Communication Efficient Training of Federated Model Over Unbalanced Labels

Yawei Zhao, Qinghe Liu, Mingming Jiang, Kunlun He

***Abstract*—xxx**

***Index Terms*—xxxx**

I. Introduction

xxx

II. Related Work

III. Preliminary

A. Notations

IV. Formulation

A. Personalized Representation based on Similarity Network

Personalized models are built based on the similarity network. The similarity network measures the similarity of data distribution under data/feature/model space.

- **Data space.** In the case, local datasets of every node are used to construct a *kernel* matrix. The similarity of data distribution is measured by xxxxx.
- **Feature space.** In the case, local datasets of every node are used to construct a *covariance* matrix. It represents the dependence structure among features. The similarity of data distribution is measured based on the distance between covariance matrices.
- **Model space.** In the case, local model of every node is trained by using the local dataset. The similarity of data distribution is measured based on the distance between local models.

Based on similarity under those space, the similarity network $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ is built by using the KNN method [**?**]. $\mathcal{N} = \{1, 2, ..., N\}$ represents the node set, consisting of N nodes. $\mathcal{E} = \{e_{i,j} : i \in \mathcal{N}, j \text{ is the node } i\text{'s neighbour}\}$ represents the edge set, consisting of M edges.

B. Final Formulation

Given the mapping matrix $\mathbf{M} \in \mathbb{R}^{d \times d_1}$ and $\mathbf{N} \in \mathbb{R}^{d \times d_2}$, the objective problem is formulated by

$$\min_{\{\mathbf{x}^{(n)}\}_{n=1}^{N}} \frac{1}{N} \sum_{n \in \mathcal{N}} f_n \left(\mathbf{x}^{(n)}; \mathcal{D}_n \right) + \lambda \sum_{\substack{e_{i,j} \in \mathcal{E}, \\ \forall i,j \in \mathcal{N}}} \left\| \mathbf{z}^{(i)} - \mathbf{z}^{(j)} \right\|_p ,$$

Yawei Zhao, Qinghe Liu, Mingming Jiang, and Kunlun He are with the Medical Big Data Research Center, Chinese PLA General Hospital, Beijing, 100039, China. E-mail: csyawei.zhao@gmail.com, Liuqinghe9638@163.com, jiangmingming1994@163.com, kunlunhe@plagh.org.

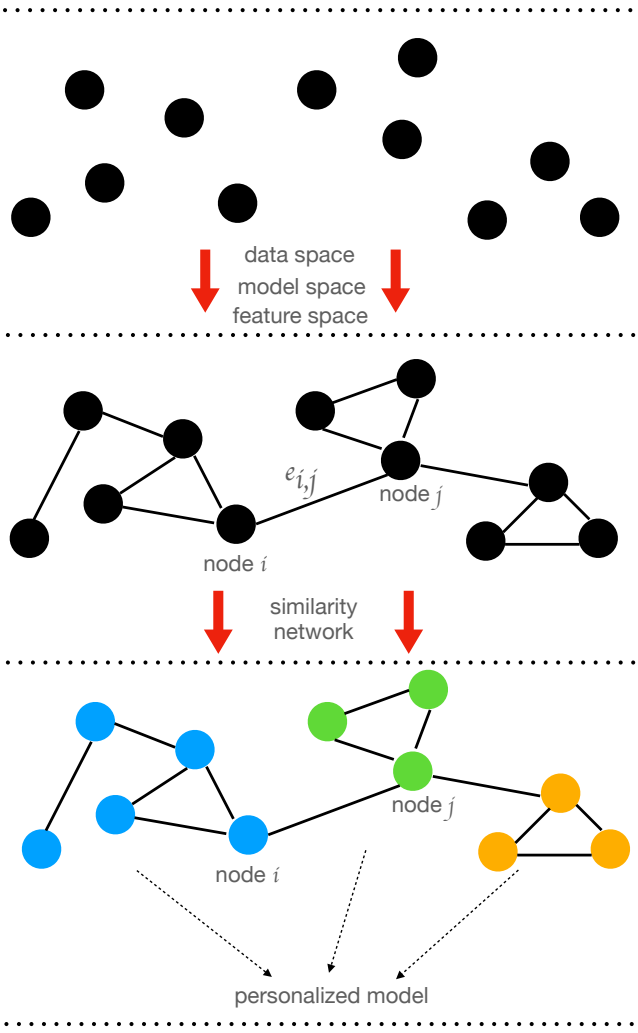


Fig. 1. Personalized representation based on similarity network

subject to:

$$\mathbf{x}^{(n)} = \mathbf{M}\mathbf{x} + \mathbf{N}\mathbf{z}^{(n)}, \quad \forall n \in \mathcal{N}, \mathbf{x} \in \mathbb{R}^{d_1}, \mathbf{z}^{(n)} \in \mathbb{R}^{d_2}.$$

Here, $p \in \{1, 2, \infty\}$. $\mathbf{M}$ and $\mathbf{N}$ has special structure, where every row of them has at most one non-zero value, and the non-zero value is 1.

The formulation can be equally transformed as follows.

$$\min_{\{\mathbf{x}^{(n)}\}_{n=1}^{N}} \frac{1}{N} \sum_{n=1}^{N} f_n \left(\mathbf{x}^{(n)}; \mathcal{D}_n \right) + \lambda \left\| \mathbf{ZQ} \right\|_{1,p}, \tag{1}$$

subject to:

$$\mathbf{x}^{(n)} = \mathbf{M}\mathbf{x} + \mathbf{N}\mathbf{z}^{(n)}. \tag{2}$$

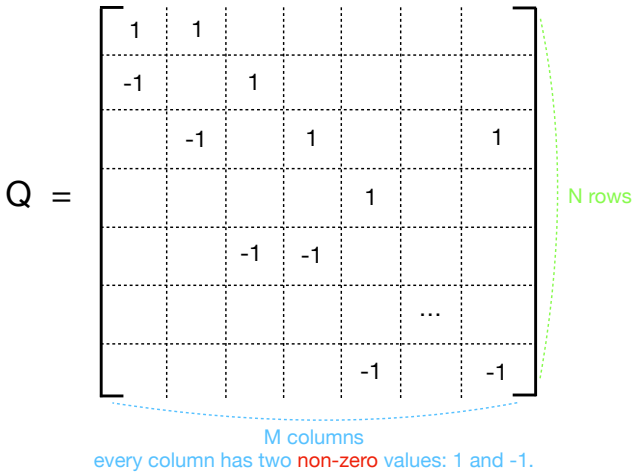


Fig. 2. $\mathbf{Q}$ matrix.

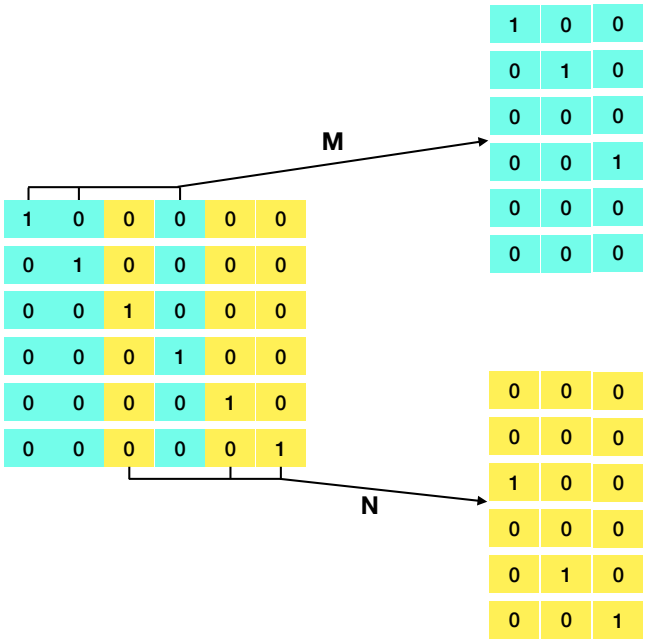


Fig. 3. for example: $\mathbf{M}$ and $\mathbf{N}$

Here, $\mathbf{Z} \in \mathbb{R}^{d_2 \times N}$, $\mathbf{Q} \in \mathbb{R}^{N \times M}$. N and M represent the total number of nodes and edges in the network $\mathcal{G}$, respectively. $\mathbf{Z}$ represents some a variable matrix, consisting of N variables as columns, that is, $\mathbf{Z} = \left[\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(N)}\right]$. $\mathbf{Q}$ is the given auxiliary matrix, which has M columns and every column has two non-zero values: 1 and -1. Note that $\|\cdot\|_{1,p}$ is denoted by $\ell_{1,p}$ norm. Given a matrix $\mathbf{U} \in \mathbb{R}^{d_2 \times M}$, it is defined by

$$\|\mathbf{U}\|_{1,p} := \sum_{m=1}^{M} \|\mathbf{U}_{:,m}\|_p .$$

C. Optimization via Alternative method

$$\min_{\{\mathbf{x}^{(n)}\}_{n=1}^{N}} \frac{1}{N} \sum_{n=1}^{N} f_n \left(\mathbf{x}^{(n)}; \mathcal{D}_n\right) + \lambda \|\mathbf{ZQ}\|_{1,p}, \tag{3}$$

subject to:

$$\mathbf{x}^{(n)} = \mathbf{M}\mathbf{x} + \mathbf{N}\mathbf{z}^{(n)}. \tag{4}$$

The variable $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$ is obtained by alternatively optimizing $\mathbf{x}$ and $\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(N)}\}$.

Optimizing $\mathbf{x}$ by given $\mathbf{Z}$. $\mathbf{x}$ is optimized by solving the following problem.

$$\min_{\mathbf{x} \in \mathbb{R}^{d_1}} \frac{1}{N} \sum_{n=1}^{N} f_n \left(\mathbf{M}\mathbf{x} + \mathbf{N}\mathbf{z}^{(n)}; \mathcal{D}_n\right). \tag{5}$$

By using the data-driven stochastic optimization method [**?**], we need to perform the following problem to obtain $\mathbf{x}$, iteratively.

$$\min_{\mathbf{x} \in \mathbb{R}^{d_1}} \frac{1}{N} \sum_{n=1}^{N} \left\langle \mathbf{M}^{-1}\mathbf{g}_t^{(n)}, \mathbf{x} \right\rangle + \frac{1}{2\eta_t} \|\mathbf{x} - \mathbf{x}_t\|^2 ,$$

where $\mathbf{g}_t^{(n)}$ is a stochastic gradient of f_n with $\mathbf{M}\mathbf{x} + \mathbf{N}\mathbf{z}_t^{(n)}$ by using data drawn from the local dataset $\mathcal{D}_n$.

Optimizing $\mathbf{Z}$ by given $\mathbf{x}$.

$$\min_{\mathbf{Z} \in \mathbb{R}^{d_2 \times N}} \frac{1}{N} \sum_{n=1}^{N} f_n \left(\mathbf{M}\mathbf{x} + \mathbf{N}\mathbf{z}^{(n)}; \mathcal{D}_n\right) + \lambda \|\mathbf{ZQ}\|_{1,p}. \tag{6}$$

By using the data-driven stochastic optimization method [**?**], we need to perform the following problem.

$$\min_{\mathbf{Z} \in \mathbb{R}^{d_2 \times N}} \frac{1}{N} \sum_{n=1}^{N} \left\langle \mathbf{N}^{-1}\mathbf{g}_t^{(n)}, \mathbf{z}^{(n)} \right\rangle + \lambda \|\mathbf{ZQ}\|_{1,p} + \frac{1}{2\eta_t} \|\mathbf{Z} - \mathbf{Z}_t\|_F^2 .$$

$\mathbf{g}_t^{(n)}$ is a stochastic gradient of f_n with $\mathbf{M}\mathbf{x}_t + \mathbf{N}\mathbf{z}^{(n)}$ by using stochastic data drawn from the local dataset $\mathcal{D}_n$. Suppose $\mathbf{G}_t = \left[\mathbf{g}_t^{(1)}, \mathbf{g}_t^{(2)}, ..., \mathbf{g}_t^{(N)}\right]$, $\mathbf{Z}$ is optimized by performing the following problem.

$$\min_{\mathbf{Z} \in \mathbb{R}^{d_2 \times N}} \frac{1}{N} \mathbf{1}_d^\top \left(\mathbf{G}_t \odot \mathbf{Z}\right) \mathbf{1}_N + \lambda \|\mathbf{ZQ}\|_{1,p} + \frac{1}{2\eta_t} \|\mathbf{Z} - \mathbf{Z}_t\|_F^2 .$$

On client.

Algorithm 1 Compute local stochastic gradient at the n-th client for the $t+1$-th iteration.

1: Receive the parameter $\mathbf{y}_t^{(n)} := \mathbf{M}\mathbf{x}_t + \mathbf{N}\mathbf{z}_t^{(n)}$ from the server.
2: Randomly sample an instance $\mathbf{a} \sim \mathcal{D}_n$, and compute the stochastic gradient $\mathbf{g}_t^{(n)} = \nabla f(\mathbf{y}_t^{(n)}; \mathbf{a})$ with $\mathbf{a} \sim \mathcal{D}_n$.
3: Send $\mathbf{g}_t^{(n)}$ to the server.

On server.

D. Workflow

V. Communication Efficient Update of Model

Recall that stochastic gradient descent is usually used to solve a general optimization problem in the distributed setting, namely Distributed Stochastic Gradient Descent (DSGD). Since every client has to send the update of parameter $\mathbf{g}_t^{(n)}$ to the server, the workload of the network communication

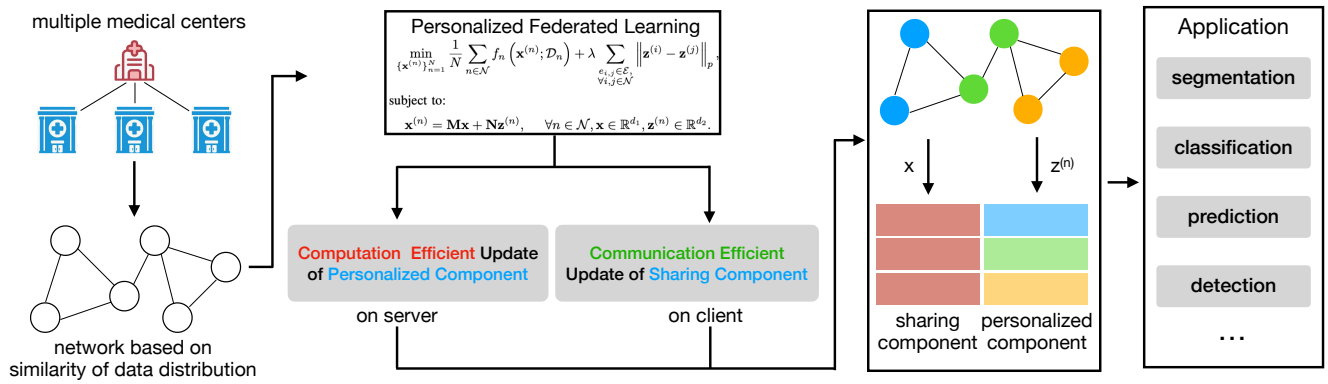


Fig. 4. workflow

Algorithm 2 Training of personalized models on the server.

Require: The number of total iterations T, and the initial parameter $\mathbf{x}_1$.

1: Deliver the parameter $\mathbf{y}_1^{(n)} = \mathbf{M}\mathbf{x}_1 + \mathbf{N}\mathbf{z}_1^{(n)}$ to all client n with $n \in \{1, 2, ..., N\}$.

2: **for** $t = 1, 2, ..., T$ **do**

3: Collect stochastic gradient $\mathbf{G}_t = \left[\mathbf{g}_t^{(1)}, \mathbf{g}_t^{(2)}, ..., \mathbf{g}_t^{(N)}\right]$ from all client n with $n \in \{1, 2, ..., N\}$.

4: Update the global parameter $\mathbf{x}$ by performing:

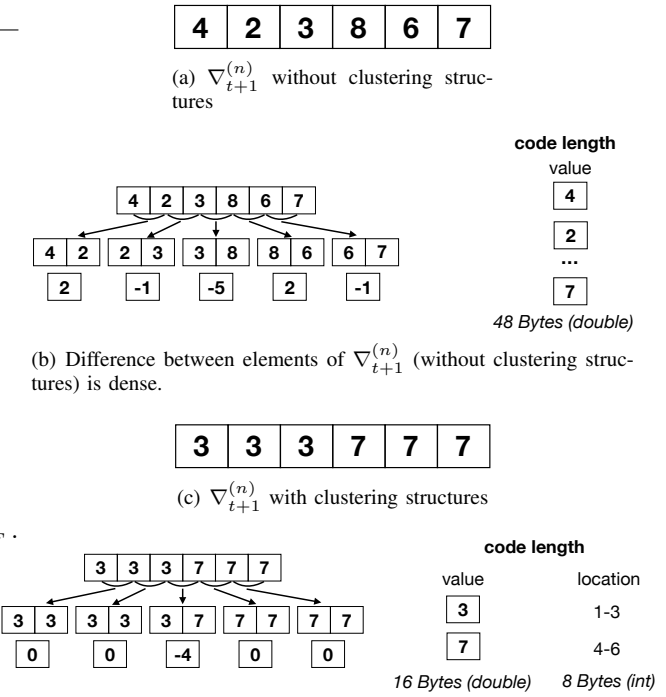
$$\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathbb{R}^{d_1}}{\operatorname{argmin}} \frac{1}{N}\sum_{n=1}^{N}\left\langle \mathbf{M}^{-1}\mathbf{g}_t^{(n)}, \mathbf{x}\right\rangle + \frac{1}{2\eta_t}\left\|\mathbf{x}-\mathbf{x}_t\right\|^2.$$

5: Update the personalized parameter $\mathbf{Z}$ by performing:

$$\mathbf{Z}_{t+1} = \underset{\mathbf{Z}\in\mathbb{R}^{d_2\times N}}{\operatorname{argmin}} \frac{\mathbf{1}_d^\top\left(\mathbf{G}_t \odot \mathbf{Z}\right)\mathbf{1}_N}{N} + \lambda\left\|\mathbf{Z}\mathbf{Q}\right\|_{1,p} + \frac{1}{2\eta_t}\left\|\mathbf{Z}-\mathbf{Z}_t\right\|_F^2.$$

6: Deliver the parameter $\mathbf{y}_{t+1}^{(n)} = \mathbf{M}\mathbf{x}_{t+1} + \mathbf{N}\mathbf{z}_{t+1}^{(n)}$ to every client.

return $\mathbf{x}_{T+1}^{(n)} = \mathbf{M}\mathbf{x}_{T+1} + \mathbf{N}\mathbf{z}_{T+1}^{(n)}$ with $n \in \{1, 2, ..., N\}$.



(a) $\nabla_{t+1}^{(n)}$ without clustering structures

(b) Difference between elements of $\nabla_{t+1}^{(n)}$ (without clustering structures) is dense.

(c) $\nabla_{t+1}^{(n)}$ with clustering structures

(d) Difference between elements of $\nabla_{t+1}^{(n)}$ (with clustering structures) is **sparse**.

Fig. 5. The illustrative example shows that $\nabla_{t+1}^{(n)}$ 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 $\nabla_{t+1}^{(n)}$ sparse.**

becomes very heavy for a large number of clients. 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.

A. Communication Efficient Regularizer

In the work, we propose a communication efficient method, which can let $\mathbf{g}_t^{(n)}$ be encoded by using few bits. Since the code length of $\mathbf{g}_t^{(n)}$ is much reduced, the communication efficiency is significantly increased. The basic idea is to induce the cluster structure of elements of $\mathbf{g}_t^{(n)}$ by using differential sparsity regularizer. The regularizer encourages the update of parameter $\nabla_{t+1}^{(n)}$ to own clustering structures. Figure 5 presents an illustrative example. According to Figures 5(a) and 5(c), we observe when the elements of $\nabla_{t+1}^{(n)}$ 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 clients and the server efficiently. According to Figures 5(b) and 5(d), our basic idea is to let the difference between the elements of $\nabla_{t+1}^{(n)}$ be sparse, which encourages the elements of $\nabla_{t+1}^{(n)}$ 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.

To improve the communication efficiency, we propose a new method to conduct the update of the parameter, which

is formulated as

$$\nabla_{t+1}^{(n)} = \frac{\mathbf{x}_t - \mathbf{v}}{\eta_t},$$

where $\mathbf{v}$ is obtained by performing the following problem Eq. 7.

$$\mathbf{v} = \underset{\mathbf{y}\in\mathbb{R}^d}{\operatorname{argmin}} \left\langle \mathbf{g}_t^{(n)}, \mathbf{y} \right\rangle + \underbrace{\gamma \left\| \mathbf{\Lambda}(\mathbf{y} - \mathbf{x}_t) \right\|_1}_{\text{communication efficient regularizer}} + \frac{\|\mathbf{y} - \mathbf{x}_t\|^2}{2\eta_t}. \tag{7}$$

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

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

Notice that $\mathbf{\Lambda}$ is a full rank square matrix, whose smallest singular value, denoted by σ, is positive, that is, $\sigma > 0$. The proposed communication efficient regularizer is an ℓ_1 norm square. It punishes the difference between elements of $\nabla_{t+1}^{(n)}$, and encourages them to be small or even zero. Thus, those corresponding elements of $\nabla_{t+1}^{(n)}$ are very similar or even identical. That is, the elements of $\nabla_{t+1}^{(n)}$ 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.

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

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., $\mathbf{\Lambda}(\mathbf{x}_2 - \mathbf{x}_1)$, contains as least two 0s. It explains our basic idea, that is, to let the difference between elements of successive update, i.e., $\mathbf{\Lambda}(\mathbf{x}_{t+1}^{(n)} - \mathbf{x}_t)$, be sparse.

Note that 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. $\nabla_{t+1}^{(n)}$ has 55 elements. As illustrated in Figure 6, the x-axis represents the elements, and the y-axis represents values of an element. When the basic gradient descent method, that is $\gamma = 0.$ is used to yield $\nabla_{t+1}^{(n)}$, the top-left sub-figure shows that its elements are usually different. But, when the communication efficient regularizer is used to update $\nabla_{t+1}^{(n)}$, 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.

Algorithm 3 Communication efficient update of personalized component on the n-th client for the $t+1$ iteration.

Require: A positive γ to improve communication efficiency.

1: Receive the parameter $\mathbf{y}_t^{(n)} := \mathbf{M}\mathbf{x}_t + \mathbf{N}\mathbf{z}_t^{(n)}$ from the server.

2: Randomly sample an instance $\mathbf{a} \sim \mathcal{D}_n$, and compute the stochastic gradient $\mathbf{g}_t^{(n)} = \nabla f(\mathbf{y}_t^{(n)}; \mathbf{a})$ with $\mathbf{a} \sim \mathcal{D}_n$.

3: Compute $\mathbf{v}$ by performing:

$$\mathbf{v} = \underset{\mathbf{y}\in\mathbb{R}^d}{\operatorname{argmin}} \left\langle \mathbf{g}_t^{(n)}, \mathbf{y} \right\rangle + \gamma \left\| \mathbf{\Lambda}(\mathbf{y} - \mathbf{x}_t) \right\|_1 + \frac{\|\mathbf{y} - \mathbf{x}_t\|^2}{2\eta_t}.$$

4: Obtain the communication efficient update of local model by performing:

$$\nabla_{t+1}^{(n)} = \frac{\mathbf{x}_t - \mathbf{v}}{\eta_t}.$$

5: Send $\nabla_{t+1}^{(n)}$ to the server.

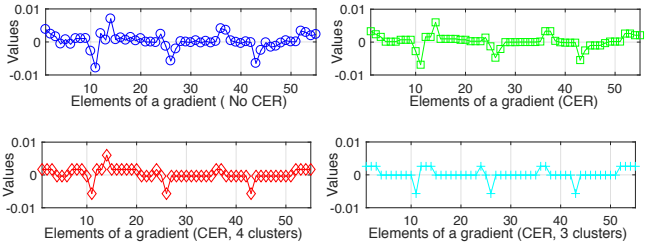


Fig. 6. Comparison of elements of $\nabla_{t+1}^{(n)}$. The first sub-figure is plotted by using the classic gradient descent method without the communication efficient regularizer. The second sub-figure is plotted by using our method. The third and fourth sub-figures are plotted by using our method and the k-means clustering with $k = 3$ and $k = 4$, respectively.

$$\nabla_{t+1}^{(n)} = \frac{\mathbf{x}_t - \mathbf{v}}{\eta_t},$$

where $\mathbf{v}$ is obtained by performing the following problem Eq. 7.

$$\mathbf{v} = \underset{\mathbf{y}\in\mathbb{R}^d}{\operatorname{argmin}} \left\langle \mathbf{g}_t^{(n)}, \mathbf{y} \right\rangle + \gamma \left\| \mathbf{\Lambda}(\mathbf{y} - \mathbf{x}_t) \right\|_1 + \frac{\|\mathbf{y} - \mathbf{x}_t\|^2}{2\eta_t}.$$

VI. Computation Efficient Update of Model

A. Efficient update of $\mathbf{x}$

When the server collects $\nabla_{t+1}^{(n)}$ from client, the sharing component $\mathbf{x}$ is updated by performing as follows.

$$\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathbb{R}^{d_1}}{\operatorname{argmin}} \left\langle \frac{1}{N}\sum_{n=1}^{N} \mathbf{M}^{-1}\nabla_{t+1}^{(n)}, \mathbf{x} \right\rangle + \frac{1}{2\eta_t} \|\mathbf{x} - \mathbf{x}_t\|^2.$$

Since it is a unconstrained optimization problem, it equals to the following update rule.

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \left(\frac{1}{N} \sum_{n=1}^{N} \mathbf{M}^{-1} \nabla_{t+1}^{(n)} \right).$$

That is, the sharing component $\mathbf{x}$ can be updated with low computational cost.

B. Efficient update of $\mathbf{Z}$

Denote

$$h(\mathbf{Z}) := \frac{\mathbf{1}_d^\top (\mathbf{G}_t \odot \mathbf{Z}) \mathbf{1}_N}{N} + \frac{1}{2\eta_t} \|\mathbf{Z} - \mathbf{Z}_t\|_F^2 .$$

The update of $\mathbf{Z}$ can be formulated by the following problem.

$$\min_{\mathbf{Z} \in \mathbb{R}^{d_2 \times N}, \mathbf{W} \in \mathbb{R}^{d_2 \times M}} H(\mathbf{Z}, \mathbf{W}) := h(\mathbf{Z}) + \lambda \|\mathbf{W}\|_{1,p},$$

subject to:

$$\mathbf{W} = \mathbf{Z}\mathbf{Q}.$$

Denote the augmented Lagrangian multiplier of $H(\mathbf{Z}, \mathbf{W})$ by $L(\mathbf{Z}, \mathbf{W}, \mathbf{\Omega})$, and we have

$$\begin{aligned} &L(\mathbf{Z}, \mathbf{W}, \mathbf{\Omega}) \\ :=& h(\mathbf{Z}) + \lambda \|\mathbf{W}\|_{1,p} + \mathbf{1}_{d_2}^\top (\mathbf{\Omega} \odot (\mathbf{W} - \mathbf{Z}\mathbf{Q})) \mathbf{1}_M + \frac{\rho}{2} \|\mathbf{W} - \mathbf{Z}\mathbf{Q}\|_F^2 . \end{aligned}$$

Update of Z. Given $\mathbf{W}_k$ and $\mathbf{\Omega}_k$, $\mathbf{Z}_{k+1}$ is obtained by performing the following problem.

$$\begin{aligned} \mathbf{Z}_{k+1} =& \operatorname*{argmin}_{\mathbf{Z} \in \mathbb{R}^{d_2 \times N}} L(\mathbf{Z}, \mathbf{W}_k, \mathbf{\Omega}_k) \\ =& \operatorname*{argmin}_{\mathbf{Z} \in \mathbb{R}^{d_2 \times N}} h(\mathbf{Z}) - \mathbf{1}_{d_2}^\top (\mathbf{\Omega}_k \odot (\mathbf{Z}\mathbf{Q})) \mathbf{1}_M + \frac{\rho}{2} \|\mathbf{W}_k - \mathbf{Z}\mathbf{Q}\|_F^2 . \end{aligned}$$

Since it is unconstrained optimization problem, we can obtain $\mathbf{Z}_{k+1}$ as follows.

$$\frac{\mathbf{G}_t}{N} + \frac{\mathbf{Z}_{k+1} - \mathbf{Z}_t}{\eta_t} - \mathbf{\Omega}_k \mathbf{Q}^\top + \rho(\mathbf{Z}_{k+1}\mathbf{Q} - \mathbf{W}_k)\mathbf{Q}^\top = \mathbf{0}$$

That is, we have

$$\begin{aligned} &\mathbf{Z}_{k+1} \\ =& \left[\eta_t \left(\mathbf{\Omega}_k \mathbf{Q}^\top - \frac{\mathbf{G}_t}{N} + \rho \mathbf{W}_k \mathbf{Q}^\top \right) + \mathbf{Z}_t \right] \left(\mathbf{I}_N + \eta_t \rho \mathbf{Q}\mathbf{Q}^\top \right)^{-1} . \end{aligned}$$

Update of W. Given $\mathbf{Z}_{k+1}$ and $\mathbf{\Omega}_k$, $\mathbf{W}_{k+1}$ is obtained by performing the following problem.

$$\begin{aligned} \mathbf{W}_{k+1} =& \operatorname*{argmin}_{\mathbf{W} \in \mathbb{R}^{d_2 \times M}} L(\mathbf{Z}_{k+1}, \mathbf{W}, \mathbf{\Omega}_k) \\ =& \operatorname*{argmin}_{\mathbf{W} \in \mathbb{R}^{d_2 \times M}} \lambda \|\mathbf{W}\|_{1,p} + \mathbf{1}_{d_2}^\top (\mathbf{\Omega}_k \odot \mathbf{W}) \mathbf{1}_M + \frac{\rho}{2} \|\mathbf{W} - \mathbf{Z}_{k+1}\mathbf{Q}\|_F^2 \\ =& \operatorname*{argmin}_{\mathbf{W} \in \mathbb{R}^{d_2 \times M}} \lambda \|\mathbf{W}\|_{1,p} + \frac{\rho}{2} \left\| \mathbf{W} - \left(\mathbf{Z}_{k+1}\mathbf{Q} - \frac{1}{\rho} \mathbf{\Omega}_k \right) \right\|_F^2 \\ =& \mathbf{Prox}_{\frac{\rho}{\lambda}, \|\cdot\|_{1,p}} \left(\mathbf{Z}_{k+1}\mathbf{Q} - \frac{1}{\rho} \mathbf{\Omega}_k \right). \end{aligned}$$

Here, **Prox** represents the *proximal operator*, which is defined by

$$\mathbf{Prox}_{\nu, \phi}(\mathbf{a}) := \operatorname*{argmin}_{\mathbf{b}} \phi(\mathbf{b}) + \frac{\nu}{2} \|\mathbf{b} - \mathbf{a}\|^2 .$$

Recall that $\|\cdot\|_{1,p}$ is the sum of norms, its proximal operator has a closed form [**?**]. Specifically, for the m-th column with $m \in \{1, 2, ..., M\}$ of $\mathbf{W}_{k+1}$ is obtained by performing:

$$\begin{aligned} [\mathbf{W}_{k+1}]_{:,m} =& \left[\mathbf{Prox}_{\frac{\rho}{\lambda}, \|\cdot\|_{1,p}} \left(\mathbf{Z}_{k+1}\mathbf{Q} - \frac{1}{\rho} \mathbf{\Omega}_k \right) \right]_{:,m} \\ =& \left(1 - \frac{1}{\left\| \frac{\rho}{\lambda} \mathbf{Z}_{k+1}\mathbf{Q}_{:,m} - \frac{1}{\lambda} [\mathbf{\Omega}_k]_{:,m} \right\|_q} \right)_+ \left(\mathbf{Z}_{k+1}\mathbf{Q}_{:,m} - \frac{1}{\rho} [\mathbf{\Omega}_k]_{:,m} \right), \end{aligned}$$

where $[\mathbf{A}]_{:,m}$ represents the m-th column of $\mathbf{A}$, and $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$ such that $\frac{1}{p} + \frac{1}{q} = 1$.

Update of Ω. Given $\mathbf{Z}_{k+1}$ and $\mathbf{W}_{k+1}$, $\mathbf{\Omega}_{k+1}$ is obtained by performing the following problem.

$$\mathbf{\Omega}_{k+1} = \mathbf{\Omega}_k + \rho \left(\mathbf{W}_{k+1} - \mathbf{Z}_{k+1}\mathbf{Q} \right).$$

Algorithm 4 Computation Efficient Update of $\mathbf{Z}$.

Require: The number of total iterations K, a positive ρ, and the initial model $\mathbf{Z}_t$.

1: **for** $k = 0, 1, 2, ..., K-1$ **do**

2: Update $\mathbf{Z}_{k+1}$ by performing:

$$\begin{aligned} &\mathbf{Z}_{k+1} \\ =& \left[\eta_t \left(\mathbf{\Omega}_k \mathbf{Q}^\top - \frac{\mathbf{G}_t}{N} + \rho \mathbf{W}_k \mathbf{Q}^\top \right) + \mathbf{Z}_t \right] \left(\mathbf{I}_N + \eta_t \rho \mathbf{Q}\mathbf{Q}^\top \right)^{-1} . \end{aligned}$$

3: Update the m-th column of $\mathbf{W}_{k+1}$, that is $[\mathbf{W}_{k+1}]_{:,m}$ by performing:

$$\begin{aligned} &[\mathbf{W}_{k+1}]_{:,m} \\ =& \left[1 - \frac{1}{\left\| \frac{\rho}{\lambda} \mathbf{Z}_{k+1}\mathbf{Q}_{:,m} - \frac{1}{\lambda} [\mathbf{\Omega}_k]_{:,m} \right\|_q} \right]_+ \left[\mathbf{Z}_{k+1}\mathbf{Q}_{:,m} - \frac{[\mathbf{\Omega}_k]_{:,m}}{\rho} \right], \end{aligned}$$

4: Update $\mathbf{\Omega}_{k+1}$ by performing:

$$\mathbf{\Omega}_{k+1} = \mathbf{\Omega}_k + \rho \left(\mathbf{W}_{k+1} - \mathbf{Z}_{k+1}\mathbf{Q} \right).$$

5: **return** $\mathbf{Z}_K$.

VII. Empirical Studies

Acknowledgment

This work was supported by the xx.

References

[1] C. Song, S. Cui, Y. Jiang, and S.-T. Xia, "Accelerated stochastic greedy coordinate descent by soft thresholding projection onto simplex," in *Proceedings of Advances in Neural Information Processing Systems (NIPS'17)*, 2017, pp. 4838–4847.

[2] J. C. Duchi, S. Shalev-shwartz, Y. Singer, and A. Tewari, "Composite objective mirror descent," in *Proceedings of the Annual Conference on Learning Theory (COLT)*, 2010, pp. 14–26.

[3] Wikipedia, "Elias gamma coding," https://en.wikipedia.org/wiki/Elias_gamma_coding, 2018, accessed Septemper 10, 2018.

[4] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "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., 2017, pp. 1709–1720.

[5] Wikipedia, "Bloom filter," https://en.wikipedia.org/wiki/Bloom_filter, 2018, accessed Septemper 10, 2018.

Algorithm 5 Computation Efficient Training of personalized models on the server.

Require: The number of total iterations T, and the initial parameter $\mathbf{x}_1$.

Deliver the parameter $\mathbf{y}_1^{(n)} = \mathbf{M}\mathbf{x}_1 + \mathbf{N}\mathbf{z}_1^{(n)}$ to all client n with $n \in \{1, 2, ..., N\}$.

for $t = 1, 2, ..., T$ **do**

Collect stochastic gradient $\mathbf{G}_t = \left[\mathbf{g}_t^{(1)}, \mathbf{g}_t^{(2)}, ..., \mathbf{g}_t^{(N)}\right]$ from all client n with $n \in \{1, 2, ..., N\}$.

Update the global parameter $\mathbf{x}$ by performing:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \left(\frac{1}{N} \sum_{n=1}^{N} \mathbf{M}^{-1} \nabla_{t+1}^{(n)} \right).$$

Update the personalized model $\mathbf{Z}_{t+1}$ according to Algorithm 4.

Deliver the parameter $\mathbf{y}_{t+1}^{(n)} = \mathbf{M}\mathbf{x}_{t+1} + \mathbf{N}\mathbf{z}_{t+1}^{(n)}$ to every client.

return $\mathbf{x}_{T+1}^{(n)} = \mathbf{M}\mathbf{x}_{T+1} + \mathbf{N}\mathbf{z}_{T+1}^{(n)}$ with $n \in \{1, 2, ..., N\}$.