Distributed Optimization with the Communication Efficient Regularizer

YAWEI ZHAO, Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, China. and School of Computer, National University of Defense Technology, China.
QIAN ZHAO, College of Mathematics and System Science, Xinjiang University, China.
SHIXIONG WANG, Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, China.
EN ZHU and XINWANG LIU, School of Computer, National University of Defense Technology, China.
LAILONG LUO, Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, China.
JIANPING YIN, School of Computer, Dongguan University of Technology, China.

We propose a new distributed optimization method to jointly optimize a machine learning model and the communication efficiency in a parameter server system. Specifically, we propose a communication efficient regularizer, which encourages the update of the parameter to own clustering structures. Thus, the update of the parameter can be encoded by using few bits, reducing the workload of data transmission. Additionally, we provide the sublinear convergence rate for the proposed method theoretically. Finally, the superiority of the proposed method is verified by extensive empirical studies.

Additional Key Words and Phrases: Distributed optimization, communication efficiency, gradient quantization, convergence rate.

Authors' addresses: Yawei Zhao, zhaoyawei@nudt.edu.cn, Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Beijing, 100071, China. School of Computer, National University of Defense Technology, 109 Deya Road, Changsha, Hunan, 410073, China. Qian Zhao, mcsqian.zhao@gmail.com, College of Mathematics and System Science, Xinjiang University, Urumqi, Xinjiang, 830001, China. Shixiong Wang, wsx09@foxmail.com, Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Beijing, 100071, China. En Zhu, enzhu@nudt.edy.cn; Xinwang Liu, xinwangliu@nudt.edu.cn, School of Computer, National University of Defense Technology, 109 Deya Road, Changsha, Hunan, 410073, China. Lailong Luo, luolailong09@nudt.edu.cn, Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, 109 Deya Road, Changsha, Hunan, 410073, China. Jianping Yin, jpyin@dgut.edu.cn, School of Computer, Dongguan University of Technology, Dongguan, Guangdong, 523808, China.

XXXX-XXXX/2020/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Yawei Zhao, Qian Zhao, Shixiong Wang, En Zhu, Xinwang Liu, Lailong Luo, and Jianping Yin. 2020. Distributed Optimization with the Communication Efficient Regularizer. 1, 1 (April 2020), 20 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Many machine learning tasks, e.g., ridge regression and logistic regression, are usually formulated as an optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad f(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^d} \quad \mathbb{E}_a F(\mathbf{x}; a), \tag{1}$$

where a represents a random variable, and $\mathbb{E}$ represents to take its mathematical expectation. For example, the data matrix $\mathbf{A} \in \mathbb{R}^{M\times d}$ consists of M instances, and $\mathbf{A}_i \in \mathbb{R}^{1\times d}$ represents the i-th instance. Generally, a is picked from $\{1, 2, ..., M\}$ with an uniformly random distribution. Let us take some examples for more explanation.

- Suppose we use the ridge regression model [13] to conduct a prediction task. The label is $\mathbf{y} \in \mathbb{R}^M$. Without loss of generality, when a is randomly sampled from $\{1, 2, ..., M\}$ with the equal probability, e.g., $a = i$, $F(\mathbf{x}; a = i) = (\mathbf{A}_i\mathbf{x} - y_i)^2 + \gamma \|\mathbf{x}\|_2^2$, and $f(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{A}_i\mathbf{x} - y_i)^2 + \gamma \|\mathbf{x}\|_2^2$.
- Suppose we use the logistic regression model [13] to conduct a classification task. The label is $\mathbf{y} \in \mathbb{R}^M$. The elements of $\mathbf{y}$ consist of either 1 or -1. Without loss of generality, when a is randomly sampled from $\{1, 2, ..., M\}$ with the equal probability, e.g., $a = i$, $F(\mathbf{x}; a = i) = \log(1 + \exp(-y_i\mathbf{A}_i\mathbf{x}))$, and $f(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M}\log(1 + \exp(-y_i\mathbf{A}_i\mathbf{x}))$.

With the proliferation of data, (1) is usually conducted in a *parameter server* system [9, 22]. As illustrated in Figure 1, a classic parameter server system is abstracted to be a server and multiple workers[1]. The whole dataset is partitioned into W parts, and every part of the dataset is thus stored in a worker. The previous distributed optimization method to solve Eq. (1) includes the following steps.

(1) The server sends $\mathbf{x}_t$ to every worker at the t-th iteration.
(2) For the w-th worker, it randomly samples an instance from the local data to compute the unbiased stochastic gradient $\mathbf{g}_t^{(w)}$ such that $\mathbb{E}_a\mathbf{g}_t^{(w)} = \mathbb{E}_a F(\mathbf{x}_t; a) = \nabla f(\mathbf{x}_t)$.
(3) The classic update of the parameter in the w-th worker at the t-th iteration is

$$\mathbf{x}_{t+1}^{(w)} = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^d} \quad \langle \mathbf{g}_t^{(w)}, \mathbf{y} - \mathbf{x}_t\rangle + \frac{1}{2\eta_t}\|\mathbf{y} - \mathbf{x}_t\|_2^2 . \tag{2}$$

(4) All local parameters $\mathbf{x}_{t+1}^{(w)}$ with $1 \le w \le W$ are sent to the server, and they are taken average to obtain $\mathbf{x}_{t+1}$ for the next iteration.

To solve Eq. (2), the above iteration has to be performed for hundreds or thousands of rounds. Since every worker has to send its local parameter to the server for every iteration, the previous distributed optimization method leads to heavy workload of network communication, especially when there are a large number of workers [7]. Previous studies have proposed many quantization methods to compress the stochastic gradients and improve the communication efficiency by sending those compressed

[1] A parameter server system may consist of multiple servers.

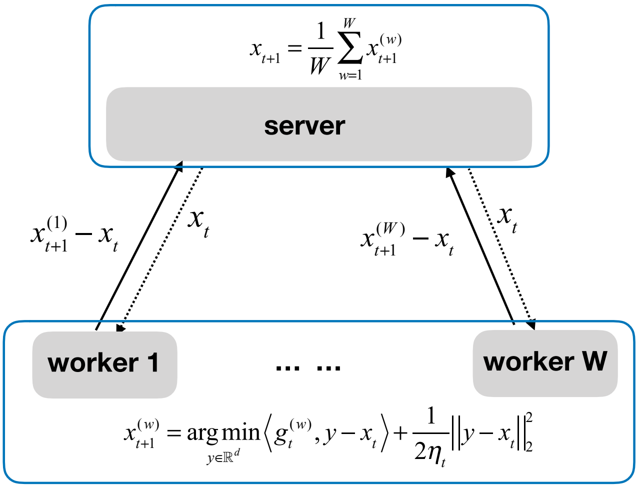

$x_{t+1} = \frac{1}{W}\sum_{w=1}^{W} x_{t+1}^{(w)}$
server
$x_{t+1}^{(1)} - x_t$
x_t
$x_{t+1}^{(W)} - x_t$
x_t
worker 1
… …
worker W
$x_{t+1}^{(w)} = \arg\min_{y\in\mathbb{R}^d} \langle g_t^{(w)}, y - x_t \rangle + \frac{1}{2\eta_t}\|y - x_t\|_2^2$

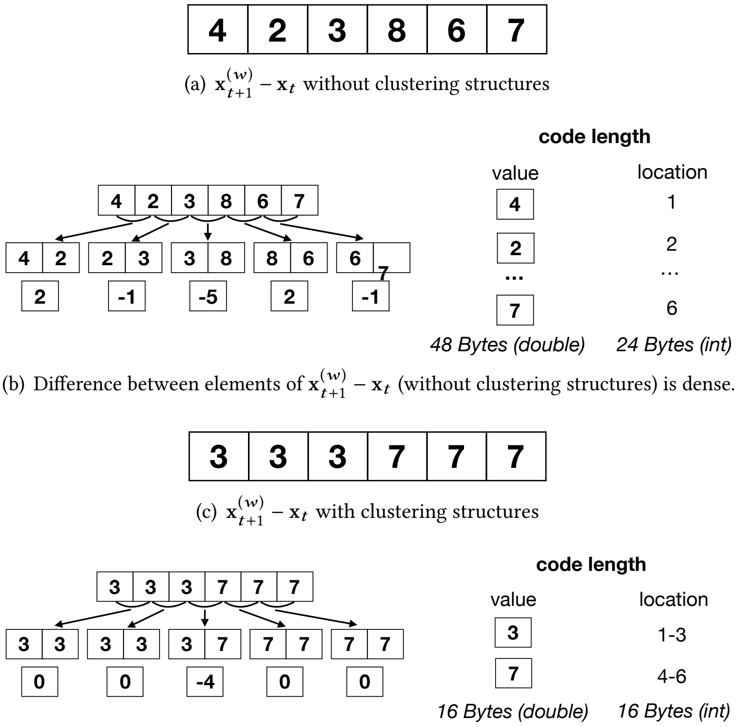
Fig. 1. The illustration of the $t + 1$-th iteration of the gradient descent method, which is performed in a parameter server system.

gradients [2, 10, 15, 18, 24]. But, those gradient quantization methods have to lose some accuracy of gradient, i.e., $\mathbf{g}_t^{(w)}$, which impairs the convergence performance. Since those methods are independent to Eq. (1), they cannot make a good tradeoff between the convergence performance and the communication efficiency.

In the paper, we jointly optimize Eq. (1) in the distributed setting by using a communication efficient regularizer. We propose a distributed optimization method, which encourages the update of parameter $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ to own clustering structures. Figure 2 presents an illustrative example. According to Figures 2(a) and 2(c), we observe when the elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ own clustering structures, they can be encoded by using fewer bits. Its code length can be reduced a lot. The update of parameter can be transmitted from workers and the server efficiently. According to Figures 2(b) and 2(d), our basic idea is to let the difference between the elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ be sparse, which encourages the elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ to have clustering structures.

Comparing with the gradient quantization methods in the previous studies, the proposed method is able to find a good tradeoff between the convergence performance and the communication efficiency. It is highlighted that the proposed method does not impair the convergence performance theoretically. Additionally, we propose an Alternating Direction of Method of Multipliers (ADMM) method to solve Eq. (1) efficiently. Furthermore, we improve the proposed method with some useful coding strategies. Finally, extensive empirical studies verify the advantages of the proposed method. In summary, our main contributions are outlined as follows.

- We propose a communication efficient regularizer to jointly optimize a machine learning model and the communication efficiency.
- We propose a new ADMM method to update the parameter efficiently.
- We prove that the proposed distributed optimization method converges with a sublinear rate theoretically.



(a) $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ without clustering structures

(b) Difference between elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ (without clustering structures) is dense.

(c) $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ with clustering structures

(d) Difference between elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ (with clustering structures) is **sparse**.

Fig. 2. The illustrative example shows that $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ with clustering structures can be compressed by using fewer bits, and thus the code length is reduced effectively. **Our basic idea is to make the difference between elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ sparse.**

The paper is organized as follows. Section 2 outlines the related work. Section 3 presents the proposed distributed optimization method by using a communication efficient regularizer. Section 4 presents the efficient method to conduct the update of parameters. Section 5 presents the convergence rate of the proposed method. Section 6 presents some useful coding strategies to improve the communication efficiency. Section 7 presents extensive empirical studies. Section 8 concludes the paper.

2 RELATED WORK

We review the previous related literatures briefly.

2.1 Methods to reduce the number of rounds of communication

[8, 11, 14, 15, 17, 25, 26] develop flexible methods to reduce the number of rounds of communication for the distributed optimization. For example, [25] proposes a new sub-sampling method to reduce the number of rounds of communication. [17] proposes a method to reduce the required memory for communication and the number of rounds of communication. Additionally, [12] designs a communication-efficient protocol to reduce the workload of communication when handling the sparse dataset.

[23] develops a new coding method for gradients to make a good trade-off between the computation and communication, but there is no theoretical guarantee for the convergence rate. Comparing with the proposed method, those previous researches do not involve in the compression of the update of parameter, and are orthogonal to our work. Therefore, these researches can be used to improve our method.

2.2 Methods to reduce the code length of gradients

[21] proposes a sufficient factor broadcasting method to reduce the workload of network communication in a parameter server system. Since it is designed for matrix-parametrized models, it is not suitable for a general machine learning model such as ridge regression. [1] improves the communication efficiency by making the gradient sparse with no theoretical guarantee. [2, 24] propose gradient quantization methods to reduce the code length of gradients. [18] makes the gradient sparse to reduce its code length. Comparing with the proposed method, those previous researches separately consider the learning of a machine learning model and the improvement of the communication efficiency. The proposed method outperforms them by making a good tradeoff between the convergence and the communication efficiency, which does not impair the convergence rate theoretically.

3 DISTRIBUTED OPTIMIZATION WITH THE COMMUNICATION EFFICIENT REGULARIZER

In the section, we first show the major notations, and propose the communication efficient regularizer. Then, we present our distributed optimization method.

3.1 Notations

- The regular letters W and w represent scalars. W represents the number of workers in a parameter server system, and $w \in [W]$ means the w-th worker.
- The bold lower-case letter $\mathbf{x}$ represents the parameter of a machine learning problem, and its i-th element is represented by $\mathbf{x}^i$.
- The bold upper-case letter $\mathbf{A}$ represents the data matrix, and the i-th row of $\mathbf{A}$ represents the i-th instance.
- The bold upper-case letter $\mathbf{Q}$ represents a matrix, and its i-th row is represented by $\mathbf{Q}^i$.
- $\mathbf{g}_t^{(w)}$ represents the stochastic gradient at the t-th iteration for the w-th worker. M and d represent the number of instances in the dataset and the dimension of every instance, respectively.
- σ represents the smallest singular value of $\mathbf{Q}$.
- $\|\cdot\|_s$ represents the ℓ_s-norm with $s \in \{1, 2, \infty\}$.
- $B_p(\mathbf{z}, \mathbf{x})$ represents the p-norm Bregman divergence, which is defined by $B_p(\mathbf{z}, \mathbf{x}) := \phi(\mathbf{z}) - \phi(\mathbf{x}) - \langle \nabla\phi(\mathbf{x}), \mathbf{z} - \mathbf{x} \rangle$ with $\phi(\mathbf{x}) := \frac{1}{2} \|\mathbf{x}\|_p^2$. Here, $1 < p \leq 2$.
- For any $\mathbf{a}$ and $\mathbf{b}$, $\langle \mathbf{a}, \mathbf{b} \rangle$ means $\mathbf{a}^\top \mathbf{b}$, that is the inner product of $\mathbf{a}$ and $\mathbf{b}$.
- $\mathbb{E}$ represents to take mathematical expectation, and ∇ represents the gradient operator.
- $\mathrm{sign}(a) = 1$ if $a > 0$, $\mathrm{sign}(a) = -1$ if $a < 0$, and $\mathrm{sign}(a) = 0$ if $a = 0$.

3.2 CER: communication Efficient Regularizer

Recall that stochastic gradient descent is usually used to solve a general optimization problem Eq. (1). The Distributed Stochastic Gradient Descent (DSGD) method is performed iteratively. The previous DSGD conducts the update of parameter by solving Eq. (2). Since every worker has to send the update of parameter $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ to the server, the workload of the network communication becomes very heavy for a large number of workers. When the workload of the the network communication is heavy, the previous DSGD has to consume much time to complete the data transmission, which impairs the convergence performance of DSGD.

To improve the communication efficiency of DSGD, we propose a new method to conduct the update of the parameter, instead of solving Eq. (2) directly. The proposed method use a communication efficient regularizer to find clustering structures of the elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$. It is formulated as

$$\mathbf{x}_{t+1}^{(w)} = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^d} \quad \left\langle \mathbf{g}_t^{(w)}, \mathbf{y} - \mathbf{x}_t \right\rangle + \frac{1}{2\eta_t} \left\|\mathbf{Q}(\mathbf{y} - \mathbf{x}_t)\right\|_1^2 . \tag{3}$$

Here, $\mathbf{g}_t^{(w)}$ is a stochastic gradient, which is obtained by using the local data in the w-th worker. The given full rank square matrix $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is defined by

$$\mathbf{Q} := \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \cdots & & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix}.$$

Notice that $\mathbf{Q}$ is a full rank square matrix, whose smallest singular value, denoted by σ, is positive, that is, $\sigma > 0$. Comparing with the previous updating rule, i.e., Eq. (2), the proposed communication efficient regularizer is an ℓ_1 norm square. It punishes the difference between elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$, and encourages them to be small or even zero. Thus, those corresponding elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ are very similar or even identical. That is, the elements of $\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t$ own clustering structures. Exploiting the clustering structures, $\mathbf{x}_{t+1}$ can be compressed by using few bits, and thus improves the communication efficiency in the distributed setting.

We give a demo example to explain the communication efficient regularizer intuitively. Suppose the optimum of $\mathbf{x} \in \mathbb{R}^4$ is $\mathbf{x}_* = (1, -1, 1, -1)^\top$. If $\mathbf{x}_0 = (0, 0, 0, 0)^\top$, the communication efficient regularizer may update $\mathbf{x}$ as follows.

$$\begin{aligned} \mathbf{x}_1 &= (-1, -1, 1, 1)^\top, \text{ and } \mathbf{Q}(\mathbf{x}_1 - \mathbf{x}_0) = (0, -2, 0, 1)^\top \\ \mathbf{x}_2 &= (1, -1, 1, 1)^\top, \text{ and } \mathbf{Q}(\mathbf{x}_2 - \mathbf{x}_1) = (2, 0, 0, 0)^\top \\ \mathbf{x}_3 &= (1, -1, 1, -1)^\top, \text{ and } \mathbf{Q}(\mathbf{x}_3 - \mathbf{x}_2) = (0, 0, 2, -2)^\top. \end{aligned}$$

For $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, there are only two different elements: '1' and '−1', which shows that there are two clusters among those elements, and their cluster centers are '1' and '−1', respectively. Meanwhile, the difference of elements between successive update, e.g., $\mathbf{Q}(\mathbf{x}_2 - \mathbf{x}_1)$, containts as least two 0s. **It explains our basic idea, that is, to let the difference between elements of successive update, i.e., $\mathbf{Q}(\mathbf{x}_{t+1}^{(w)} - \mathbf{x}_t)$, be sparse.**

Algorithm 1 Distributed optimization with the communication efficient regularizer

```
Require: The number of total iterations $T$, and the initial parameter $\mathbf{x}_1$.
                         On the server:
for $t = 1, 2, ..., T$ do
    Deliver the parameter $\mathbf{x}_t$ to every worker.
    Wait to collect all updates of parameter $\mathbf{y}_{t+1}^{(w)} - \mathbf{x}_t$ with $1 \leq w \leq W$.
    Update the global parameter $\mathbf{x}_{t+1} = \frac{1}{W} \sum_{n=1}^{W} \mathbf{y}_{t+1}^{(w)}$.
return $\mathbf{x}_{T+1}$.
              On the $w$-th worker for the $t+1$-th iteration:
Receive the parameter $\mathbf{x}_t$ from the server.
$\eta_t = \frac{\sigma^2}{2L_1\sqrt{t}} d^{-1}$, and $\mathbf{x}_{t+1}^{(w)} = \frac{1}{2}(\mathbf{z}_t^{(w)} + \mathbf{x}_t)$.
Randomly sample an instance $\mathbf{a}$, and compute the stochastic gradient $\mathbf{g}_t^{(w)} = \nabla f(\mathbf{x}_{t+1}^{(w)}; \mathbf{a})$.
$\mathbf{y}_{t+1}^{(w)} = \mathrm{argmin}_{\mathbf{y} \in \mathbb{R}^d} \left\langle \mathbf{g}_t^{(w)}, \mathbf{y} - \mathbf{x}_t \right\rangle + \frac{1}{2\eta_t} \|\mathbf{Q}(\mathbf{y} - \mathbf{x}_t)\|_1^2$.
$\mathbf{z}_{t+1}^{(w)} = \mathrm{argmin}_{\mathbf{z} \in \mathbb{R}^d} \left\langle \mathbf{g}_t^{(w)}, \mathbf{z} - \mathbf{x}_{t+1}^{(w)} \right\rangle + \frac{1}{2\alpha_t} B_p(\mathbf{z}, \mathbf{x}_{t+1}^{(w)})$, and $B_p(\mathbf{z}, \mathbf{x}_{t+1}^{(w)}) = \frac{1}{2} \left\| \mathbf{z} - \mathbf{x}_{t+1}^{(w)} \right\|_p^2$.
Send $\mathbf{y}_{t+1}^{(w)} - \mathbf{x}_t$ to the server.
```

However, comparing with Eq. (2), it is challenging for Eq. (3) to obtain sublinear convergence rate in theoretical. The reason is that the communication efficient regularizer is non-smooth. Intuitively, it makes a tradeoff between the convergence performance and the communication efficiency. To improve the communication efficiency, the convergence performance may be impaired. This challenging problem is solved in the following section.

3.3 Communication efficient update with a linear coupling strategy

Consider Eq. (3) on the w-th worker in the distributed setting. We do not transmit the sequence of $\{\mathbf{x}_{t+1}^{(w)}\}_{t=1}^{T}$ directly. Instead, we maintain two sequences $\{\mathbf{y}_t^{(w)}\}_{t=1}^{T}$ and $\{\mathbf{z}_t^{(w)}\}_{t=1}^{T}$ on the w-th worker, and a sequence $\{\mathbf{x}_t\}_{t=1}^{T}$ on the server. Every $\mathbf{x}_{t+1}$ is defined by

$$\mathbf{x}_{t+1} = \frac{1}{W} \sum_{w=1}^{W} \mathbf{y}_{t+1}^{(w)}.$$

We then use a linear coupling strategy $\mathbf{x}_{t+1}^{(w)} = \frac{1}{2}\left(\mathbf{z}_t^{(w)} + \mathbf{y}_t\right)$ to guarantee the sublinear convergence rate.

$\mathbf{y}_{t+1}^{(w)}$ is yielded based on $\mathbf{x}_{t+1}^{(w)}$ by using Eq. (3), that is,

$$\mathbf{y}_{t+1}^{(w)} = \underset{\mathbf{y} \in \mathbb{R}^d}{\mathrm{argmin}} \quad \left\langle \mathbf{g}_t^{(w)}, \mathbf{y} - \mathbf{x}_{t+1}^{(w)} \right\rangle + \frac{1}{2\eta_t} \left\| \mathbf{Q}(\mathbf{y} - \mathbf{x}_{t+1}^{(w)}) \right\|_1^2. \tag{4}$$

Here, η_t is a learning rate, which is usually set to be positively proportional to $\frac{1}{\sqrt{t}}$. $\mathbf{y}_{t+1}^{(w)} - \mathbf{x}_t$ is transmitted to the server, and shared with other workers. Benefiting from the intrinsic clustering structures in elements of $\mathbf{y}_{t+1}^{(w)} - \mathbf{x}_t$, it can be transmitted in a communication efficient way. Additionally, we maintain a local sequence of $\{\mathbf{z}_t^{(w)}\}_{t=1}^T$, which is obtained by

$$\mathbf{z}_{t+1}^{(w)} = \operatorname*{argmin}_{\mathbf{z}\in\mathbb{R}^d} \left\langle \mathbf{g}_t^{(w)}, \mathbf{z} - \mathbf{x}_{t+1}^{(w)} \right\rangle + \frac{1}{2\alpha_t} B_p(\mathbf{z}, \mathbf{x}_{t+1}^{(w)}). \tag{5}$$

Here, $B_p(\mathbf{z}, \mathbf{x}_{t+1}^{(w)}) := \phi(\mathbf{z}) - \phi(\mathbf{x}_{t+1}^{(w)}) - \left\langle \nabla\phi(\mathbf{x}_{t+1}^{(w)}), \mathbf{z} - \mathbf{x}_{t+1}^{(w)} \right\rangle$. $\phi(\mathbf{x}) := \frac{1}{2}\|\mathbf{x}\|_p^2$ and $1 < p \le 2$ is the p-norm Bregman divergence [6]. Note that $\mathbf{z}_{t+1}^{(w)}$ is stored in the local worker, and is not shared with other workers.

The introduction of $\mathbf{z}_{t+1}^{(w)}$ and $\mathbf{z}_{t+1}^{(w)}$

- The sequence $\{\mathbf{y}_t^{(w)}\}_{t=1}^T$ is shared with other workers. Since the elements of every $\mathbf{y}_t^{(w)}$ own clustering structures, it can be transmitted efficiently.
- The sequence $\{\mathbf{z}_t^{(w)}\}_{t=1}^T$ is not shared with other workers. It is used to help the sequence $\{\mathbf{y}_t\}_{t=1}^T$ converge at the sublinear rate theoretically.

The details of our method are presented in Algorithm 1.

However, it is challenging to solve Eq. (4) (or Eq. (3)) efficiently. The reason is that the communication efficient regularizer in Eq. (4) (or Eq. (3)) is non-smooth and non-separable. In the next section, we propose an efficient ADMM method to solve Eq. (4) (or Eq. (3)), and present the closed form to solve Eq. (5).

4 EFFICIENT METHODS TO UPDATE $\mathbf{y}_{t+1}^{(w)}$ AND $\mathbf{z}_{t+1}^{(w)}$

In the section, we propose an efficient ADMM method to update $\mathbf{y}_{t+1}^{(w)}$, and then present the closed form to update $\mathbf{z}_{t+1}^{(w)}$.

As we have shown, the communication efficient regularizer makes the update rule, i.e., Eq. (4) non-separable and non-smooth. It is challenging to solve it directly. By using the ADMM method, we decompose Eq. (4) into some sub-problems, and each of them is able to be solved efficiently.

4.1 Update of $\mathbf{y}_{t+1}^{(w)}$.

Recall that the update of $\mathbf{y}_{t+1}^{(w)}$ is equivalent to be formulated as the following problem.

$$\min_{\mathbf{y}\in\mathbb{R}^d, \mathbf{u}\in\mathbb{R}^d} \left\langle \mathbf{g}_t^{(w)}, \mathbf{y} - \mathbf{x}_{t+1}^{(w)} \right\rangle + \frac{1}{2\eta_t} \|\mathbf{u}\|_1^2$$

subject to

$$\mathbf{u} = Q(\mathbf{y} - \mathbf{x}_{t+1}^{(w)}).$$

The augmented Lagrangian multiplier is

$$L(\mathbf{y}, \mathbf{u}, \boldsymbol{\lambda}) = \left\langle \mathbf{g}_t^{(w)}, \mathbf{y} - \mathbf{x}_{t+1}^{(w)} \right\rangle + \frac{1}{2\eta_t} \|\mathbf{u}\|_1^2 + \boldsymbol{\lambda}^\top \left(Q(\mathbf{y} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u} \right) + \frac{\rho}{2} \left\| Q(\mathbf{y} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u} \right\|_2^2,$$

Algorithm 2 The efficient projected gradient descent method to solve Eq. (8) [5]

```
Require: The positive integer $R$, and positive $\gamma_r$.
for $r = 0, 1, ..., R-1$ do
    $\nabla h(\boldsymbol{\delta}_r) = (h_r(\boldsymbol{\delta}_r^1), ..., h_r(\boldsymbol{\delta}_r^d))^\top$.
    Sort elements of $\nabla h(\boldsymbol{\delta}_r)$ in the decreasing order, and denote it by $\boldsymbol{\nu} := (\boldsymbol{\nu}^1, ..., \boldsymbol{\nu}^d)$ with $\boldsymbol{\nu}^1 > \boldsymbol{\nu}^2 > ... > \boldsymbol{\nu}^d$.
    Find $J = \max\left\{ j | \boldsymbol{\nu}^j - \frac{1}{j}\left(\sum_{i=1}^{j} \boldsymbol{\nu}^i - 1\right) > 0 \right\}$.
    $\tilde{\nabla} h(\boldsymbol{\delta}) := \max\left\{ \boldsymbol{\nu}^i - \boldsymbol{\beta} \right\}$ where $\boldsymbol{\beta} := \frac{1}{J}\left(\sum_{j=1}^{J} \boldsymbol{\nu}^j - 1\right)$.
    $\boldsymbol{\delta}_{r+1} = \boldsymbol{\delta}_r - \gamma_r \tilde{\nabla} h(\boldsymbol{\delta})$.
return $\boldsymbol{\delta}_R$.
```

Algorithm 3 ADMM method to update $\mathbf{y}_{t+1}^{(w)}$

```
Require: The matrix $\mathbf{Q}$, the integer $R$, the step size $\rho$.
for $k = 0, 1, ..., K-1$ do
    $\hat{\mathbf{y}}_{k+1} = \mathbf{x}_{t+1}^{(w)} - \frac{1}{\rho}(\mathbf{Q}^\top \mathbf{Q})^{-1} \mathbf{g}_t^{(w)} + \mathbf{Q}^{-1}(\mathbf{u}_k - \frac{1}{\rho}\boldsymbol{\lambda}_k)$.
    Obtain $\boldsymbol{\delta}_R = (\boldsymbol{\delta}_R^1, ..., \boldsymbol{\delta}_R^d)$ for any $1 \le i \le d$ according to Algorithm 2.
    $\mathbf{u}_{k+1}^i = \frac{\boldsymbol{\delta}_R^i}{2+\rho\boldsymbol{\delta}_R^i}\left(\boldsymbol{\lambda}_k^i + \rho \mathbf{Q}^i(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)})\right)$ for any $1 \le i \le d$.
    $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mathbf{Q}(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u}_{k+1}$.
$\mathbf{y}_{t+1}^{(w)} = \hat{\mathbf{y}}_K$.
```

where ρ is the step size. At the k-th iteration, the update of $\mathbf{y}$ is $\hat{\mathbf{y}}_{k+1} = \operatorname{argmin}_{\mathbf{y} \in \mathbb{R}^d} L(\mathbf{y}, \mathbf{u}_k, \boldsymbol{\lambda}_k)$, which is equivalent to

$$\begin{aligned}\hat{\mathbf{y}}_{k+1} =& \operatorname*{argmin}_{\mathbf{y} \in \mathbb{R}^d} \left\langle \mathbf{g}_t^{(w)}, \mathbf{y} \right\rangle + \boldsymbol{\lambda}_k^\top (\mathbf{Q}\mathbf{y} - \mathbf{u}_k) + \frac{\rho}{2} \left\| \mathbf{Q}(\mathbf{y} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u}_k \right\|_2^2 \\ =& \operatorname*{argmin}_{\mathbf{y} \in \mathbb{R}^d} \left\langle \mathbf{g}_t^{(w)} + \mathbf{Q}^\top \boldsymbol{\lambda}_k, \mathbf{y} \right\rangle + \frac{\rho}{2} \left\| \mathbf{Q}(\mathbf{y} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u}_k \right\|_2^2\end{aligned}$$

According to KKT conditions, we obtain

$$\hat{\mathbf{y}}_{k+1} = \mathbf{x}_t - \frac{1}{\rho}\left(\mathbf{Q}^\top \mathbf{Q}\right)^{-1} \mathbf{g} + \mathbf{Q}^{-1}\left(\mathbf{u}_k - \frac{1}{\rho}\boldsymbol{\lambda}_k\right).$$

At the k-th iteration, the update of $\mathbf{u}$ is $\mathbf{u}_{k+1} = \operatorname{argmin}_{\mathbf{u} \in \mathbb{R}^d} L(\hat{\mathbf{y}}_{k+1}, \mathbf{u}, \boldsymbol{\lambda}_k)$, which is equivalent to

$$\mathbf{u}_{k+1} = \operatorname*{argmin}_{\mathbf{u} \in \mathbb{R}^d} \frac{1}{2\eta_t} \|\mathbf{u}\|_1^2 - \boldsymbol{\lambda}_k^\top \mathbf{u} + \frac{\rho}{2} \left\| \mathbf{Q}(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u} \right\|_2^2.$$

Here, $\|\mathbf{u}\|_1^2$ makes the update of $\mathbf{u}$ challenging due to its non-smooth and non-separable. We use a variational identity to replace it equivalently [4]. The variational identity is

$$\|\mathbf{u}\|_1^2 = \min_{\boldsymbol{\delta} \in \Delta_d} \sum_{i=1}^{d} \frac{(\mathbf{u}^i)^2}{\boldsymbol{\delta}^i},$$

where $\Delta_d := \{\boldsymbol{\delta} \in \mathbb{R}_+^d | \sum_{i=1}^d \boldsymbol{\delta}^i = 1\}$. Here, $\mathbf{u}^i$ and $\boldsymbol{\delta}^i$ represents the i-th element of $\mathbf{u}$ and $\boldsymbol{\delta}^i$, respectively. Since the variational identity is independent to $\mathbf{u}$, we obtain

$$\begin{aligned}
&\min_{\mathbf{u}\in\mathbb{R}^d} \frac{1}{2\eta}\|\mathbf{u}\|_1^2 - \boldsymbol{\lambda}_k^\top \mathbf{u} + \frac{\rho}{2}\left\|\mathbf{Q}(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u}\right\|_2^2 \\
=& \min_{\mathbf{u}\in\mathbb{R}^d} \left(\frac{1}{2\eta} \min_{\boldsymbol{\delta}\in\Delta_d} \sum_{i=1}^d \frac{(\mathbf{u}^i)^2}{\boldsymbol{\delta}^i}\right) - \boldsymbol{\lambda}_k^\top \mathbf{u} + \frac{\rho}{2}\left\|\mathbf{Q}(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u}\right\|_2^2 \\
=& \min_{\boldsymbol{\delta}\in\Delta_d} \min_{\mathbf{u}\in\mathbb{R}^d} \left(\frac{1}{2\eta} \sum_{i=1}^d \frac{(\mathbf{u}^i)^2}{\boldsymbol{\delta}^i}\right) - \boldsymbol{\lambda}_k^\top \mathbf{u} + \frac{\rho}{2}\left\|\mathbf{Q}(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u}\right\|_2^2 .
\end{aligned} \tag{6}$$

Fixing $\boldsymbol{\delta}$ and according to KKT conditions, for the i-th element of $\mathbf{u}$, i.e., $\mathbf{u}^i$, we obtain the optimum of $\mathbf{u}_*$

$$\mathbf{u}_*^i = \frac{\eta\boldsymbol{\delta}^i}{1+\rho\eta\boldsymbol{\delta}^i}\left(\boldsymbol{\lambda}_k^i + \rho\mathbf{Q}^i(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)})\right) \tag{7}$$

holds for any $1 \le i \le d$. Here, $\mathbf{Q}^i$ represents the i-th row of $\mathbf{Q}$. Substituting it into (6), we denote

$$h(\boldsymbol{\delta}^i) := \frac{1}{2\eta}\sum_{i=1}^d \frac{(\mathbf{u}_*^i)^2}{\boldsymbol{\delta}^i} - \boldsymbol{\lambda}_k^\top \mathbf{u}_* + \frac{\rho}{2}\left(\mathbf{Q}^i(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u}_*^i\right)^2 .$$

Thus, (6) is equivalently to

$$\min_{\boldsymbol{\delta}\in\Delta_d} \sum_{i=1}^d h(\boldsymbol{\delta}^i). \tag{8}$$

Using the efficient projection method proposed in [5], (8) is easily solved by using projected gradient descent method. The details are presented in Algorithm 2. We recommend [5] for more details about the efficient projection method.

When (8) is solved, and $\boldsymbol{\delta}_R$ is obtained, the update of $\mathbf{u}$ is immediately obtained by substituting $\boldsymbol{\delta}_R$ into (7). In summary, the update of $\mathbf{u}_{k+1}$ is

$$\mathbf{u}_{k+1}^i = \frac{\boldsymbol{\delta}_R^i}{2+\rho\boldsymbol{\delta}_R^i}\left(\boldsymbol{\lambda}_k^i + \rho\mathbf{Q}^i(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)})\right)$$

for any $1 \le i \le d$.

Finally, the update of $\boldsymbol{\lambda}_{k+1}$ is $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mathbf{Q}(\hat{\mathbf{y}}_{k+1} - \mathbf{x}_{t+1}^{(w)}) - \mathbf{u}_{k+1}$. Thus, the ADMM method to update $\mathbf{y}_{t+1}^{(w)}$ is presented in Algorithm 3.

4.2 Update of $\mathbf{z}_{t+1}^{(w)}$.

The update of $\mathbf{z}_{t+1}^{(w)}$ is a special case of the mirror descent with p-norm Bregman divergence [6]. It has the closed form, which is presented in Algorithm 4. We recommend [6] for more details about the closed form of the mirror descent with the p-norm Bregman divergence.

5 CONVERGENCE RATE

In this section, we first present the assumptions, and then present the convergence rate of the proposed method.

Algorithm 4 Closed form of the p-norm Bregman divergence [6] for the w-th worker at the t-th iteration

```
Require: The positive $\alpha_t$ and $q$ with $1 < q \le 2$. $\boldsymbol{\theta}_t^{(w)}$, and $\boldsymbol{\theta}_1^{(w)} = \mathbf{0}$. The stochastic gradient $\mathbf{g}_t^{(w)}$.
$\boldsymbol{\theta}_{t+1}^{(w)} \leftarrow \boldsymbol{\theta}_t^{(w)} - \alpha_t \mathbf{g}_t^{(w)}$, and $\hat{\boldsymbol{\theta}} := \boldsymbol{\theta}_{t+1}^{(w)}$.
For any $1 \le i \le d$, $\hat{\mathbf{z}}_{t+1}^i = \frac{\mathrm{sign}(\hat{\theta}^i)|\hat{\theta}^i|^{q-1}}{\|\hat{\theta}\|_q^{q-2}}$.
$\mathbf{z}_{t+1}^{(w)} = \hat{\mathbf{z}}_{t+1}$.
```

5.1 Assumptions

The assumptions used in the paper are presented as follows.

Assumption 1. *For any t and w, assume that* $\left\|\mathbf{g}_t^{(w)} - \nabla f(\mathbf{x}_t)\right\|_2^2 \le G$.

Assumption 2. *For any t, w, $\mathbf{u}$, and any vector $\mathbf{z}_t^{(w)}$, assume that $B_p(\mathbf{u}, \mathbf{z}_t^{(w)}) \le R$.*

Assumption 3. *Assume that $f(\cdot)$ is L_1-smooth with respect to $\|\cdot\|_1$, which guarantees that for any vectors $\mathbf{x}$ and $\mathbf{y}$, $f(\mathbf{x}) \le f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{L_1}{2}\|\mathbf{x} - \mathbf{y}\|_1^2$.*

These assumptions are the basic assumptions, and are widely used to analyze the convergence rate of an optimization method [6, 16].

5.2 Useful lemmas

Lemma 1 ([6]). *Given $\mathbf{g}_t^{(w)}$, θ_t, q, α_t, and $1 < p \le 2$, $\mathbf{z}_{t+1}^{(w)}$ is yielded by solving (5). For any vector $\mathbf{u}$, we have*

$$\alpha_t \left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_{t+1}^{(w)} - \mathbf{u} \right\rangle \le -B_p(\mathbf{z}_{t+1}^{(w)}, \mathbf{z}_t^{(w)}) + B_p(\mathbf{u}, \mathbf{z}_t^{(w)}) - B_p(\mathbf{u}, \mathbf{z}_{t+1}^{(w)}).$$

Lemma 2 ([6]). *Given a scalar $1 < p \le 2$, and any two vectors $\mathbf{x}$ and $\mathbf{y}$, we have $B_p(\mathbf{y}, \mathbf{x}) \ge \frac{p-1}{2}\|\mathbf{y} - \mathbf{x}\|_p^2$.*

Lemma 3 ([6]). *Given a scalar $p \ge 1$ and a vector $\mathbf{x} \in \mathbb{R}^d$, $\|\mathbf{x}\|_p \le \|\mathbf{x}\|_1 \le d^{1-\frac{1}{p}}\|\mathbf{x}\|_p$.*

Lemma 4. *Given that σ is the smallest singular value of $\mathbf{Q}$. For any vector $\mathbf{v} \in \mathbb{R}^d$, by setting $\eta_t = \frac{\tau\alpha_t\|\mathbf{Q}\|_1^2}{p-1} d^{\frac{2(p-1)}{p}}$, we have*

$$\begin{aligned} &\left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_t - \mathbf{v} \right\rangle - \frac{1}{2\eta_t}\|\mathbf{Q}(\mathbf{x}_t - \mathbf{v})\|_1^2 \\ \le & f(\mathbf{x}_{t+1}^{(w)}) - f(\mathbf{y}_{t+1}^{(w)}) + \frac{\tau\alpha_t\|\mathbf{Q}\|_1^2 d^{\frac{3p-2}{p}} G}{2((p-1)\sigma^2 - \tau L_1\alpha_t\|\mathbf{Q}\|_1^2 d^{\frac{3p-2}{p}})}, \end{aligned}$$

where $\mathbf{y}_{t+1}^{(w)} = \mathrm{argmin}_{\mathbf{y}\in\mathbb{R}^d} \left\langle \mathbf{g}_t^{(w)}, \mathbf{y} - \mathbf{x}_{t+1}^{(w)} \right\rangle + \frac{1}{\eta_t}\left\|\mathbf{Q}(\mathbf{y} - \mathbf{x}_{t+1}^{(w)})\right\|_1^2$.

The proof details of Lemma 4 are presented in the appendix.

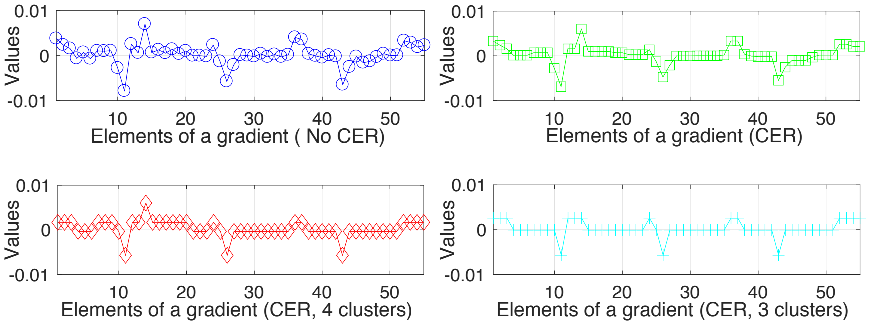


Fig. 3. Comparison of elements of $\mathbf{y}_{t+1}^{(w)}$. The first sub-figure is plotted by using the classic gradient descent method without the Communication Efficient Regularizer (CER). The second sub-figure is plotted by using our method, i.e., Eq. (4). The third and fourth sub-figures are plotted by using our method and the k-means clustering coding strategy with $k = 3$ and $k = 4$, respectively.

5.3 Sublinear convergence rate

The following theorem shows that the proposed method to update parameter with the communication efficient regularizer, i.e., Eq. (4) obtains the sublinear convergence rate. Comparing with the previous method, i.e., Eq. (2), our method does not impair the convergence rate theoretically.

Theorem 1. *Given that σ is the smallest singular value of $\mathbf{Q}$. Let $\eta_t = \frac{\sigma^2}{2L_1\sqrt{t}}d^{-1}$, and $\alpha_t = \frac{(p-1)\sigma^2}{L_1\|\mathbf{Q}\|_1^2\sqrt{t}d^{\frac{3p-2}{p}}}$. Denote $\mathbf{y}_* = \operatorname{argmin}_{\mathbf{y}\in\mathbb{R}^d} f(\mathbf{y})$. When Algorithm 1 is run for T iterations to yield $f(\mathbf{y}_T)$, it satisfies*

$$\begin{aligned}&\mathbb{E}f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t\right)-f(\mathbf{y}_*)\\ \le&\frac{2(f(\mathbf{y}_1)-f(\mathbf{y}_{T+1}))}{T}+\frac{2G}{L_1\sqrt{T}}+\frac{2L_1\|\mathbf{Q}\|_1^2 d^{\frac{3p-2}{p}}}{(p-1)\sigma^2}\left(\frac{R}{T}+\frac{R\sqrt{T+1}}{T}\right)\\ \lesssim&\frac{1}{T}+\frac{1}{\sqrt{T}}.\end{aligned}$$

The proof details are presented in the appendix.

Remark 1. *Theorem 1 implies that Algorithm 2 yields $O\left(\frac{1}{\sqrt{T}}\right)$ convergence rate.*

6 CODING STRATEGY

In this section, we present some coding strategies to encode the sequence $\{\mathbf{y}_t^{(w)}\}_{t=1}^T$.

As we have shown, the elements of $\mathbf{y}_{t+1}^{(w)}$ own some clustering structures. Exploiting the intrinsic clustering structures, it is able to encode $\mathbf{y}_{t+1}^{(w)}$ with few bits. We furthermore conduct clustering, e.g., k-means on elements of $\mathbf{y}_{t+1}^{(w)}$. Here, there is a trade-off between the accuracy and communication efficiency. When elements of a gradient are partitioned into more clusters, the higher accuracy of the gradient is guaranteed. Meanwhile, the gradient has to be encoded by using more bytes, thus leading to the decrease of the communication efficiency.

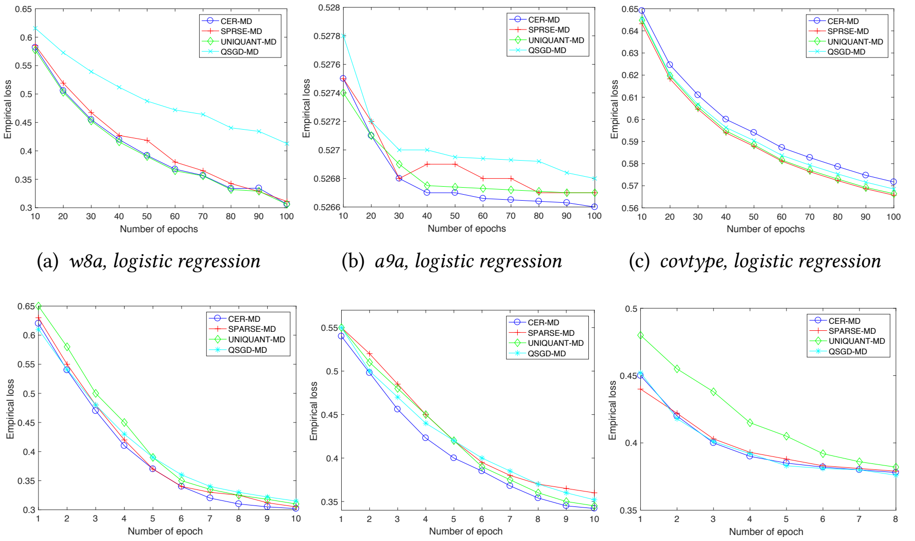
We take an example for more explanation. We conduct logistic regression on the *covtype* dataset. $\mathbf{y}_{t+1}^{(w)}$ has 55 elements. As illustrated in Figure 3, the x-axis represents the elements, and the y-axis represents values of an element. When the basic gradient descent method, i.e., Eq. (2) is used to yield $\mathbf{y}_{t+1}^{(w)}$, the top-left sub-figure shows that its elements are usually different. But, when the communication efficient regularizer is used to update $\mathbf{y}_{t+1}^{(w)}$, the top-right sub-figure shows that its elements can be partitioned into multiple clusters. When k-means clustering is furthermore conducted on those elements, the bottom-left and bottom-right sub-figures show that those elements can be represented with $k = 4$ and $k = 3$ cluster centers, respectively. Suppose every cluster center needs b bits to be encoded. Those cluster centers need kd bits. Additionally, we present two method to encode the clustering membership.

- **Elias integer encoding method.** For every element of $\mathbf{y}_{t+1}^{(w)}$, we use Elias integer coding method to encode the index of its cluster [20]. The code length is not larger than $\log d + 1$. For every cluster, the code length to represent its clustering membership is not larger than $d(\log d + 1)$. Finally, the required code length is not larger than $kd(\log d + 1)$. We recommend [2] for more details about the Elias integer encoding method.
- **Bloom filter encoding method.** For every cluster, we use the hashing method to map its members into a sequence of bits. Every member may be mapped to more than one bits by using more than one hash functions. If a member is mapped to a bit by using a hash function, the bit will turn to 1. Note that bloom filter may lead to false positive error. Suppose that the false positive probability is $p > 0$, and the required code length is thus $-\frac{kd \ln p}{(\ln 2)^2}$. We recommend [19] for more details about the Bloom filter.

7 EMPIRICAL STUDIES

7.1 Experimental settings

We conduct classification tasks via *logistic regression* and prediction tasks *ridge regression* to evaluate the proposed method. The formulation of the logistic regression is $\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{M}\sum_{i=1}^{M} \log(1 + \exp(-\mathbf{y}_i \mathbf{A}_i \mathbf{x}))$. The formulation of the ridge regression is $\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{M}\sum_{i=1}^{M} \|\mathbf{A}\mathbf{x} - \mathbf{y}_i\|^2 + \gamma \|\mathbf{x}\|^2$. Here, M represents the number of instances, $\mathbf{A}_i$ and $\mathbf{y}_i$ represent the i-th instance and its label, respectively. γ is a hyper-parameter which is given before conducting the task. Those two tasks are implemented by using the parameter server system *DMTK* [3]. The parameter server system is deployed by using five computing machines (four workers and one server).



(a) *w8a, logistic regression*

(b) *a9a, logistic regression*

(c) *covtype, logistic regression*

(d) *year, ridge regression*

(e) *cadata, ridge regression*

(f) *cpusmall, ridge regression*

Fig. 4. Comparsion of the empirical loss. Our method *CER-MD* obtains the comparable convergence performance with the state-of-the-art methods.

We conduct the logistic regression on datasets: *w8a*[2], *a9a*[3], and *covtype* [4], and conduct ridge regression on datasets: *year* [5], *cadata* [6] and *cpusmall* [7] The statics of those datasets are presented in Table 1. Additionally, our method is denoted by *CER-MD*, and the compared algorithms are presented as follows.

- **SPARSE-MD.** It proposes a gradient sparsification method to compress gradients [18].
- **UNIQUANT-MD.** It uses the uniformly random quantization method to compress gradients [24].
- **QSGD-MD.** It proposes a new gradient quantization and coding method to compress gradient. [2].

7.2 Numerical results

First, we compare the convergence performance. As illustrated in Figure 4, the y-axis represents the empirical loss, and the x-axis represents the number of epoches. Here, an epoch means the entire dataset is scanned once. Our proposed method, i.e., *CER-MD*

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#w8a

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a9a

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html
#covtype.binary

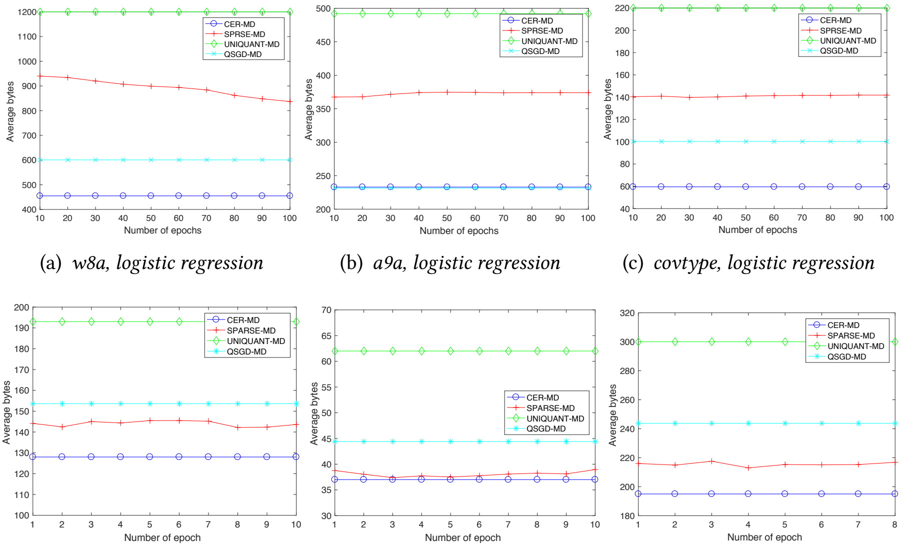
[5] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html
#YearPredictionMSD

[6] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html
#cadata

[7] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html
#cpusmall

Table 1. Statistics of datasets.

datasets	instances	features	tasks
w8a	49, 749	300	clssification, logistic regression
a9a	32, 561	123	clssification, logistic regression
covtype	581, 012	54	clssification, logistic regression
year	463, 715	90	prediction, ridge regression
cadata	20, 640	8	prediction, ridge regression
cpusmall	8192	12	prediction, ridge regression



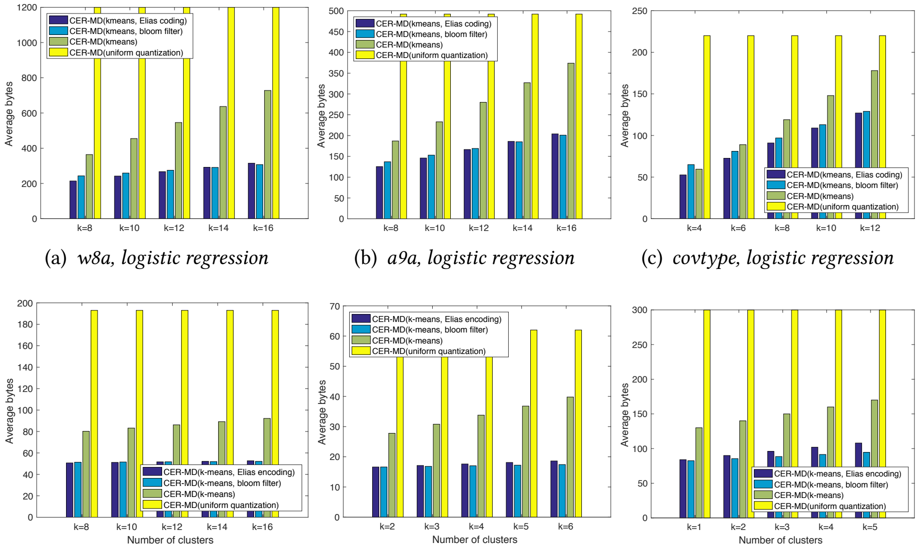
(a) *w8a, logistic regression* (b) *a9a, logistic regression* (c) *covtype, logistic regression*

(d) *year, ridge regression* (e) *cadata, ridge regression* (f) *cpusmall, ridge regression*

Fig. 5. Comparsion of the average bytes for data transimission. Our method *CER-MD* leads to the smallest amount of data transmission.

obtains the best convergence performance in the *a9a* dataset, and the comparable convergence performance with other methods in the *w8a* and *covtype*. It shows that our method *CER-MD* does not impair the convergence performance with the state-of-the-art methods. The reason is that our method jointly optimizes the empirical loss and the communication efficiency, which makes a good tradeoff between the convergence performance and the communication efficiency.

Second, we compare the workload of the data transmission. As shown in Figure 5, the y-axis represents the average bytes which are needed to be transmitted via network communication. The x-axis represents the number of epoches, which means how many times to scan the entire dataset. Our method *CER-MD* leads to much smaller workload of data transmission. The main reason is that *CER-MD* uses the communication efficient regularizer to encourage the update of parameter to own



(a) *w8a, logistic regression*

(b) *a9a, logistic regression*

(c) *covtype, logistic regression*

(d) *year, ridge regression*

(e) *cadata, ridge regression*

(f) *cpusmall, ridge regression*

Fig. 6. Comparsion of the average bytes for data transimission by varying coding methods.

clustering structures. The update of parameter can be encoded by using a few bits, and thus is compressed effectively.

Finally, we compare the workload of the data transmission for the proposed advanced coding methods. Those coding methods includes: *kmeans clustering coding method*, *uniform quantilization coding method*, *kmeans clustering and Elias coding method*, and *kmeans clustering and bloom filter coding method*. As illustrated in Figure 6, the x-axis represents the number of clusters. Since the uniform quantization method does not need to conduct clustering, the required amount of data transmission does not change with the number of clusters. We obtain some interesting observations. First, the kmeans clustering method is effective to reduce the amount of data transmission, but it leads to a large amount of data transmission with the increase of clusters. Second, with the increase of the clusters, the Elias coding method leads to more data transmission than the bloom filter method.

8 CONCLUSION

We propose a communication efficient regularizer to jointly optimize a machine learning model and the communication efficiency. To solve it efficiently, we propose a new ADMM method. By using the linear coupling strategy, the proposed communication efficient update of the parameter obtains the sublinear convergence rate, and outperforms the state-of-the-art methods in extensive empirical studies.

Proof to Lemma 4:

Proof. Since $\mathbf{y}_{t+1}^{(w)} = \operatorname{argmin}_{\mathbf{y}\in\mathbb{R}^d} \left\langle \mathbf{g}_t^{(w)}, \mathbf{y} - \mathbf{x}_{t+1}^{(w)} \right\rangle + \frac{1}{2\eta_t} \left\| \mathbf{Q}(\mathbf{y} - \mathbf{x}_{t+1}^{(w)}) \right\|_1^2$, by setting $\eta_t = \frac{\tau\alpha_t \|\mathbf{Q}\|_1^2}{p-1} d^{\frac{2(p-1)}{p}}$ and denoting $C := \frac{\tau\|\mathbf{Q}\|_1^2}{(p-1)\sigma^2} d^{\frac{3p-2}{p}}$, we have

$$\begin{aligned}
&\left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_t - \mathbf{v} \right\rangle - \frac{1}{2\eta_t} \|\mathbf{Q}(\mathbf{x}_t - \mathbf{v})\|_1^2 \\
\leq &\left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_t - \mathbf{y}_{t+1}^{(w)} \right\rangle - \frac{1}{2\eta_t} \left\| \mathbf{Q}(\mathbf{x}_t - \mathbf{y}_{t+1}^{(w)}) \right\|_1^2 \\
\overset{\textcircled{1}}{\leq} &\left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_t - \mathbf{y}_{t+1}^{(w)} \right\rangle - \frac{1}{2d\eta_t} \left\| \mathbf{Q}(\mathbf{x}_t - \mathbf{y}_{t+1}^{(w)}) \right\|_2^2 \\
\overset{\textcircled{1}}{\leq} &\left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_t - \mathbf{y}_{t+1}^{(w)} \right\rangle - \frac{1}{2\eta_t} \left\| \mathbf{Q}(\mathbf{x}_t - \mathbf{y}_{t+1}^{(w)}) \right\|_2^2 \\
\overset{\textcircled{2}}{\leq} &\left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_t - \mathbf{y}_{t+1}^{(w)} \right\rangle - \frac{\sigma^2}{2\eta_t} \left\| \mathbf{x}_t - \mathbf{y}_{t+1}^{(w)} \right\|_2^2 \\
\overset{\textcircled{3}}{\leq} &\left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_t - \mathbf{y}_{t+1}^{(w)} \right\rangle - \frac{\sigma^2}{2d\eta_t} \left\| \mathbf{x}_t - \mathbf{y}_{t+1}^{(w)} \right\|_1^2 \\
= &- \left(\left\langle \nabla f(\mathbf{x}_t), \mathbf{y}_{t+1}^{(w)} - \mathbf{x}_t \right\rangle + \frac{L_1}{2} \left\| \mathbf{y}_{t+1}^{(w)} - \mathbf{x}_t \right\|_1^2 \right) + \left\langle \nabla f(\mathbf{x}_t) - \mathbf{g}_t^{(w)}, \mathbf{y}_{t+1}^{(w)} - \mathbf{x}_t \right\rangle \\
&- \frac{1}{2} \left(\frac{1}{\alpha_t C} - L_1 \right) \left\| \mathbf{y}_{t+1}^{(w)} - \mathbf{x}_t \right\|_1^2 \\
\overset{\textcircled{4}}{\leq} &f(\mathbf{x}_t) - f(\mathbf{y}_{t+1}^{(w)}) + \frac{\tau\alpha_t \|\mathbf{Q}\|_1^2 d^{\frac{3p-2}{p}} \left\| \nabla f(\mathbf{x}_{t+1}^{(w)}) - \mathbf{g}_t^{(w)} \right\|_\infty^2}{2((p-1)\sigma^2 - \tau L_1 \alpha_t \|\mathbf{Q}\|_1^2 d^{\frac{3p-2}{p}})} \\
\leq &f(\mathbf{x}_t) - f(\mathbf{y}_{t+1}^{(w)}) + \frac{\tau\alpha_t \|\mathbf{Q}\|_1^2 d^{\frac{3p-2}{p}} G}{2((p-1)\sigma^2 - \tau L_1 \alpha_t \|\mathbf{Q}\|_1^2 d^{\frac{3p-2}{p}})}.
\end{aligned}$$

① holds because that for any vector $\mathbf{v} \in \mathbb{R}^d$, $\|\mathbf{v}\|_1 \geq \|\mathbf{v}\|_2$. ② holds because that for any vector $\mathbf{u} \in \mathbb{R}^d$, $\|\mathbf{Qu}\|_2 \geq \lambda_{\min(\mathbf{Q})} \|\mathbf{u}\|_2$. ③ holds because that for any vector $\mathbf{v} \in \mathbb{R}^d$, $\|\mathbf{v}\|_2 \geq \frac{1}{\sqrt{d}} \|\mathbf{v}\|_1$. ④ holds due to Assumption 3, and for any vectors $\mathbf{u}$ and $\mathbf{v}$, $\langle \mathbf{u}, \mathbf{v} \rangle \leq \frac{1}{2} \|\mathbf{u}\|_1^2 + \frac{1}{2} \|\mathbf{v}\|_\infty^2$. The proof is thus completed.

□

Proof to Theorem 1:

Proof. Consider $\mathbf{x}_{t+1}^{(w)} = \tau \mathbf{z}_t^{(w)} + (1-\tau)\mathbf{x}_t$ for $0 < \tau < 1$. For any vector $\mathbf{u} \in \mathbb{R}^d$, we have

$$\begin{aligned}
&f(\mathbf{x}_{t+1}^{(w)}) - f(\mathbf{u}) \leq \mathbb{E} \left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_{t+1}^{(w)} - \mathbf{u} \right\rangle \\
= &\left\langle \nabla f(\mathbf{x}_{t+1}^{(w)}), \mathbf{x}_{t+1}^{(w)} - \mathbf{z}_t^{(w)} \right\rangle + \left\langle \nabla f(\mathbf{x}_{t+1}^{(w)}), \mathbf{z}_t^{(w)} - \mathbf{u} \right\rangle \\
= &\frac{1-\tau}{\tau} \left\langle \nabla f(\mathbf{x}_{t+1}^{(w)}), \mathbf{x}_t - \mathbf{x}_{t+1}^{(w)} \right\rangle + \left\langle \nabla f(\mathbf{x}_{t+1}^{(w)}), \mathbf{z}_t^{(w)} - \mathbf{u} \right\rangle \\
\leq &\frac{1-\tau}{\tau} \left(f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}^{(w)}) \right) + \left\langle \nabla f(\mathbf{x}_{t+1}^{(w)}), \mathbf{z}_t^{(w)} - \mathbf{u} \right\rangle
\end{aligned} \tag{9}$$

Since $\left\langle \nabla f(\mathbf{x}_{t+1}^{(w)}), \mathbf{z}_t^{(w)} - \mathbf{u} \right\rangle = \mathbb{E} \left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_t^{(w)} - \mathbf{u} \right\rangle$, we begin to upper bound $\left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_t^{(w)} - \mathbf{u} \right\rangle$.

$$\begin{aligned}
&\left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_t^{(w)} - \mathbf{u} \right\rangle = \left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_t^{(w)} - \mathbf{z}_{t+1}^{(w)} \right\rangle + \left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_{t+1}^{(w)} - \mathbf{u} \right\rangle \\
\overset{①}{\le} & \left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_t^{(w)} - \mathbf{z}_{t+1}^{(w)} \right\rangle + \frac{1}{\alpha_t} \left(-B_p(\mathbf{z}_{t+1}^{(w)}, \mathbf{z}_t^{(w)}) + B_p(\mathbf{u}, \mathbf{z}_t^{(w)}) - B_p(\mathbf{u}, \mathbf{z}_{t+1}^{(w)}) \right) \\
\overset{②}{\le} & \left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_t^{(w)} - \mathbf{z}_{t+1}^{(w)} \right\rangle - \frac{p-1}{2\alpha_t} \left\| \mathbf{z}_{t+1}^{(w)} - \mathbf{z}_t^{(w)} \right\|_p^2 + \frac{1}{\alpha_t} \left(B_p(\mathbf{u}, \mathbf{z}_t^{(w)}) - B_p(\mathbf{u}, \mathbf{z}_{t+1}^{(w)}) \right) \\
\overset{③}{\le} & \left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_t^{(w)} - \mathbf{z}_{t+1}^{(w)} \right\rangle - \frac{p-1}{2\alpha_t} d^{-\frac{2(p-1)}{p}} \left\| \mathbf{z}_{t+1}^{(w)} - \mathbf{z}_t^{(w)} \right\|_1^2 + \frac{1}{\alpha_t} \left(B_p(\mathbf{u}, \mathbf{z}_t^{(w)}) - B_p(\mathbf{u}, \mathbf{z}_{t+1}^{(w)}) \right).
\end{aligned} \tag{10}$$

① holds due to Lemma 1. ② holds due to Lemma 2. ③ holds due to Lemma 3.

Define $\mathbf{v} := \tau \mathbf{z}_{t+1}^{(w)} + (1-\tau)\mathbf{y}_t$. Thus, $\mathbf{x}_t - \mathbf{v} = \tau(\mathbf{z}_t^{(w)} - \mathbf{z}_{t+1}^{(w)})$.

$$\begin{aligned}
& \left\langle \mathbf{g}_t^{(w)}, \mathbf{z}_t^{(w)} - \mathbf{z}_{t+1}^{(w)} \right\rangle - \frac{p-1}{2\alpha_t} d^{-\frac{2(p-1)}{p}} \left\| \mathbf{z}_{t+1}^{(w)} - \mathbf{z}_t^{(w)} \right\|_1^2 \\
=& \frac{1}{\tau} \left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_t - \mathbf{v} \right\rangle - \frac{p-1}{2\alpha_t \tau^2} d^{-\frac{2(p-1)}{p}} \left\| \mathbf{x}_t - \mathbf{v} \right\|_1^2 \\
\le& \frac{1}{\tau} \left(\left\langle \mathbf{g}_t^{(w)}, \mathbf{x}_t - \mathbf{v} \right\rangle - \frac{p-1}{2\tau \left\| \mathbf{Q} \right\|_1^2 \alpha_t} d^{-\frac{2(p-1)}{p}} \left\| \mathbf{Q}(\mathbf{x}_t - \mathbf{v}) \right\|_1^2 \right) \\
\overset{①}{\le}& \frac{1}{\tau} \left(f(\mathbf{x}_{t+1}^{(w)}) - f(\mathbf{y}_{t+1}^{(w)}) \right) + \frac{\alpha_t \left\| \mathbf{Q} \right\|_1^2 d^{\frac{3p-2}{p}} G}{2((p-1)\sigma^2 - \tau L_1 \alpha_t \left\| \mathbf{Q} \right\|_1^2 d^{\frac{3p-2}{p}})}.
\end{aligned} \tag{11}$$

① holds due to Lemma 4. Therefore, setting $\tau = \frac{1}{2}$, and substituting (11) and (10) into (9), we obtain

$$\begin{aligned}
& 2(f(\mathbf{y}_{t+1}^{(w)}) - f(\mathbf{u})) \\
\le& f(\mathbf{y}_t) - f(\mathbf{u}) + \frac{\alpha_t \left\| \mathbf{Q} \right\|_1^2 d^{\frac{3p-2}{p}} G}{2(p-1)\sigma^2 - L_1 \alpha_t \left\| \mathbf{Q} \right\|_1^2 d^{\frac{3p-2}{p}}} + \frac{1}{\alpha_t} \left(B_p(\mathbf{u}, \mathbf{z}_t^{(w)}) - B_p(\mathbf{u}, \mathbf{z}_{t+1}^{(w)}) \right).
\end{aligned}$$

When $\alpha_t = \frac{(p-1)\sigma^2}{L_1 \|\mathbf{Q}\|_1^2 d^{\frac{3p-2}{p}} \sqrt{t}}$ and $\eta_t = \frac{\tau \alpha_t \|\mathbf{Q}\|_1^2}{p-1} d^{\frac{2(p-1)}{p}} = \frac{\sigma^2}{2L_1\sqrt{t}} d^{-1}$, we have $2(p-1)\sigma^2 - L_1 \alpha_t \left\| \mathbf{Q} \right\|_1^2 d^{\frac{3p-2}{p}} \ge (p-1)\sigma^2$. Denote $\alpha := \frac{(p-1)\sigma^2}{L_1 \|\mathbf{Q}\|_1^2 d^{\frac{3p-2}{p}}}$, that is, $\alpha_t = \frac{\alpha}{\sqrt{t}}$.

$$2 \left(f(\mathbf{y}_{t+1}^{(w)}) - f(\mathbf{u}) \right) \le f(\mathbf{y}_t) - f(\mathbf{u}) + \frac{G}{L_1 \sqrt{t}} + \frac{\sqrt{t}}{\alpha} \left(B_p(\mathbf{u}, \mathbf{z}_t^{(w)}) - B_p(\mathbf{u}, \mathbf{z}_{t+1}^{(w)}) \right).$$

Thus, we have

$$\begin{aligned}
& 2(f(\mathbf{x}_{t+1}) - f(\mathbf{u})) = 2 \left(f\left(\frac{1}{W} \sum_{w=1}^{W} \mathbf{y}_{t+1}^{(w)} \right) - f(\mathbf{u}) \right) \\
\overset{①}{\le}& \frac{2}{W} \sum_{w=1}^{W} \left(f\left(\mathbf{y}_{t+1}^{(w)} \right) - f(\mathbf{u}) \right)
\end{aligned}$$

$$\leq f(\mathbf{y}_t) - f(\mathbf{u}) + \frac{G}{L_1\sqrt{t}} + \frac{\sqrt{t}}{\alpha W}\sum_{w=1}^{W}\left(B_p(\mathbf{u}, \mathbf{z}_t^{(w)}) - B_p(\mathbf{u}, \mathbf{z}_{t+1}^{(w)})\right).$$

① holds due to the convexity of $f(\cdot)$. Then, telescoping t from 1 to T, and setting $\mathbf{u} = \mathbf{y}_*$, we have

$$\sum_{t=1}^{T}(f(\mathbf{y}_t) - f(\mathbf{y}_*))$$

$$\leq 2(f(\mathbf{y}_1) - f(\mathbf{y}_{T+1})) + \frac{G}{L_1}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} + \frac{2}{\alpha W}\sum_{w=1}^{W}\left(B_p(\mathbf{y}_*, \mathbf{z}_1^{(w)}) + \sum_{t=2}^{T}(\sqrt{t+1} - \sqrt{t})B_p(\mathbf{y}_*, \mathbf{z}_t^{(w)})\right)$$

$$\overset{①}{\leq} 2(f(\mathbf{y}_1) - f(\mathbf{y}_{T+1})) + \frac{2G\sqrt{T}}{L_1} + \frac{2}{\alpha}\left(R + R\sqrt{T+1}\right).$$

① holds due to $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T}$ and $\sum_{t=2}^{T}(\sqrt{t+1} - \sqrt{t}) \leq \sqrt{T+1}$. Finally, we obtain

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t\right) - f(\mathbf{y}_*) \leq \frac{1}{T}\sum_{t=1}^{T}(f(\mathbf{y}_t) - f(\mathbf{y}_*))$$

$$\leq \frac{2(f(\mathbf{y}_1) - f(\mathbf{y}_{T+1}))}{T} + \frac{2G}{L_1\sqrt{T}} + \frac{2L_1\left\|\mathbf{Q}\right\|_1^2 d^{\frac{3p-2}{p}}}{(p-1)\sigma^2}\left(\frac{R}{T} + \frac{R\sqrt{T+1}}{T}\right).$$

□

A ACKNOWLEDGMENTS

This work was supported by the National Key R & D Program of China 2018YFB1003203 and the National Natural Science Foundation of China (Grant No. 61672528, 61773392, and 61671463).

REFERENCES

[1] Alham Aji and Kenneth Heafield. 2017. Sparse Communication for Distributed Gradient Descent. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*. 440–445.

[2] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'17)*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). 1709–1720.

[3] Microsoft Research Lab Asia. 2018. Distributed Machine Learning Toolkit. http://www.dmtk.io. Accessed Septemper 10, 2018.

[4] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. 2012. Optimization with Sparsity-Inducing Penalties. *Foundations & Trends in Machine Learning* 4, 1 (2012), 1–106.

[5] Duchi, John, ShalevShwartz, Shai, Singer, Yoram, Chandra, and Tushar. 2008. Efficient projections onto the l 1 -ball for learning in high dimensions. In *Proceedings of the International Conference on Machine Learning (ICML'08)*. 272–279.

[6] John C. Duchi, Shai Shalev-shwartz, Yoram Singer, and Ambuj Tewari. 2010. Composite objective mirror descent. In *Proceedings of the Annual Conference on Learning Theory (COLT)*. 14–26.

[7] Aaron Harlap, Henggang Cui, Wei Dai, Jinliang Wei, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson, and Eric P. Xing. 2016. Addressing the straggler problem for iterative convergent parallel ML. In *Proceedings of the ACM Symposium on Cloud Computing (SOCC'16)*. 98–111.

[8] Martin Jaggi, Virginia Smith, Martin Takávc, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I. Jordan. 2014. Communication-efficient Distributed Dual Coordinate Ascent. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14)*. 3068–3076.

[9] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. 2014. Scaling Distributed Machine Learning with the Parameter Server. *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI'14)* (2014).

[10] Mu Li, David G. Andersen, Alexander Smola, and Kai Yu. 2014. Communication Efficient Distributed Machine Learning with the Parameter Server. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14)*. 19–27.

[11] Sashank J Reddi, Jakub Konecný, Peter Richtárik, Barnabás Póczos, and Alexander J Smola. 2016. AIDE - Fast and Communication Efficient Distributed Optimization. *CoRR* math.OC (2016).

[12] Cèdric Renggli, Dan Alistarh, and Torsten Hoefler. 2018. SparCML: High-Performance Sparse Communication for Machine Learning. *arXiv.org* (2018). arXiv:cs.DC/1802.08021v1

[13] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press.

[14] Ohad Shamir, Nathan Srebro, and Tong Zhang. 2014. Communication-efficient Distributed Optimization Using an Approximate Newton-type Method. In *Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML'14)*. 1000–1008.

[15] Virginia Smith, Simone Forte, Chenxin Ma, Martin Takávc, Michael I Jordan, and Martin Jaggi. 2016. CoCoA: A General Framework for Communication-Efficient Distributed Optimization. *arXiv.org* (Nov. 2016). arXiv:cs.LG/1611.02189v1

[16] Chaobing Song, Shaobo Cui, Yong Jiang, and Shu-Tao Xia. 2017. Accelerated Stochastic Greedy Coordinate Descent by Soft Thresholding Projection onto Simplex. In *Proceedings of Advances in Neural Information Processing Systems (NIPS'17)*. 4838–4847.

[17] Jialei Wang, Weiran Wang, and Nathan Srebro. 2017. Memory and Communication Efficient Distributed Stochastic Optimization with Minibatch Prox. In *Proceedings of the Conference on Learning Theory (COLT'17)*, Vol. 65. 1882–1919.

[18] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. 2018. Gradient Sparsification for Communication-Efficient Distributed Optimization. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'18)*. 1–9.

[19] Wikipedia. 2018. Bloom filter. https://en.wikipedia.org/wiki/Bloom_filter. Accessed Septemper 10, 2018.

[20] Wikipedia. 2018. Elias gamma coding. https://en.wikipedia.org/wiki/Elias_gamma_coding. Accessed Septemper 10, 2018.

[21] Pengtao Xie, Jin Kyu Kim, Yi Zhou, Qirong Ho, Abhimanu Kumar, Yaoliang Yu, and Eric P Xing. 2016. Lighter-Communication Distributed Machine Learning via Sufficient Factor Broadcasting. *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI'16)* (2016).

[22] Eric P Xing, Qirong Ho, Wei Dai, Jin Kyu Kim, Jinliang Wei, Seunghak Lee, Xun Zheng, Pengtao Xie, Abhimanu Kumar, and Yaoliang Yu. 2015. Petuum: A New Platform for Distributed Machine Learning on Big Data. *IEEE Transactions on Big Data* 1, 2 (2015), 49–67.

[23] Min Ye and Emmanuel Abbe. 2018. Communication-Computation Efficient Gradient Coding. In *Proceedings of the International Conference on Machine Learning (ICML'18)*. 1–9.

[24] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. 2017. The ZipML Framework for Training Models with End-to-End Low Precision: The Cans, the Cannots, and a Little Bit of Deep Learning. In *Proceedings of the International Conference on Machine Learning (ICML'17)*. 1–9.

[25] Yuchen Zhang, John C Duchi, and Martin J Wainwright. 2013. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research* (2013), 3321–3363.

[26] Yuchen Zhang and Xiao Lin. 2015. DiSCO: Distributed Optimization for Self-Concordant Empirical Loss. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*. 362–370.