{v}_i s_i$, where $\boldsymbol{v}_i \in \mathbb{C}^r$ is the precoding vector for data point W_i. Likewise, let $\boldsymbol{u}_k \in \mathbb{C}^r$ be the decoding vector at mobile device k. Each device k decodes a data point from the received signal as follows:

$$\tilde{z}_k = \boldsymbol{u}_k^{\mathsf{H}} \boldsymbol{y}_k = h_k \sum_{i=1}^{M} \boldsymbol{u}_k^{\mathsf{H}} \boldsymbol{v}_i s_i + \boldsymbol{u}_k^{\mathsf{H}} \boldsymbol{n}_k. \tag{2}$$

The index coding based data shuffling scheme requires each device decode a specific data point based on the random data placement. This makes devices get a nearly "fresh" data sample, yielding better statistical performance [3]. However, in many distributed learning problems (eg., classification [9]), the learning performance can be improved as long as each device can decode a new data point that is not cached at its local storage unit [9]. This key observation offers new opportunities to further reduce communication costs.

Specifically, to decode a new data point at device k, we impose the following ***pliable interference alignment condition*** for the precoding and decoding vectors [7]:

$$\boldsymbol{u}_k^{\mathsf{H}} \boldsymbol{v}_j \neq 0, \quad \text{for some } j \notin \mathcal{V}_k, \tag{3}$$

$$\boldsymbol{u}_k^{\mathsf{H}} \boldsymbol{v}_i = 0, \quad \forall i \neq j, i \notin \mathcal{V}_k, \tag{4}$$

which implies one of the desired data points W_j is preserved and all other messages from the received signal are eliminated. For notational simplicity, let $\mathcal{I}_k(\{\boldsymbol{u}_k, \boldsymbol{v}_i\}) := \{j \notin \mathcal{V}_k | \boldsymbol{u}_k^{\mathsf{H}} \boldsymbol{v}_j \neq 0\}$, then the above pliable interference alignment condition can be written as $|\mathcal{I}_k| = 1$ for device k.

The data shuffling in each epoch is accomplished by multiple transmissions, and for each transmission it is not necessary to impose that all mobile devices update exact one data point [9]. We thus relax the pliable interference alignment condition for all devices as

$$|\mathcal{I}_k(\{\boldsymbol{u}_k, \boldsymbol{v}_i\})| \leq 1, \quad k = 1, \cdots, K. \tag{5}$$

If $|\mathcal{I}_k| = 1$, then mobile device k is able to decode a new data point successfully and shall replace an old data points by the new one. On the other hand, if $|\mathcal{I}_k| = 0$, then mobile device k will do nothing during this transmission. After multiple rounds of transmission, mobile devices have updated its cache data points for next epoch.

Furthermore, the high similarity in data among mobile devices will cause performance degradation [1, 9]. We thus impose an extra constraint, i.e., ***diversity constraint***, such that each data point can only be distributed to at most w mobile devices ($1 \leq w \leq K$). Specifically, for the j-th data point, let $\mathcal{D}_j(\{\boldsymbol{u}_k, \boldsymbol{v}_i\}) := \{k | \boldsymbol{u}_k^{\mathsf{H}} \boldsymbol{v}_j \neq 0, j \notin \mathcal{V}_k, \ k = 1, \cdots, K\}$ denotes the set of devices that will decode the j-th data point. Therefore, we impose

$$|\mathcal{D}_j(\{\boldsymbol{u}_k, \boldsymbol{v}_i\})| \leq w, \quad j = 1, \cdots, M, \tag{6}$$

to reduce the correlation of shuffled data across the mobile devices.

Given the channel uses r and diversity constraint (6), our goal is to maximize the number of decodable data points at the devices, thereby improving the statistical learning performance of distributed learning. This pliable data shuffling problem thus can be formulated as follows:

$$\begin{aligned} \underset{\{\boldsymbol{u}_k\}, \{\boldsymbol{v}_i\}}{\text{maximize}} \quad & \sum_{k=1}^{K} |\mathcal{I}_k(\{\boldsymbol{u}_k, \boldsymbol{v}_i\})| \\ \text{subject to} \quad & |\mathcal{I}_k(\{\boldsymbol{u}_k, \boldsymbol{v}_i\})| \leq 1, k = 1, \cdots, K, \\ & |\mathcal{D}_j(\{\boldsymbol{u}_k, \boldsymbol{v}_i\})| \leq w, j = 1, \cdots, M, \end{aligned} \tag{7}$$

which is a mixed combinatorial and highly intractable optimization problem. In the next section, we shall propose a sparse and low-rank optimization framework to facilitate efficient algorithms design.

3. SPARSE AND LOW-RANK OPTIMIZATION FOR PLIABLE DATA SHUFFLING

In this section, we propose a sparse and low-rank optimization approach to solve the pliable data shuffling problem (7).

3.1. Sparse and Low-Rank Optimization Framework

Let $X_{ki} = \boldsymbol{u}_k^{\mathsf{H}} \boldsymbol{v}_i, \forall k = 1, \cdots, K,\ i = 1, \cdots, M$. By defining the $K \times M$ matrix $\boldsymbol{X} = [X_{ki}]$, we have the rank of matrix $\boldsymbol{X}$ as $\mathrm{rank}(\boldsymbol{X}) = r$ to represent the channel uses. Therefore, the relaxed pliable interference alignment condition (5) can be rewritten as the following sparsity constraint

$$\|\boldsymbol{x}_{k,\mathcal{V}_k^c}\|_0 \leq 1, , \quad k = 1, \cdots, K, \tag{8}$$

where $\mathcal{V}_k^c$ is the complementary set of $\mathcal{V}_k$, i.e., $\mathcal{V}_k^c := \{1, \cdots, K\} \backslash \mathcal{V}_k$, and $\boldsymbol{x}_{k,\mathcal{V}_k^c}$ is the subvector of the k-th row of matrix $\boldsymbol{X}$, which only keeps the elements in the index set $\mathcal{V}_k^c$. Similarly, the diversity constraint (6) can be formulated as the following sparsity constraint

$$\|\boldsymbol{x}_{\mathcal{C}_j^c,j}\|_0 \leq w, \quad j = 1, \cdots, M, \tag{9}$$

where $\mathcal{C}_j := \{k : j \in \mathcal{V}_k\} \subseteq [K]$ and $\boldsymbol{x}_{\mathcal{C}_j^c,j}$ is the subvector of the j-th column of matrix $\boldsymbol{X}$, which only keeps the elements in the index set $\mathcal{C}_j^c$.

Therefore, to support efficient algorithm design for the proposed pliable data shuffling problem (7), we propose the following sparse and low-rank optimization approach

$$\begin{aligned} \mathscr{P}: \quad \underset{\boldsymbol{X} \in \mathbb{R}^{K \times M}}{\text{maximize}} \quad & \sum_{k=1}^{K} \|\boldsymbol{x}_{k,\mathcal{V}_k^c}\|_0 \\ \text{subject to} \quad & \|\boldsymbol{x}_{k,\mathcal{V}_k^c}\|_0 \leq 1, k = 1, \cdots, K, \\ & \|\boldsymbol{x}_{\mathcal{C}_j^c,j}\|_0 \leq w, j = 1, \cdots, M, \\ & \mathrm{rank}(\boldsymbol{X}) \leq r. \end{aligned} \tag{10}$$

Note that we only need to consider problem $\mathscr{P}$ in real field without loss of any performance in terms of eliminating interference and channel uses, since the encoding and decoding vector can be restricted to the real field [11]. Although the sparse and low-rank optimization problem $\mathscr{P}$ is still nonconvex, we will show that it enjoys algorithmic advantages.

3.2. Problem Analysis

Sparse and low-rank optimization plays a key role in various scenarios in signal processing [14], wireless communication [15], distributed machine learning [16] and high-dimensional data analysis [17]. Despite the non-convexity of ℓ_0-norm and low rank function, a number of computationally efficient numerical algorithms have been developed. In particular, ℓ_1-norm relaxation is widely used as a convex surrogate for ℓ_0-norm, while the low-rankness is often induced by its convex surrogate nuclear norm. However, these approaches can not induce exact sparsity and low-rankness, which is required in our problem $\mathscr{P}$. To address this issue, we shall develop novel DC representations for the ℓ_0-norm and the rank function, thereby guaranteeing the exact sparsity and low-rankness constraints.

4. DC ALGORITHM FOR SPARSE AND LOW-RANK OPTIMIZATION

In this section, we develop a difference-of-convex (DC) algorithm for solving the sparse and low-rank optimization problem $\mathscr{P}$ via the novel sparse and low-rank representations via DC functions .

4.1. DC Representation

To derive the DC representation for the sparse and low-rank functions, we need a couple of definitions.

Definition 1 (Largest-k ℓ_2-norm[18]) *For an integer $k \in \{1, \cdots, n\}$, the largest-k ℓ_2-norm of $\boldsymbol{x} \in \mathbb{R}^n$ denoted by $|||\boldsymbol{x}|||_{k,2}$, is defined as the square root of the sum of the largest k entries in square value, i.e.,*

$$|||\boldsymbol{x}|||_{k,2} = (x_{\pi(1)}^2 + x_{\pi(2)}^2 + \cdots + x_{\pi(k)}^2)^{\frac{1}{2}},$$

where π is an arbitrary permutation such that $x_{\pi(1)}^2 \geq x_{\pi(2)}^2 \geq \cdots \geq x_{\pi(n)}^2$.

Definition 2 (Ky Fan 2-k norm [19]) *For an integer $1 \leq k \leq \min\{m, n\}$, the Ky Fan 2-k norm of matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ is defined as the ℓ_2-norm of the sub-vector formed by the largest-k singular values of $\boldsymbol{X}$. That is,*

$$|||\boldsymbol{X}|||_{k,2} = \left(\sum_{i=1}^{k} \sigma_i^2(\boldsymbol{X}) \right)^{1/2},$$

where $\sigma_i(\boldsymbol{X})$ is the i-th largest singular value of matrix $\boldsymbol{X}$.

We thus express ℓ_0-norm as a DC function [18]:

$$\|\boldsymbol{x}\|_0 \leq k \Longleftrightarrow \|\boldsymbol{x}\|_2^2 - |||\boldsymbol{x}|||_{k,2}^2 = 0. \tag{11}$$

Furthermore, the rank constrain in problem $\mathscr{P}$ can also be expressed as a DC function [16]:

$$\mathrm{rank}(\boldsymbol{X}) \leq k \Longleftrightarrow \|\boldsymbol{X}\|_F^2 - |||\boldsymbol{X}|||_{k,2}^2 = 0. \tag{12}$$

We finally arrive at the DC representation for the sparse and low-rank constraints in problem $\mathscr{P}$, i.e., $\|\boldsymbol{x}_{k,\mathcal{V}_k^c}\|_2^2 - |||\boldsymbol{x}_{k,\mathcal{V}_k^c}|||_{1,2}^2 = 0, k = 1, \cdots, K, \|\boldsymbol{x}_{\mathcal{C}_j^c,j}\|_2^2 - |||\boldsymbol{x}_{\mathcal{C}_j^c,j}|||_{w,2}^2 = 0, j = 1, \cdots, M$, and $\|\boldsymbol{X}\|_F^2 - |||\boldsymbol{X}|||_{r,2}^2 = 0$.

By further relaxing the objective function in (10) to $\sum_{k=1}^{K} \|\boldsymbol{X}_{k,\mathcal{V}_k^c}\|_\infty$, we propose the following DC programming approach to solve problem $\mathscr{P}$:

$$\begin{aligned} \underset{\boldsymbol{X} \in \mathbb{R}^{K \times M}}{\text{minimize}} \quad & \varphi(\boldsymbol{X}) - \sum\nolimits_{k=1}^{K} \|\boldsymbol{x}_{k,\mathcal{V}_k^c}\|_\infty \\ \text{subject to} \quad & \|\boldsymbol{X}\|_F^2 \leq c, \end{aligned} \tag{13}$$

where $\varphi(\boldsymbol{X}) = \gamma(\|\boldsymbol{X}\|_F^2 - |||\boldsymbol{X}|||_{r,2}^2) + \rho \sum_{k=1}^{K} (\|\boldsymbol{x}_{k,\mathcal{V}_k^c}\|_2^2 - |||\boldsymbol{x}_{k,\mathcal{V}_k^c}|||_{1,2}^2) + \lambda \sum_{i=1}^{M} (\|\boldsymbol{x}_{\mathcal{C}_j^c,j}\|_2^2 - |||\boldsymbol{x}_{\mathcal{C}_j^c,j}|||_{w,2}^2)$, and the constraint $\|\boldsymbol{X}\|_F^2 \leq c$ with $c > 0$ is added to avoid unboundedness of the objective value in problem (13), while it doesn't change the rank and sparsity pattern of matrix $\boldsymbol{X}$. If the value of $\varphi(\boldsymbol{X})$ in problem (13) achieves zero, then we are successful in finding a feasible solution to the pliable data shuffling problem $\mathscr{P}$. To simplify the notation, let

$$g(\boldsymbol{X}) := \gamma \|\boldsymbol{X}\|_F^2 + \rho \sum_{k=1}^{K} \|\boldsymbol{X}_{k,\mathcal{V}_k^c}\|_2^2 + \lambda \sum_{j=1}^{M} \|\boldsymbol{X}_{\mathcal{C}_j^c,j}\|_2^2,$$

$$h(\boldsymbol{X}) := \gamma \, |||\, \boldsymbol{X} \,|||_{r,2}^2 + \sum_{k=1}^{K} (\|\boldsymbol{x}_{k,\mathcal{V}_k^c}\|_\infty + \rho \, |||\, \boldsymbol{X}_{k,\mathcal{V}_k^c} |||_{1,2}^2) + \lambda \sum_{j=1}^{M} |||\boldsymbol{X}_{\mathcal{C}_j^c,j}\,|||_{w,2}^2 .$$

The DC program (13) can be rewritten as

$$\begin{aligned} \underset{\boldsymbol{X} \in \mathbb{R}^{K \times M}}{\text{minimize}} \quad & g(\boldsymbol{X}) - h(\boldsymbol{X}) \\ \text{subject to} \quad & \|\boldsymbol{X}\|_F^2 \leq c. \end{aligned} \tag{14}$$

4.2. DC Algorithm

Although problem (14) is still non-convex, we shall develop a simplified form of DC algorithm [20] to solve it efficiently. At each iteration, we solve a convex subproblem which is defined by linearizing the concave term $-h(\boldsymbol{X})$ in problem (14). Specifically, the

subproblem at iteration t is given by

$$\begin{aligned} \underset{\boldsymbol{X}^t \in \mathbb{R}^{K\times M}}{\text{minimize}} \quad & g(\boldsymbol{X}^t) - \langle \boldsymbol{X}^t, \boldsymbol{S}^{t-1} \rangle, \\ \text{subject to} \quad & \|\boldsymbol{X}^t\|_F^2 \le c, \end{aligned} \tag{15}$$

where $\boldsymbol{S}^{t-1}$ is a subgradient of $h(\boldsymbol{X})$ at $\boldsymbol{X}^{t-1}$, i.e.,

$$\begin{aligned} \boldsymbol{S}^{t-1} \in \partial h(\boldsymbol{X}^{t-1}) = \gamma \cdot \partial \,|||\, \boldsymbol{X}^{t-1} \,|||_{r,2}^2 + \lambda \sum_{j=1}^{M} \partial \,|||\, \boldsymbol{x}_{\mathcal{C}_j^c,j}^{t-1} |||_{w,2}^2 \\ + \sum_{k=1}^{K} (\rho \cdot \partial \,|||\, \boldsymbol{x}_{k,\mathcal{V}_k^c}^{t-1} \,|||_{1,2}^2 + \partial \|\boldsymbol{x}_{k,\mathcal{V}_k^c}\|_\infty). \end{aligned} \tag{16}$$

One subgradient of the square of Ky Fan 2-k norm $|||\boldsymbol{X}|||_{r,2}^2$ at point $\boldsymbol{X}$ is given as [16] $\partial \,|||\, \boldsymbol{X}|||_{r,2}^2 = 2\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$, where $\{\sigma_i\}_{i=1}^r$, $\{\boldsymbol{u}_i\}_{i=1}^r$ and $\{\boldsymbol{v}_i\}_{i=1}^r$ are the r largest singular values and the corresponding left and right singular vectors of $\boldsymbol{X}$. For a vector $\boldsymbol{x} \in \mathbb{R}^d$, the subdifferential of the square of the largest-k ℓ_2-norm is given as [18]

$$\partial \,|||\, \boldsymbol{x}|||_{k,2}^2 = \left\{ \boldsymbol{z} : z_i = \begin{cases} 2x_i, & \text{if } \pi(i) \le k \\ 0, & \text{if } \pi(i) \ge k \end{cases} \right\}. \tag{17}$$

A subgradient of $\boldsymbol{s} \in \partial \|\boldsymbol{x}\|_\infty$ at point $\boldsymbol{x}$ can be computed by assigning the sign of the largest magnitude of $\boldsymbol{x}$ to the corresponding element of $\boldsymbol{s}$ and 0 to others [21].

The overall iterates of our proposed DC algorithm are listed in Algorithm 1. Note that we need to solve the convex subproblem

Algorithm 1: DC algorithm for solving problem (14)

Input : Side information $\mathcal{V}_k,\ k = 1, \cdots K$, rank r, $\boldsymbol{X}^0$, c
for $t = 1, 2, \cdots$ **do**
  Compute a subgradient: $\boldsymbol{S}^{t-1} \in \partial h(\boldsymbol{X}^{t-1})$
  Solve the convex subproblem (15), and obtain $\boldsymbol{X}^t$
end

(15) at each iteration, however, this subproblem exists closed form solution which is given by

$$\boldsymbol{X}^t = \begin{cases} \boldsymbol{Z}^t / \|\boldsymbol{Z}^t\|_F, & \text{if } \|\boldsymbol{Z}^t\|_F > c, \\ \boldsymbol{Z}^t, & \text{otherwise}, \end{cases} \tag{18}$$

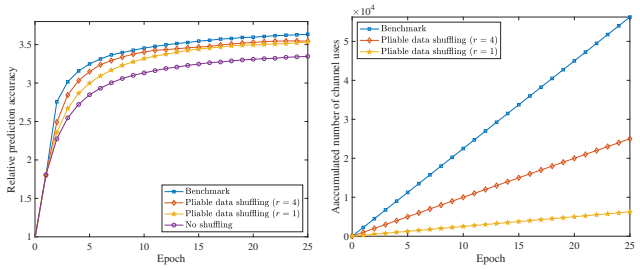
where

$$\boldsymbol{Z}_{i,j}^t = \begin{cases} \frac{1}{2(\gamma+\rho+\lambda)} \boldsymbol{S}_{i,j}^{t-1}, & \text{if } j \in \mathcal{V}_i^c, \\ \frac{1}{2\gamma} \boldsymbol{S}_{i,j}^{t-1}, & \text{otherwise}. \end{cases} \tag{19}$$

The proposed DC algorithm involves computing a subgradient by (16) and solving a convex subproblem (15) at each iteration. The computational complexity of computing the subgradient via (16) is dominated by the truncated SVD of $\boldsymbol{X}$. That is, we only need to calculate the largest r singular values and their corresponding singular vectors of matrix $\boldsymbol{X}$, for which the computational cost is $\mathcal{O}(rKM)$ [22]. Furthermore, the closed form solution (19) to the subproblem (15) is computationally trivial. Therefore, the proposed DC algorithm is very efficient with overall computational time complexity $\mathcal{O}(trKM)$, where t is the required number of iterations to achieve desired accuracy. Furthermore, given rank parameter r and diversity constraint parameter w, the proposed Algorithm 1 for solving Problem (14) converges to critical points from arbitrary initial points [20].

5. SIMULATION RESULTS

To demonstrate the efficiency of the proposed pliable data shuffling scheme, we conduct a classification experiment for the on-device distributed learning system over CIFAR10 dataset [23], which contains $60,000$ samples of size of 32×32 color images in 10 different classes. For the sake of simplicity, we study the multiclass support vector machine (SVM) classifier. We train an SVM classifier via the distributed stochastic gradient descent method [24] using 10000 samples as the training set, and then apply the classifier to the independent testing set with size of 10000. The number of mobile devices is 10 and the cache size of each device is set to 10. Without loss of generality, we take 100 training samples as one data point. That is, each mobile device can store 1000 training samples in total.



(a) Prediction accuracy (b) Number of channel uses

Fig. 2: Experiments of SVM classifier on CIFAR10 dataset.

For our proposed pliable data shuffling scheme, we set the diversity constraint parameter $w = 2$, constant parameter $c = 1$ and the regularizer parameter $\gamma = \rho = \lambda = 10^6$. We solve the data shuffling problem (14) under channel uses $r = 4$ and $r = 1$, respectively. We compare our proposed data shuffling scheme with no shuffling scheme and random shuffling scheme, which is chosen as our benchmark. For the benchmark, we divide the training set into 10 groups with 10 data points in each, and randomly select one group for each mobile device without replacement. Once the data points is selected for mobile devices, the transmission procedure can be modeled as solving a index coding problems [3, 25] to minimize channel uses, and we implement index coding using alternating projection method in [25].

At each epoch, we carry out 10 times data shuffling across the mobile devices. When each device receives a new data point, it will randomly delete a data point from its storage. We illustrate the relative prediction accuracy (normalized by the random classification accuracy) in Fig. 2a. The accumulated number of channel uses for the benchmark and the proposed pliable data shuffling are shown in Fig.2b. Each point is averaged for 100 times. Fig.2a show that data shuffling significantly improves the relative testing accuracy compared with the no shuffling scheme. Furthermore, for channel uses $r = 4$ and $r = 1$ in problem (14), the average accumulated channel uses of our proposed pliable data shuffling scheme are 44% and 11% of the benchmark as demonstrated in Fig. 2b, respectively, but only with a slightly loss of prediction accuracy.

6. CONCLUSION

In this paper we propose a pliable data shuffling approach to jointly design data placement and communication scheme for distributed learning among mobile devices. This is achieved by establishing the novel interference alignment conditions for communication-efficient data delivery, and the diversity constraints to avoid similarity during the data shuffling procedure. To solve the resulting mixed combinational optimization problem for pliable data shuffling, we propose a novel sparse and low-rank framework, for which an efficient DC algorithm was further developed via the DC representation of the sparse and low-rank functions. Numerical results demonstrate that the proposed approach significantly reduces the communication cost for on-device distributed training.

7. REFERENCES

[1] Dong Yin, Ashwin Pananjady, Max Lam, Dimitris Papailiopoulos, Kannan Ramchandran, and Peter Bartlett, "Gradient diversity: a key ingredient for scalable distributed learning," in *Proc. Int. Conf. on Mach. Learn. (ICML)*, 2018, pp. 1998–2007.

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: a system for large-scale machine learning.," in *Proc. OSDI*, 2016, vol. 16, pp. 265–283.

[3] Kangwook Lee, Maximilian Lam, Ramtin Pedarsani, Dimitris Papailiopoulos, and Kannan Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, Aug. 2018.

[4] Benjamin Recht and Christopher Ré, "Parallel stochastic gradient algorithms for large-scale matrix completion," *Math. Program. Comput.*, vol. 5, no. 2, pp. 201–226, 2013.

[5] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE Conf. Computer Commun. (INFOCOM)*, Apr. 2018.

[6] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artificial Intell. Stat. (AISTATS)*, 2017, vol. 54, pp. 1273–1282.

[7] Tao Jiang and Yuanming Shi, "Sparse and low-rank optimization for pliable index coding," in *Proc. IEEE Global Conf. Signal and Inf. Process. (GlobalSIP)*, Nov. 2018.

[8] Fatemeh Arbabjolfaei and Young-Han Kim, "Fundamentals of index coding," *Found. Trends in Commun. and Inf. Theory*, vol. 14, no. 3-4, pp. 163–346, 2018.

[9] Linqi Song, Christina Fragouli, and Tianchu Zhao, "A pliable index coding approach to data shuffling," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2558–2562.

[10] Siddhartha Brahma and Christina Fragouli, "Pliable index coding," *IEEE Trans. Inf. Theory*, vol. 61, no. 11, pp. 6192–6203, 2015.

[11] Yuanming Shi, Jun Zhang, and Khaled B. Letaief, "Low-rank matrix completion for topological interference management by Riemannian pursuit," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4703–4717, Jul. 2016.

[12] Yuanming Shi, Bamdev Mishra, and Wei Chen, "Topological interference management with user admission control via Riemannian optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7362–7375, Nov. 2017.

[13] Jonathan I Tamir, Ethan R Elenberg, Anurag Banerjee, and Sriram Vishwanath, "Wireless index coding through rank minimization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014.

[14] Tao Jiang, Yuanming Shi, Zhang Jun, and K. B. Letaief, "Joint activity detection and channel estimation for IoT networks: phase transition and computation-estimation tradeoff," *IEEE Internet of Things J.*, accepted, preprint 2018.

[15] Yuanming Shi, Jun Zhang, Wei Chen, and Khaled B Letaief, "Generalized sparse and low-rank optimization for ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 42–48, Jun. 2018.

[16] Kai Yang, Yuanming Shi, and Zhi Ding, "Data shuffling in wireless distributed computing via low-rank optimization," *arXiv preprint arXiv:1809.07463*, 2018.

[17] Samet Oymak, Amin Jalali, Maryam Fazel, Yonina C Eldar, and Babak Hassibi, "Simultaneously structured models with application to sparse and low-rank matrices," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2886–2908, 2015.

[18] Katsuya Tono, Akiko Takeda, and Jun-ya Gotoh, "Efficient DC algorithm for constrained sparse optimization," *arXiv preprint arXiv:1701.08498*, 2017.

[19] Xuan Vinh Doan and Stephen Vavasis, "Finding the largest low-rank clusters with Ky Fan 2-k-norm and ℓ_1-norm," *SIAM J. on Optim.*, vol. 26, no. 1, pp. 274–312, 2016.

[20] Pham Dinh Tao and Le Thi Hoai An, "Convex analysis approach to DC programming: Theory, algorithms and applications," *Acta Math. Vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.

[21] Jun-ya Gotoh, Akiko Takeda, and Katsuya Tono, "DC formulations and algorithms for sparse optimization problems," *Math. Program.*, vol. 169, no. 1, pp. 141–176, 2018.

[22] Lloyd N Trefethen and David Bau III, *Numerical linear algebra*, vol. 50, SIAM, 1997.

[23] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., University of Toronto, 2009.

[24] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola, "Parallelized stochastic gradient descent," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2010, pp. 2595–2603.

[25] Xiao Huang and Salim El Rouayheb, "Index coding and network coding via rank minimization," in *Proc. IEEE Int. Symp. Inf. theory Workshop (ITW)*, Oct. 2015, pp. 14–18.