Communication Efficient Tensor Factorization for Decentralized Healthcare Networks

Jing Ma[1], Qiuchen Zhang[1], Jian Lou[1,2], Li Xiong[1], Sivasubramanium Bhavani[1], Joyce C. Ho[1]
[1]*Emory University, Atlanta, Georgia*
[2]*Xidian University, Guangzhou*
{jing.ma, qiuchen.zhang, jian.lou, lxiong, sivasubramanium.bhavani, joyce.c.ho}@emory.edu, {jlou}@xidian.edu.cn

2021 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-2398-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDM51629.2021.00147

***Abstract*—Tensor factorization has been proved as an efficient unsupervised learning approach for health data analysis, especially for computational phenotyping, where the high-dimensional Electronic Health Records (EHRs) with patients history of medical procedures, medications, diagnosis, lab tests, etc., are converted to meaningful and interpretable medical concepts. Federated tensor factorization distributes the tensor computation to multiple workers under the coordination of a central server, which enables jointly learning the phenotypes across multiple hospitals while preserving the privacy of the patient information. However, existing federated tensor factorization algorithms encounter the single-point-failure issue with the involvement of the central server, which is not only easily exposed to external attacks, but also limits the number of clients sharing information with the server under restricted uplink bandwidth. In this paper, we propose `CiderTF`, a communication-efficient decentralized generalized tensor factorization, which reduces the uplink communication cost by leveraging a four-level communication reduction strategy designed for a generalized tensor factorization, which has the flexibility of modeling different tensor distribution with multiple kinds of loss functions. Experiments on two real-world EHR datasets demonstrate that `CiderTF` achieves comparable convergence with the communication reduction up to 99.99%.**

***Index Terms*—Tensor Factorization, Decentralized Optimization, Federated Learning, Communication efficient, EHRs.**

I. Introduction

The widespread adoption of EHR systems has facilitated the rapid accumulation of the patients' clinical data from numerous medical institutions. Yet, successfully mining the massive, high-dimensional EHR data is a challenging task due to sparse, missing, and noisy measurements [1], [2]. Computational phenotyping is the process of mapping the high-dimensional EHR data into meaningful medical concepts, which characterize a patient's clinical behavior and corresponding treatments. Tensor factorization has been proven as an efficient unsupervised learning approach to automatically extract phenotypes without the process of manual labeling [3]–[5].

Recently, federated tensor factorization [6]–[8] has been developed as a special distributed tensor factorization paradigm which not only parallelizes the tensor computation, but is also able to preserve the data privacy by distributing the horizontally partitioned tensors to multiple medical institutions to avoid direct data sharing, and aims to learn the shared phenotypes through joint tensor factorization without communicating the individual-level data. Moreover, with the participation of different data sources, federated tensor factorization also helps mitigate the bias of analyzing data from single source, and achieves better generalizability.

Under the federated learning settings, the central server is the most important computation resource as it is in charge of picking clients to communicate at each iteration, aggregating the clients' intermediate results, and updating the global model. However, a single server might have several shortcomings: 1) limited connectivity and bandwidth, which restricts the server from collecting data from as many clients as possible; 2) vulnerability to malfunctions, which can cause inaccurate model updates, or even learning failures; and 3) exposure to external attacks and malicious adversaries, which can lead to sensitive information leakage. Therefore, traditional federated tensor factorization usually suffers from the bottleneck of the central server regarding the limited communication bandwidth and is exposed to high risk of single-point-failure. To avoid relying on the server as the only source of computation, decentralization has been proposed as a solution to this single-point-failure issue [9], [10]. Decentralized federated learning is designed without the participation of the central server, while each client will rely on its own computation resources and communicate only with its neighbors in a peer-to-peer manner. Besides the necessities of a decentralized communication topology, it is also worth noting that the network capacity between clients are usually much smaller than the datacenter in many real-world applications [11]. Therefore it is necessary that the clients communicate the model updates efficiently with limited cost.

In this paper, we study the decentralized optimization of tensor factorization under the horizontal data partition setting, and propose `CiderTF`, a Communication-effIcient DEcentralized geneRalized Tensor Factorization algorithm for collaborative analysis over a communication network. To enable more flexibility on choosing different loss functions under various scenarios, we extend the classic federated tensor factorization into a more generalized tensor factorization. To the best of our knowledge, this paper is the first one proposing a decentralized generalized tensor factorization, let alone considering the decentralized setting with communication efficiency. Our contributions are briefly summarized as follows.

First, we develop a decentralized tensor factorization framework which employs four levels of communication reduction strategies to the decentralized optimization of tensor factorization to reduce the communication cost over the communication network. Second, we further incorporate Nesterov's momentum

978-1-6654-2398-4/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00147

TABLE I
SYMBOLS AND NOTATIONS USED IN THIS PAPER

Symbol	Definition
$\mathbf{x}, \mathbf{X}, \mathcal{X}$	Vector, Matrix, Tensor
$\mathcal{X}_{<d>}$	Mode-d matricization of $\mathcal{X}$
$\|\cdot\|_1$	ℓ_1-norm
$\|\cdot\|_F$	Frobenius norm
$\circledast$	Hadamard (element-wise) multiplication
$\odot$	Khatri Rao product
$\circ$	Outer product
$\langle\cdot,\cdot\rangle$	Inner product

into the local updates of `CiderTF` and propose `CiderTF_m`, in order to achieve better generalization and faster convergence. Third, we conduct comprehensive experiments on both real-world and synthetic datasets to corroborate the theoretical communication reduction and the convergence of `CiderTF`. Experiment results demonstrate that `CiderTF` achieves comparable convergence performance with the communication reduction of 99.99%.

II. Preliminaries and Background

In this section, we summarize the frequently used definitions and notations. For a D-th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times ... \times I_D}$, the tensor entry indexed by $(i_1, ..., i_D)$ is denoted by the MATLAB representation $\mathcal{X}(i_1, ..., i_D)$. Let $\mathcal{I}$ denote the index set of all tensor entries, $|\mathcal{I}| = I_\Pi = \prod_{d=1}^{D} I_d$. The mode-$d$ unfolding (also called matricization) is denoted by $\mathbf{X}_{<d>} \in \mathbb{R}^{I_d \times I_\Pi / I_d}$. Detailed background knowledge can be found in [12].

Definition II.1. (*MTTKRP*). The MTTKRP operation stands for the *matricized tensor times Khatri-Rao product*. Given a tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times ... \times I_D}$, its mode-$d$ matricization is $\mathbf{Y}_{<d>}$, $[\mathbf{A}_{(1)}, ..., \mathbf{A}_{(D)}]$ is the set of CP factor matrices. $\mathbf{H}_d \in \mathbb{R}^{I_\Pi / I_d \times R}$ is defined as

$$\mathbf{H}_d = \mathbf{A}_{(D)} \odot ... \odot \mathbf{A}_{(d+1)} \odot \mathbf{A}_{(d-1)} ... \odot \mathbf{A}_{(1)},$$

where $\odot$ is the Khatri-Rao product. The MTTKRP operation can thus be defined as the matrix product between $\mathbf{Y}_{<d>}$ and $\mathbf{H}_d$ as $\mathbf{Y}_{<d>} \cdot \mathbf{H}_d$.

Definition II.2. (*GCP*). Generalized CP (GCP) [13] extends the classic CP by using the element-wise loss function to support other loss functions. The objective function of GCP is

$$\begin{aligned} \arg\min_{\mathcal{A}} F(\mathcal{A}, \mathcal{X}) &= \sum_{i \in \mathcal{I}} f(\mathcal{A}(i), \mathcal{X}(i)) \\ s.t.\ \mathcal{A} &= \sum_{i=1}^{R} \mathbf{A}_{(1)}(:, i) \circ ... \circ \mathbf{A}_{(D)}(:, i), \end{aligned} \tag{1}$$

GCP not only preserves the low-rank constraints as CP decomposition, it also enjoys the flexibility of choosing different loss functions according to different data distributions by leveraging the elementwise objective function. For example, for data indexed by $i \in \mathcal{I}$ with Gaussian distribution, we use least square loss to model it, which in turn yields the classic CP decomposition:

$$f_{\text{square}}(\mathcal{A}(i), \mathcal{X}(i)) = (\mathcal{A}(i) - \mathcal{X}(i))^2. \tag{2}$$

On the other hand, for binary data indexed by $i \in \mathcal{I}$, we can use Bernoulli-logit loss to fit it:

$$f_{\text{logit}}(\mathcal{A}(i), \mathcal{X}(i)) = \log(1 + \mathcal{A}(i)) - \mathcal{X}(i)\mathcal{A}(i). \tag{3}$$

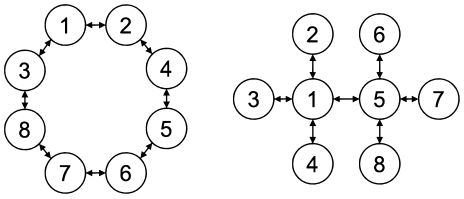


Fig. 1. Ring topology (left) and star topology (right).

III. Proposed Method

A. Problem Formulation

In the decentralized tensor factorization setting, the communication topology is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{1, 2, ..., K\}$ denotes the set of clients participating in the communication network. Each node k in the graph represents a client. The neighbors of client k is denoted as $\mathcal{N}_k := \{(k, j) : (k, j) \in \mathcal{E}\}$. There is a connectivity matrix $W \in \mathbb{R}^{K \times K}$, the (k, j)-th entry $w_{kj} \in [0, 1], \forall (k, j) \in \mathcal{E}$ in which denotes the weights of edge $(k, j) \in \mathcal{E}$ and measures how much the client k is impacted by client j.

Each client in the decentralized communication graph will hold a local tensor $\mathcal{X}^k$, which can be seen as the horizontal partition of a global tensor $\mathcal{X}$. The aim for the decentralized federated learning is to jointly factorize the local tensors $\mathcal{X}^k$ to get the globally shared feature factor matrices $\mathbf{A}_{(2)}, ..., \mathbf{A}_{(D)}$, and the individual mode factor matrices $\mathbf{A}^k_{(1)}$ from all clients. The objective function for the decentralized generalized tensor factorization is shown as

$$\begin{aligned} &\underset{(\mathbf{A}_{(1)}, ..., \mathbf{A}_{(D)})}{\text{argmin}} \sum_{k=1}^{K} F(\mathcal{A}, \mathcal{X}^k), \\ &s.t.\ \mathcal{A} = \mathbf{A}_{(1)} \circ ... \circ \mathbf{A}_{(D)}, \end{aligned} \tag{4}$$

which can be further extended to other multiblock optimization problems which are not limited to tensor factorization [14].

B. `CiderTF`

1) Overview: We propose `CiderTF`, a decentralized tensor factorization framework which achieves communication efficiency through four levels of communication reduction. At the element-level, we utilize sign compressor [15], [16] for gradient compression to reduce the number of bytes transmitted between clients by converting the partial gradient from the floating point representation to low-precision representation.

Definition III.1. (Sign Compressor) For an input tensor $\mathbf{x} \in \mathbb{R}^d$, its compression via $\texttt{Sign}(\cdot)$ is $\texttt{Sign}(\mathbf{x}) = \|\mathbf{x}\|_1 / d \cdot sign(\mathbf{x})$, where $sign$ takes the sign of each element of $\mathbf{x}$.

At the block-level, we apply the randomized block coordinate descent [17]–[19] for the factor updates, which only requires

Algorithm 1 `CiderTF`

Input: Input tensor $\mathcal{X}$, constant learning rate $\gamma[t]$, $\mathbf{A}[0], \mathbf{A}^k[0] = \mathbf{A}[0], \forall k = 1, ..., K$, randomized block sampling sequence $d_\xi[0], ..., d_\xi[T]$, event-triggering threshold $\lambda[t]$;

1: **for** $t = 0, ..., T$ **do**
2: **On Each Client Nodes** $k \in 1, ..., K$**:**
3: **if** $d = d_{(\xi)}[t]$ **then**
4: Compute stochastic gradient $\mathbf{G}^k_{(d)}[t]$ by eq. (6);
5: $\mathbf{A}^k_{(d)}[t + \frac{1}{2}] = \mathbf{A}^k_{(d)}[t] - \gamma[t]\mathbf{G}^k_{(d)}[t]$;
6: **if** $(t \mod \tau) \neq 0$ **then**
7: No communication:
$\mathbf{A}^k_{(d)}[t+1] = \mathbf{A}^k_{(d)}[t + \frac{1}{2}]$, $\widehat{\mathbf{A}}^k_{(d)}[t+1] = \widehat{\mathbf{A}}^k_{(d)}[t]$;
8: **else**
9: **for** $j \in \mathcal{N}_k \cup k$ **do**
10: **if** $\|\mathbf{A}^k_{(d)}[t + \frac{1}{2}] - \widehat{\mathbf{A}}^k_{(d)}[t]\|^2_F \geq \lambda[t](\gamma[t])^2$ **then**
11: $\mathbf{\Delta}^k_{(d)}[t] =$`Compress`$(\mathbf{A}^k_{(d)}[t + \frac{1}{2}] - \widehat{\mathbf{A}}^k_{(d)}[t])$;
12: **else**
13: $\mathbf{\Delta}^k_{(d)}[t] = \mathbf{0}_{I^k \times R}$;
14: **end if**
15: Send $\mathbf{\Delta}^k_{(d)}[t]$ to all j and receive $\mathbf{\Delta}^j_{(d)}[t]$ from all j, where $j \in \mathcal{N}_k$;
16: $\widehat{\mathbf{A}}^j_{(d)}[t+1] = \widehat{\mathbf{A}}^j_{(d)}[t] + \mathbf{\Delta}^j_{(d)}[t]$;
17: **end for**
18: $\mathbf{A}^k_{(d)}[t+1] = \mathbf{A}^k_{(d)}[t+\frac{1}{2}] + \varrho \sum_{j \in \mathcal{N}^k} w_{kj}(\widehat{\mathbf{A}}^j_{(d)}[t+1] - \widehat{\mathbf{A}}^k_{(d)}[t+1])$;
19: **end if**
20: **else if** $d \neq d_\xi[t]$ **then**
21: $\mathbf{A}^k_{(d)}[t+1] = \mathbf{A}^k_{(d)}[t]$, $\widehat{\mathbf{A}}^k_{(d)}[t+1] = \widehat{\mathbf{A}}^k_{(d)}[t]$;
22: **end if**
23: **end for**

sampling one mode from all modes of a tensor for the update per round and communicating only one mode factor updates with the neighbors. At the round-level, we adopt a periodic communication strategy [20]–[22] to reduce the communication frequency by allowing each client to perform $\tau > 1$ local update rounds before communicating with its neighbors. In addition, at the communication event-level, we apply an event-triggered communication strategy [23], [24] to boost the communication reduction at the round level.

The detailed algorithm is shown in Algorithm 1 with the key steps annotated. In `CiderTF`, each client $k \in [K]$ maintains the local factor matrices $\mathbf{A}^k_{(d)}$ from each mode $d = 1, ..., D$. The goal is to achieve consensus on the feature mode factor matrices $\mathbf{A}^k_{(d)}, \forall d = 2, ..., D$. Therefore, besides the local factor matrices, each client also need to maintain the estimation of the local factor matrices $\mathbf{A}^j_{(d)}$ from both itself k and its neighbors $\mathcal{N}_k$ ($j \in \mathcal{N}_k \cup k$). The sequence of the randomized sampling blocks for every round $t = 1, ..., T$ is denoted as $d_\xi[0], ..., d_\xi[T]$. At every round for the sampled block $d_\xi[t]$, each client checks for the triggering condition for every τ iterations at the communication round (line 10). The triggering threshold is set to be $\lambda[t]$. When the difference between the updated factor and the local estimation is larger than the threshold, each client will send and receive the compressed updates to its neighbors. While if the triggering condition is not satisfied, then the clients will just communicate a matrix of zero instead (line 10-14). After receiving the compressed updates from all its neighbors, each client will first update the local estimation of the factor matrices $\widehat{\mathbf{A}}^j_{(d)}[t+1], j \in \mathcal{N}_k \cup k$ (line 16), and conduct the consensus step and update the local factors $\mathbf{A}^k_{(d)}[t+1]$ through the decentralized consensus step (line 18). At the non-communication round, each client will just keep updating the local factor matrices (line 6-7). For the rest of the blocks not selected, they will remain the same at the last round (line 20-22).

2) Optimization: At each iteration, each client k first need to compute the GCP gradient as the partial derivative with regard to the factor matrix $\mathbf{A}^k_{(d)}$ using the MTTKRP operator

$$\frac{\partial F(\mathcal{A}^k, \mathcal{X}^k)}{\partial \mathbf{A}^k_{(d)}[t]} = \mathbf{Y}^k_{<d>}\mathbf{H}^k_d, \tag{5}$$

where $\mathbf{H}^k_d$ denotes the Khatri-Rao product of mode d of the factor matrices as is shown in definition II.1.

Fiber Sampling. Computing the full gradient $\frac{\partial F(\mathcal{A}^k, \mathcal{X}^k)}{\partial \mathbf{A}^k_{(d)}[t]}$ requires $O(R\prod_{d=1}^{D} I_d)$ time complexity and is the bottleneck of the gradient based optimization for tensor factorization, especially for EHR tensors where each dimension can be very large. Fiber sampling technique [18], [25] randomly samples $|\mathcal{S}_d|$ fibers from mode d. This provides efficient formation of $\mathbf{Y}^k_{<d>}$ as $\mathbf{Y}^k_{<d>}(:, \mathcal{S}_d)$ and efficient computation of $\mathbf{H}^k_d$ to only compute the Hadamard product ($\circledast$) of the certain rows (s-th) of the factor matrices at time t as $\mathbf{H}^k_d(s,:) = \mathbf{A}^k_{(1)}(i^s_1,:) \circledast ... \circledast \mathbf{A}^k_{(d-1)}(i^s_{d-1},:) \circledast \mathbf{A}^k_{(d+1)}(i^s_{d+1},:) \circledast ... \circledast \mathbf{A}^k_{(D)}(i^s_D,:)$ (the row indices are obtained from the index mapping $\{i^s_1, ..., i^s_D\}, s \in \mathcal{S}_d$). Therefore, we can use local partial stochastic gradient $\mathbf{G}^k_{(d)}[t]$ as an unbiased estimation of the gradient $\frac{\partial F(\mathcal{A}^k, \mathcal{X}^k)}{\partial \mathbf{A}^k_{(d)}[t]}$, which is efficiently computed with the fiber sampling technique as

$$\mathbf{G}^k_{(d)}[t] = \mathbf{Y}^k_{<d>}(:, \mathcal{S}_d)\mathbf{H}^k_d(\mathcal{S}_d, :), \tag{6}$$

Block randomization. We utilize the block randomization [18] to further improve the computation efficiency by randomly selecting a mode to update at each round. Specially for `CiderTF`, we always keep the patient mode (the 1-st mode) securely at local to avoid directly sharing patient related information, thus when $d_\xi[t] = 1$, we skip the communication of this round and only update the local patient mode factors. This not only improves the computation efficiency, but also reduces the communication cost at the block level.

C. `CiderTF_m`*:* `CiderTF` *with Nesterov's momentum*

We further propose `CiderTF_m` with Nesterov's momentum incorporated in the local SGD update step to speedup the convergence and achieve less total communication bits. After computing the partial stochastic gradient $\mathbf{G}^k_{(d)}[t]$ (line 4), we update the momentum velocity component as

$$\mathbf{M}^k_{(d)}[t] = \mathbf{G}^k_{(d)}[t] + \beta\frac{\eta[t-1]}{\eta[t]}\mathbf{M}^k_{(d)}[t-1] \tag{7}$$

where β is the momentum parameter. The intermediate factor matrix will be updated as

$$\mathbf{A}^k_{(d)}[t+\frac{1}{2}] = \mathbf{A}^k_{(d)}[t] - \gamma[t](\mathbf{G}^k_{(d)}[t] + \beta\mathbf{M}^k_{(d)}[t]) \quad (8)$$

D. Complexity Analysis

We analyze the complexity from the perspective of computation, communication, and memory cost. For computation complexity, the per-iteration computational complexity of `CiderTF` for each client is $O(\frac{1}{D}(\sum_{d=1}^{D} I_d)R|\mathcal{S}|)$. `CiderTF` reduces a lower bound of $1 - \frac{1}{32D\tau}$ communication. The total communication reduction is 99.99% compared with the full precision decentralized SGD based on experimental results. `CiderTF` has the memory complexity of $O(|\mathcal{S}|\frac{1}{D}\sum_{d=1}^{D} I_d)$. Please refer to [12] for more detailed complexity analysis.

IV. Experiment

A. Experimental Settings

1) Datasets: We conduct experiments on two real-world large volume, publicly available and de-identified datasets, MIMIC-III [26] and CMS [27], and a synthetic dataset with similar sparsity (see [12] for more detail). We follow the rules in [6] and select the top 500 diagnoses, procedures, and medications of the most frequently observed records to form the tensors with patient mode 34,272, 125,961, and 4000 for MIMIC-III, CMS, and Synthetic data, respectively.

2) Baselines: We consider the following centralized tensor factorization baselines: i) **GCP** [28] as the baseline of generalized tensor factorization; ii) **BrasCPD** [18] as the computation efficient tensor factorization baseline; iii) **Centralized** `CiderTF`, `CiderTF` with $K = 1$ and error-feedback.

We also implement the decentralized version SGD under the non-convex settings as the decentralized baselines, since there is no existing decentralized tensor factorization framework. i) **D-PSGD** [10], [29] as a pure decentralized SGD version; ii) **SPARQ-SGD** [24] as a decentralized communication-efficient stochastic gradient descent baseline; iii) **D-PSGDbras** can be considered as D-PSGD with block randomization.

3) Parameter Settings: Experiments are performed on two objective functions including Bernoulli-logit loss to fit the binary data (eq. 3) and Least Square Loss to fit the data with Gaussian distribution (eq. 2). We use a fixed learning rate $\gamma[t]$, which is determined through searching the grid of powers of 2. We follow the rules in [24] to set the triggering threshold $\lambda[t]$. The detailed parameter settings, additional experiment results (more datasets, ablation study, etc.) can be found in [12].

B. Result Analysis

We form a decentralized communication topology as a ring, and have a default of eight workers with data horizontally partitioned and distributed evenly across all the eight clients.

1) Comparison to the Baselines: From fig. 2, we have four major observations. I) `CiderTF` **converges to comparable losses as the centralized baselines.** These results empirically validate the convergence of `CiderTF`. II) `CiderTF` **has less communication cost without sacrificing the convergence.** `CiderTF` takes 99.99% less communication cost than D-PSGD, 75% less communication cost than SPARQ-SGD and 99.92% less than D-PSGDbras to achieve the same loss. III) `CiderTF` **is computationally efficient.** `CiderTF` is computationally efficient compared with GCP and D-PSGD (fig. 2) due to fiber sampling and block randomization. `CiderTF` is also slightly more efficient than BrasCPD thanks to the decentralized data distribution which helps parallelize the local tensor factorization. IV) **Nesterov's momentum can offer** `CiderTF_m` **faster convergence, leading to less overall communication cost**. `CiderTF_m` requires less epochs to converge (fig. 2), which in turn reduce the total communication bytes with little sacrifice of the accuracy.

2) Impact of Topology: We test `CiderTF` on ring topology and star topology with the same number of workers (fig. 1). From fig. 3, we observe that different topologies do not affect the convergence, which means that `CiderTF` can generalize to different kinds of communication topologies. Fig. 3 also illustrates that two topologies enjoy similar computation time due to the same number of workers, while star topology has less communication cost because the total degree of the star topology is less than the ring topology.

3) Scalability: Moreover, we test the scalability of `CiderTF`. By increasing the number of clients from $K = 8$ to $K = 16$ and $K = 32$, we observe linear scalability in the computation time (fig. 4 left) without sacrificing the accuracy. However, as the number of clients increases, the communication cost will increase accordingly (fig. 4 right). Therefore, there exists a computation-communication trade-off when increasing the number of clients involved in the decentralized tensor factorization framework.

C. Case Study on MIMIC-III

We conduct a case study on MIMIC-III to evaluate the extracted phenotypes from both quantitative and qualitative perspectives. From the quantitative aspect, we use the Factor Match Score (FMS) [30] to measure the similarity of the factor matrices of `CiderTF` with BrasCPD. FMS ranges from 0 to 1 with the best possible value of 1. Fig. 5 indicates that `CiderTF` achieves comparable FMS as the baselines with much less computation time and communication cost.

From the the qualitative perspective, we evaluate the quality of the phenotypes by patient subgroup identification ability. Following the precedent set in [5], we first identify the top three phenotypes according to the phenotype importance factor λ_r. We then group the patients by assigning each according to the largest value among the top 3 along the patient representation vector, and use tSNE to map the patient representation into two-dimensional space. Fig. II shows that `CiderTF` ($\tau = 8$) achieves comparable patient subgroup identification ability as the centralized baseline BrasCPD. While with the same communication cost, `CiderTF` achieves better clustered subgroups than the decentralized baseline SPARQ-SGD. In addition, the top 3 phenotypes extracted by `CiderTF` (table III) are clinically meaningful and interpretable as annotated by a pulmonary and critical care physician.

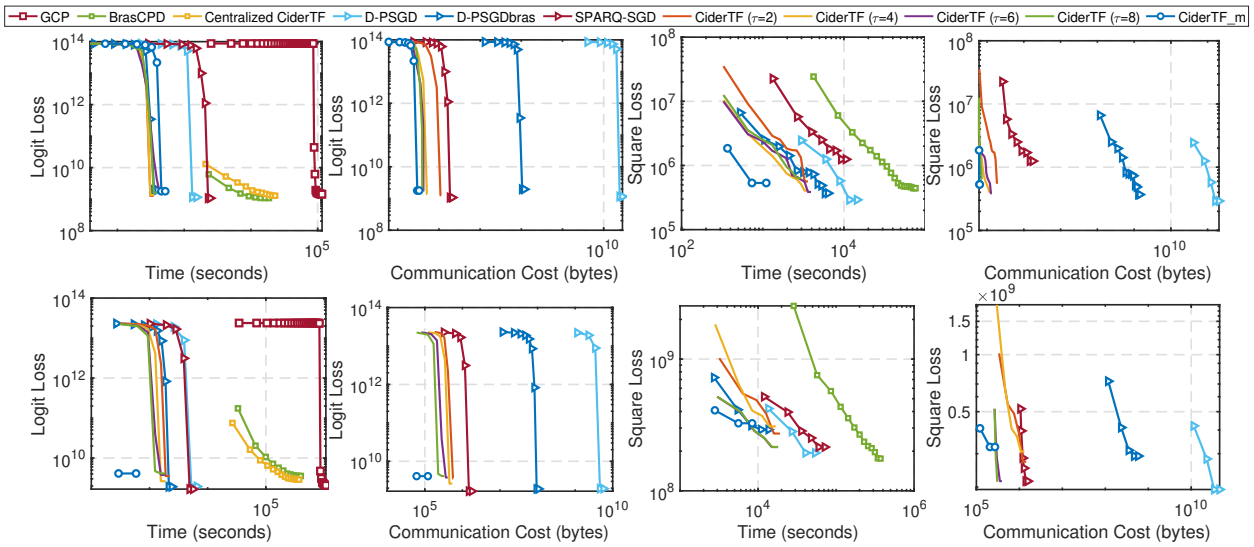


Fig. 2. Bernoulli-logit Loss (1-2 columns) and Least Square Loss (3-4 columns) with vs. time and communication for CMS (top) and MIMIC-III (bottom).

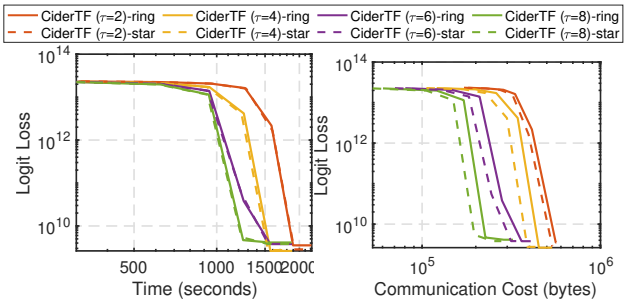


Fig. 3. Bernoulli-logit Loss for ring topology (solid lines) and star topology (dashed lines) with respect to time and communication for MIMIC-III data.

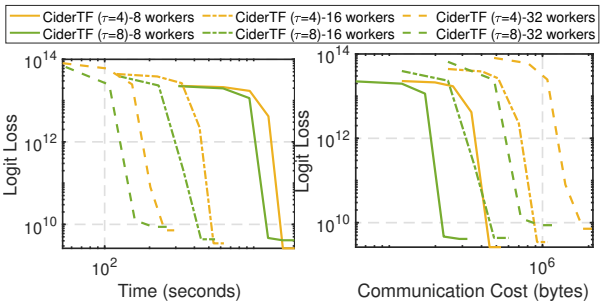


Fig. 4. Bernoulli-logit loss with respect to time and communication for MIMIC-III data with 8, 16, and 32 workers for local update rounds $\tau = 4, 8$.

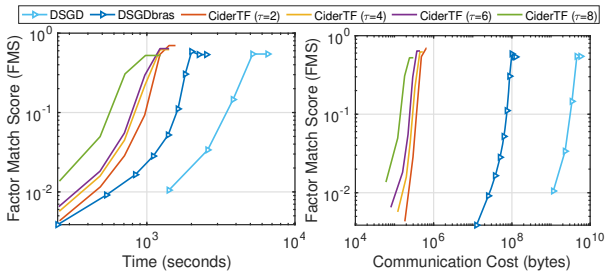
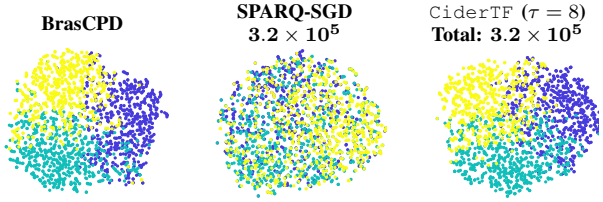


Fig. 5. Factor Match Scores (FMS) with respect to time and communication.

TABLE II

TSNE VISUALIZATION OF THE PATIENT SUBGROUP IDENTIFICATION WITH THE EXTRACTED PHENOTYPES. EACH POINT REPRESENTS A PATIENT WHICH IS COLORED ACCORDING TO THE HIGHEST-VALUED COORDINATE IN THE PATIENT REPRESENTATION VECTOR AMONG THE TOP 3 PHENOTYPES EXTRACTED BASED ON THE FACTOR WEIGHTS $\lambda_r = \|\mathbf{A}_{(1)}(:,r)\|_F \|\mathbf{A}_{(2)}(:,r)\|_F \cdots \|\mathbf{A}_{(D)}(:,r)\|_F$.



V. CONCLUSION

In this paper, we propose CiderTF, which is the first decentralized generalized tensor factorization framework. It employs aggressive communication reduction techniques and maintains low computational and memory complexity without sacrificing the accuracy. Experiments show that CiderTF preserves the quality of the extracted phenotypes and converges to similar points as the decentralized SGD baselines with theoretical guarantees. Future works include developing asynchronized communication and variance reduced techniques to the decentralized paradigm.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation under award IIS-#1838200, CNS-2124104 and CNS-1952192, National Institute of Health (NIH) under award number R01LM013323, K01LM012924 and R01GM118609, CTSA Award UL1TR002378.

REFERENCES

[1] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep patient: an unsupervised representation to predict the future of patients from the electronic health records," *Scientific reports*, vol. 6, no. 1, pp. 1–10, 2016.

[2] N. G. Weiskopf and C. Weng, "Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research," *JAMIA*, vol. 20, no. 1, pp. 144–151, 2013.

[3] J. C. Ho, J. Ghosh, and J. Sun, "Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization," in *Proceedings of the 20th ACM SIGKDD*, 2014, pp. 115–124.

[4] Y. Wang, R. Chen, J. Ghosh, J. C. Denny, A. Kho, Y. Chen, B. A. Malin, and J. Sun, "Rubik: Knowledge guided tensor factorization and completion for health data analytics," in *Proceedings of the 21th ACM SIGKDD*, 2015.

[5] I. Perros, E. E. Papalexakis, F. Wang, R. Vuduc, E. Searles, M. Thompson, and J. Sun, "Spartan: Scalable parafac2 for large & sparse data," in *Proceedings of the 23rd ACM SIGKDD*, 2017, pp. 375–384.

[6] Y. Kim, J. Sun, H. Yu, and X. Jiang, "Federated tensor factorization for computational phenotyping," in *Proceedings of the 23rd ACM SIGKDD*, 2017.

[7] J. Ma, Q. Zhang, J. Lou, J. C. Ho, L. Xiong, and X. Jiang, "Privacy-preserving tensor factorization for collaborative health data analysis," in *Proceedings of the 28th ACM CIKM*, 2019, pp. 1291–1300.

[8] J. Ma, Q. Zhang, J. Lou, L. Xiong, and J. C. Ho, "Communication efficient federated generalized tensor factorization for collaborative health data analytics," in *Proceedings of the Web Conference 2021*, 2021, pp. 171–182.

[9] J. Li, Y. Shao, M. Ding, C. Ma, K. Wei, Z. Han, and H. V. Poor, "Blockchain assisted decentralized federated learning (blade-fl) with lazy clients," *arXiv preprint arXiv:2012.02044*, 2020.

[10] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *arXiv preprint arXiv:1705.09056*, 2017.

[11] A. Vulimiri, C. Curino, P. B. Godfrey, T. Jungblut, J. Padhye, and G. Varghese, "Global analytics in the face of bandwidth and regulatory constraints," in *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, 2015, pp. 323–336.

[12] J. Ma, Q. Zhang, J. Lou, L. Xiong, S. Bhavani, and J. C. Ho, "Communication efficient tensor factorization for decentralized healthcare networks," *arXiv preprint arXiv:2109.01718*, 2021.

[13] D. Hong, T. G. Kolda, and J. A. Duersch, "Generalized canonical polyadic tensor decomposition," *arXiv preprint arXiv:1808.07452*, 2018.

[14] J. Zeng, T. T.-K. Lau, S. Lin, and Y. Yao, "Global convergence of block coordinate descent in deep learning," in *ICML*, 2019, pp. 7313–7323.

[15] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication," *arXiv preprint arXiv:1909.05350*, 2019.

[16] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," in *NeurIPS*, 2018, pp. 4447–4458.

[17] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[18] X. Fu, S. Ibrahim, H.-T. Wai, C. Gao, and K. Huang, "Block-randomized stochastic proximal gradient for low-rank tensor factorization," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2170–2185, 2020.

[19] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[20] S. U. Stich, "Local sgd converges fast and communicates little," in *ICLR*, 2018.

[21] T. Lin, S. U. Stich, K. K. Patel, and M. Jaggi, "Don't use large mini-batches, use local sgd," *arXiv preprint arXiv:1808.07217*, 2018.

[22] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-sgd: Distributed sgd with quantization, sparsification, and local computations," in *NeurIPS*, 2019.

[23] W. Du, X. Yi, J. George, K. H. Johansson, and T. Yang, "Distributed optimization with dynamic event-triggered mechanisms," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 969–974.

[24] N. Singh, D. Data, J. George, and S. Diggavi, "Sparq-sgd: Event-triggered and compressed communication in decentralized optimization," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 3449–3456.

[25] C. Battaglino, G. Ballard, and T. G. Kolda, "A practical randomized cp tensor decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 39, no. 2, pp. 876–901, 2018.

[26] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[27] https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/DE_Syn_PUF.

[28] T. G. Kolda and D. Hong, "Stochastic gradients for large-scale tensor decomposition," *arXiv preprint arXiv:1906.01687*, 2019.

[29] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *ICML*. PMLR, 2020, pp. 5381–5393.

[30] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.

TABLE III
PHENPTYPES EXTRACTED BY `CiderTF` ($\tau = 8$) ON MIMIC-III DATA. DX, PX, AND MED INDICATE DIAGNOSES, PROCEDURES, AND MEDICATION.

P1: Acute myocardial infarction	
Dx	Other and unspecified angina pectoris Coronary atherosclerosis of autologous vein bypass graft Old myocardial infarction
Px	(Aorto)coronary bypass of two coronary arteries (Aorto)coronary bypass of three coronary arteries Implant of pulsation balloon
Med	Diltiazem Hydrochloride Extended-Release Metoprolol succinate, Rosuvastatin Calcium Valsartan/hydrochlorothiazide, Losartan Potassium
P2: Respiratory failure	
Dx	Acute respiratory failure, Hypoxemia, Contusion of lung without mention of open wound into thorax Disruption of internal operation (surgical) wound
Px	Non-invasive mechanical ventilation Continuous invasive mechanical ventilation for less than 96 consecutive hours
Med	Dextrose, Albuminar-25, Plasmanate
P3: Intracranial hemorrhage or cerebral infarction	
Dx	Pure hypercholesterolemia, Subdural hemorrhage Cerebral artery occlusion
Px	Injection or infusion of thrombolytic agent Control of hemorrhage
Med	Ticagrelor, Atorvastatin Calcium