Federated-PCA on Vertical-Partitioned Data

Yiu-ming Cheung [1] and Feng Yu [1]

[1]Affiliation not available

October 30, 2023

Abstract

In the cross-silo federated learning setting, one kind of data partition according to features, which is so-called vertical federated learning (i.e. feature-wise federated learning) (Yang et al. 2019), is to apply to multiple datasets that share the same sample ID space but different feature spaces. Simultaneously, the image dataset can also be partitioned according to labels. To improve the model performance of the isolated parties based on feature-wise (i.e. label-wise) results, the most effective method is to federate the model results of the isolated parties together. However, it is a non-trivial task to allow the participating parties to share the model results without violating the data privacy of the parties. In this paper, within the framework of principal component analysis (PCA), we propose a Federated-PCA machine learning approach, in which the PCA method is used to reduce the dimensionality of sample data for all parties and extract the principal component feature information to improve the efficiency of subsequent training work. This process will not reveal the original data information of each party. The federal system can help each side build a common profit strategy. Under this federal mechanism, the identity and status of each party are the same. By comparing the federated results of the isolated parties and the result of the unseparated party through multiple sets of comparative experiments, we find that the experimental results of these two settings are close, and the proposed method can effectively improve the training model performance of most participating parties.

Federated-PCA on Vertical-Partitioned Data

Yiu-ming Cheung and Feng Yu

Department of Computer Science, Hong Kong Baptist University
Hong Kong SAR, China
ymc@comp.hkbu.edu.hk, 18430554@life.hkbu.edu.hk

Abstract. In the cross-silo federated learning setting, one kind of data partition according to features, which is so-called vertical federated learning (i.e. feature-wise federated learning)[23], is to apply to multiple datasets that share the same sample ID space but different feature spaces. Simultaneously, the image dataset can also be partitioned according to labels. To improve the model performance of the isolated parties based on feature-wise (i.e. label-wise) results, the most effective method is to federate the model results of the isolated parties together. However, it is a non-trivial task to allow the participating parties to share the model results without violating the data privacy of the parties. In this paper, within the framework of principal component analysis (PCA), we propose a Federated-PCA machine learning approach, in which the PCA method is used to reduce the dimensionality of sample data for all parties and extract the principal component feature information to improve the efficiency of subsequent training work. This process will not reveal the original data information of each party. The federal system can help each side build a common profit strategy. Under this federal mechanism, the identity and status of each party are the same. By comparing the federated results of the isolated parties and the result of the unseparated party through multiple sets of comparative experiments, we find that the experimental results of these two settings are close, and the proposed method can effectively improve the training model performance of most participating parties.

Keywords: Vertical federated learning, Feature-wise(i.e. Label-wise), Federated-PCA

1 Introduction

Federated learning (FL) was first proposed by Google in 2016 [12], whose main idea is to build machine-learning models based on datasets that are distributed across multiple devices while preventing data leakage and learning a shared model by aggregating locally computed updates via a central coordinating server. There is growing interest in applying FL to other applications, including some applications that may involve only a few reliable clients or multiple organizations working together to train models. In fact, FL is a machine learning setting, where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each clients

raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective [11].

In general, there are two FL settings, i.e. *cross-device* and *cross-silo*. In the former setting, the data is assumed to be partitioned by samples, i.e. horizontal FL [23]. By contrast, in the cross-silo, in addition to partitioning by samples, partitioning by features is of practical relevance [11], which is also called as vertical FL (VFL) [23] or feature-wise FL interchangeably. In particular, for the image datasets, it can be also partitioned by labels. It applies to the cases that two data sets share the same sample ID space but differ in feature space.

The cross-silo setting can be relevant where several companies or organizations share incentives to train a model based on all of their data but cannot share their data directly. This could be due to constraints imposed by confidentiality or due to legal constraints even within a single company when they cannot centralize their data between different geographical regions. For example, let us consider two different companies in the same city, one is a bank, and the other is an e-commerce company. The sets of their users are likely to contain most of the residents of the area, thus the intersection of their user space is large. However, since the bank records the users revenue and expenditure behavior and credit rating, and the e-commerce company retains the users browsing and purchasing history, their feature spaces are very different. Supposing both parties need to have a prediction model for product purchases based on user and product information [23], this is a typical VFL. In addition, there are also different medical centers that have large user intersection space and hold different types of personal medical image information of users, and shopping malls (online or offline) that have large user intersection space but retain different shopping records for users are worth considering. In this paper, we will focus on the cross-silo setting with data partitioned by features (i.e. vertical-partitioned data), and under which we attempt to explore the learning of principal component analysis (PCA) as PCA is a fundamental and very useful machine learning model. The basic idea of PCA is to reduce the dimensionality of a data set, in which there are a large number of interrelated variables while retaining as much as possible of the variation presented in the data set. This reduction is achieved by transforming the original variables to a new set of variables, i.e. the principal components, which are uncorrelated and ordered so that the first few retain most of the data variations [16].

This paper considers that all parties can use the model parameters of other parties to improve the model performance without revealing their raw dataset. Focusing on the vertical-partitioned data, we will propose an approach namely, Federated-PCA, which can significantly improve the models of all parties and the final results of all labels in the image dataset. Our main contributions are: (1) Proposing a federated PCA learning method and perform feature-wise(i.e. label-wise) feature extraction and data compression based on vertically partitioned data, and then obtain the results of joint PCA of all parties, (2) Building two

model protocols: Central collaborative server and Fully decentralized (i.e. peer-to-peer) model framework.

The rest of the paper is organized as follows. Section 2 makes an overview of related work. Section 3 introduces the framework of the proposed Federated-PCA machine learning. Section 4 shows the experimental results to demonstrate the effectiveness of the proposed approach. Finally, we give a concluding remark in Section 5.

2 Related Work

Most of the existing work is based on cross-device federated learning. In the cross-device setting, the data is assumed to be partitioned by examples. Different from cross-device federated learning, in the cross-silo setting, the data is assumed to be partitioned by samples and features. In particular, for the case of cross-silo FL with data partitioned by features, it may or may not involve a central server as a neutral party, and clients would exchange specific intermediate results rather than model parameters to assist in calculating the other parties' gradients [21]. In this setting, the application of techniques such as Secure Multi-party Computation or Homomorphic Encryption has been presented to limit the amount of information other participants can infer from observing the training process. The downside of this approach is that the training algorithm is typically dependent on the type of machine learning objective being pursued. Currently, the existing algorithms include trees [5], linear and logistic regression [23, 10], and neural networks [14].

Privacy-preserving is one of the most important problems in federated learning. At present, the solutions proposed in privacy-preserving learning are mainly based on the following: Secure Multi-party Computation (SMC), Differential Privacy and Homomorphic Encryption. The privacy definition of federated learning can be classified into two categories: global privacy and local privacy [13]. Global privacy requires that the model updates generated at each round are private to all untrusted third parties other than the central server, while local privacy further requires that the updates are also private to the server. Privacy-preserving machine learning algorithms have been proposed for vertically partitioned data, including Cooperative Statistical Analysis [7], association rule mining [20], secure linear regression [20, 18], classification [7] and gradient descent [21]. Recently, [10, 17] proposed a vertical federated learning scheme to train a privacy-preserving logistic regression model. The authors studied the effect of entity resolution on the learning performance and applied Taylor approximation to the loss and gradient functions so that homomorphic encryption can be adopted for privacy-preserving computations. The existing solutions in privacy-preserving learning are mainly based on *Secure Multi-party Computation* (SMC), and *Differential Privacy and Homomorphic Encryption*.

Current works that aim to improve the privacy of federated learning typically build upon previous classical cryptographic protocols such as SMC [4, 9] and differential privacy [1, 3, 8, 15]. The protocol of SMC is introduced to pro-

tect individual model updates[4]. The central server is not able to see any local updates but can still observe the exact aggregated results at each round. SMC is a lossless method and can retain the original accuracy with a very high privacy guarantee. However, the resulting method incurs significant extra communication cost. Other works [8, 15] apply differential privacy to federated learning and offer global differential privacy. These approaches have several hyperparameters, which should be carefully chosen because of their impact on communication and accuracy, although [19] has presented the adaptive gradient clipping strategies to help alleviate this issue. Besides, differential privacy can be combined with the model compression techniques to reduce communication and obtain privacy benefits simultaneously [1].

In contrast with the existing work, we focus more on the principal component extraction and privacy protection through Federated-PCA learning. Our approach is to share intermediate parameters only without directly sharing the original party's original data. The detailed privacy-preserving PCA framework will be presented in the next section.

3 The Proposed PCA on Privacy-Preserving Vertical-Partitioned Data

3.1 Proposed Method:Federated-PCA

FL aims at achieving a common profit strategy for all parties by sharing parameters. To this end, one feasible idea is to average the parameters shared by all parties. According to this idea, we first try to directly average the eigenvector shared by all parties for processing. A plausible reason is that this method can improve the overall performance of the model. Alternatively, after considering the relationship between eigenvalue and eigenvector, another idea we propose is that the party with the largest eigenvalue weight among all participants should have more direct effect on the final federated results.

Based on our idea stated above, we can achieve the FL along with two ways. The first way is to add a third-party trust coordination agent that can help all parties complete the model training work more efficiently. This third-party coordinator cannot access the data of each party, which is important for the privacy protection of all parties. On the other way, when the number of participating parties is small, as in the case of two-party or the third-party is not trusty, it is feasible to consider using a fully decentralized PCA learning method to build a co-build model provided that all parties involved are honest. Accordingly, let us consider setting up two Federated-PCA model protocols: a central collaborative server (i.e. Mode 1) and a fully decentralized (peer to peer) learning model (i.e. Mode 2) for these two ways, respectively.

Mode 1 : All parties share parameters with the help of the central collaborative server. In our setting, this third-party coordinator can be trusted, it can only accept intermediate parameters sent by each party and cannot get the original

data of any party. After adjusting the new parameters, this federated-parameter is returned to all parties. Then, each party calculates locally based on this result. The framework of this model is shown in Figure 1. The step of this method is as follows:

- **Step 1:** *Local training:* All parties involved perform PCA calculations locally, first obtain the covariance matrix $\mathbf{S}$, and based on this $\mathbf{S}$, find the corresponding eigenvector λ_i and eigenvalue $\mathbf{v}_i$, $i \in \{1, \cdots k\}$ and select the largest $\mathbf{k}$ eigenvector and eigenvalue values;
- **Step 2:** *Model integration:* The central server calculates the weight $\mathbf{W}_i$ occupied by each party in the federated model by receiving the eigenvalue $(\lambda_i)_k$, $i \in \{1, \cdots k\}$ of each party, and then combines the received eigenvector $(\mathbf{v}_i)_k$, $i \in \{1, \cdots k\}$ values to derive a joint feature vector $(\mathbf{V}_k)$;
- **Step 3:** *Parameters broadcasting:* The central collaborative server broadcasts the aggregated parameters $(\mathbf{V}_k)$ to the N parities;
- **Step 4:** *Local model updating:* Each party updates each eigenvector based on the returned joint feature vector $(\mathbf{V}_k)$, and obtains the final extracted $\mathbf{k}$ final data.

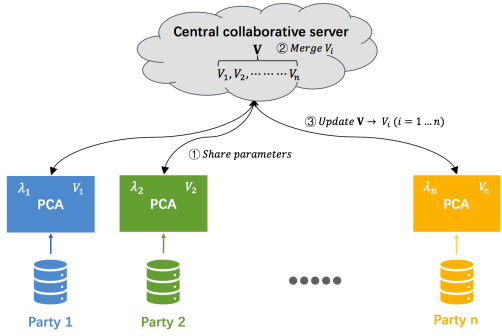


Fig. 1. The framework of Federated-PCA with Central Collaborative Server

The advantage of this model is that it is relatively efficient, and the disadvantage is that it relies on the help of third-party agencies.

Mode 2 : All parties only share principal component parameters with each other, and compute the results locally. The framework of this model is shown in Figure 2. The step of this method is as follows:

- **Step 1:** *Local training:* All parties involved perform PCA calculations locally, first obtain the covariance matrix $\mathbf{S}$, and based on this $\mathbf{S}$, find the corresponding eigenvector λ_i and eigenvalue $\mathbf{v}_i$, $i \in \{1, \cdots k\}$ and select the largest $\mathbf{k}$ eigenvector and eigenvalue values;

– **Step 2:** *Parameters sharing:* Intermediate results parameters $(\lambda_i)_k$ and $(\mathbf{v}_i)_k$ of each participant are shared;
– **Step 3:** *Local model updating:* Each party obtains the extracted **k** final data based on calculating the weight $\mathbf{W}_i$ which occupied by all parties and updates locally.

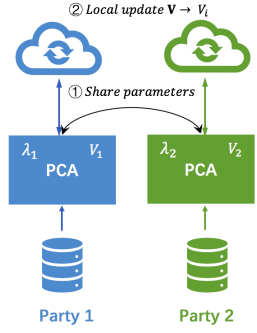


Fig. 2. The framework of Fully decentralized (*peer-to-peer*) PCA learning

This model circumvents the need for third-party agencies to help participants further reduce some external risks, especially for two-party scenarios. However, the principal components of all parties need to be calculated independently, and the computing efficiency will be greatly affected.

3.2 Algorithm

We propose a standardized representation of Federated-PCA learning. Suppose we have a data matrix $\mathbf{S} = [\mathbf{f}_1|\mathbf{f}_2|\cdots|\mathbf{f}_n]$, which contains n data parties, each party has p variables (i.e. features, labels) and q samples (i.e. users), $\mathbf{f}_i \in \mathbb{X}^{p\times n}$, where we assume that the number of samples q is the same and features $\mathbf{X}_p$ for each party are different. Each party has its own database $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p\} \in \mathfrak{R}^{q\times p}$, in particular, for image dataset, $\mathbf{x}_i \in \mathbf{R}^{m\times m\times q}$,where m is the pixel size. The final extracted principal component result is

$$\mathbf{Z}_k = \mathbf{v}_k^T\mathbf{X} = \sum_i^q \mathbf{v}_k\mathbf{x}_{ij}, j = 1, 2\cdots p$$

where the vector $(\mathbf{v}_i)_k = (\mathbf{v}_1, \mathbf{v}_2\cdots\mathbf{v}_q)_k$, $\mathbf{x}_j = (\mathbf{x}_{1j}, \mathbf{x}_{2j}, \cdots\mathbf{x}_{qj})$, **k** is the number of largest principal component representation selected. **S** is the covariance matrix, $\mathbf{S}^{q,i} = \mathbf{X}\mathbf{X}^\mathbf{T}$, $i = 1, 2\cdots N$, $j = 1, 2\cdots q, where N$ is the number of parties. Let $(\lambda_i)_k = (\lambda_{1k}, \lambda_{2k}\cdots\lambda_{qk})$ be the selected eigenvalues by each party, we have:

$$\mathbf{W}_i = \frac{\lambda_i}{\sum_{i=1}^n \lambda_i}$$

$$\mathbf{V} = \mathbf{W}_1\mathbf{V}_1 + \mathbf{W}_2\mathbf{V}_2 + \cdots \mathbf{W}_n\mathbf{V}_n$$

where $\mathbf{W}_i$ is the weight relationship between the eigenvector of each party and the shared feature vector, and $\mathbf{V}$ is the shared projection feature vector.

To achieve Mode 1 and Mode 2 stated in Section 3.1, the corresponding algorithms are given in Algorithm 1 and 2, respectively.

Algorithm 1: Federated-PCA learning with central collaborative server

⇒ **Run on party *i* server**
Input: Data $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p\} \in \Re^{q\times p}$ belongs to party i
Output: Principal eigenvalues λ_i and eigenvectors $\mathbf{V}_i$
Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p]$, $\overline{\mathbf{x}} = 0$
while *each feature value* $\mathbf{x}_i, i = \{1, 2\cdots p\}$ **do**
  $\mathbf{S}^{q,i} \leftarrow \mathbf{f}^{q,i}$,
  $[\lambda^{q,i}, \mathbf{v}^{q,i}, \mathbf{k}] \leftarrow PCA(\mathbf{S}^{q,i})$,
  Each party send $(\lambda^{q,i}, \mathbf{v}^{q,i})_k$ to the central collaborative server.
end
⇒ **Run on central collaborative server**
Input: Quantity of parties $p \in \{1, 2\cdots N\}$
Output: Value of each $\mathbf{W}_i$, $n \in \{1, 2\cdots N\}$ and shared feature vector $\mathbf{V}$
Receive λ_i, $\mathbf{V}_i$ from n parties
for $i = 1$ *to* n **do**
  $\mathbf{W}_i \leftarrow \lambda^{i,n}$,
  $\mathbf{V} \leftarrow Merge(\mathbf{W}^{i,n}, \mathbf{v}^{i,n})$
  Broadcast shared feature vector $\mathbf{V}$ to all parties.
end
⇒ **Run on central collaborative server**
Update eigenvector $\mathbf{V}_i \leftarrow \mathbf{V}$
return final results.

Algorithm 2: Fully decentralized Federated-PCA learning

⇒ **Run on party** i **server**
Input: Data $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p\} \in \Re^{q \times p}$ belongs to party i, quantity of parties $n \in \{1, 2 \cdots N\}$
Output: Final PCA learning results locally
Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p]$, $\overline{\mathbf{x}} = 0$
while *each feature value* $\mathbf{x}_i, i = \{1, 2 \cdots p\}$ **do**
  $\mathbf{S}^{q,i} \leftarrow \mathbf{f}^{q,i}$,
  $[\lambda^{q,i}, \mathbf{v}^{q,i}, \mathbf{k}] \leftarrow PCA(\mathbf{S}^{q,i})$,
  Send $(\lambda^{q,i}, \mathbf{v}^{q,i})_k$ to other parties.
  for $n = 1$ *to* N **do**
    $\mathbf{W}_i \leftarrow \lambda^{i,n}$,
    $\mathbf{V} \leftarrow Merge(\mathbf{W}^{i,n}, \mathbf{v}^{i,n})$
  end
  Update eigenvector $\mathbf{v}^{i,n} \leftarrow \mathbf{V}$
end
return final results.

4 Experiments

This section will empirically evaluate our proposed method.

4.1 Datasets

In our experiments, first, we used structured datasets from different domains. We split these datasets differently based on feature-wise. We used a total of 10 datasets for comparative experimental testing. The information on ten data sets are given in Table 1 and then we used image dataset Fashion-MNIST [22] on Table 2. We resplit this data set according to different labels, divided into training and test sets, and performed comparison experiments on the unsplit data set and the split data set.

4.2 Comparative Results

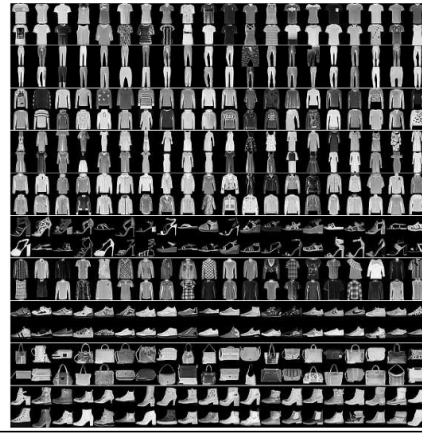
In our setting, each party will obtain a set of eigenvectors and eigenvalues, we choose the top k PCA that basically cover 90% of the explained variance ratios and the choice of the number of principal components is determined by the party with the largest number of principal components demand. In particular, after performing PCA in a label-wise way on the Fashion-MNIST dataset [22], the explained variance ratio almost covers 70% when k = 1.

We conducted a series of comparative experiments to perform split experiments on the data set in various forms. In total, we set up 4 comparative experimental groups: undivided-party, isolated PCA party, federated-PCA, federated-avg PCA. By comparing with the four situations on the different split-datasets,

Table 1. Information on ten data sets

Dataset	Parties	Features	Samples
Boston	2	[8, 7]	506
	3	[5, 4, 4]	506
	4	[4, 3, 3, 3]	506
GlaucomaM	2	[15, 47]	196
Ozone	2	[5, 5]	203
Diabetes	2	[4, 4]	768
Seeds	2	[4, 3]	210
Vehicle	3	[7, 7, 4]	846
College	3	[4, 6, 7]	777
Musk1	2	[28, 27]	476
Musk	3	[55, 55, 56]	476
Vowel	2	[4, 4]	990
Glass	2	[4, 3]	214

Table 2. Labels and example images in Fashion-MNIST dataset.

Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

we finally get the results of each group of experiments. A snapshot of the principal component eigenvector and the final results on the 10 datasets is shown in Figure 3, 4 and 5.

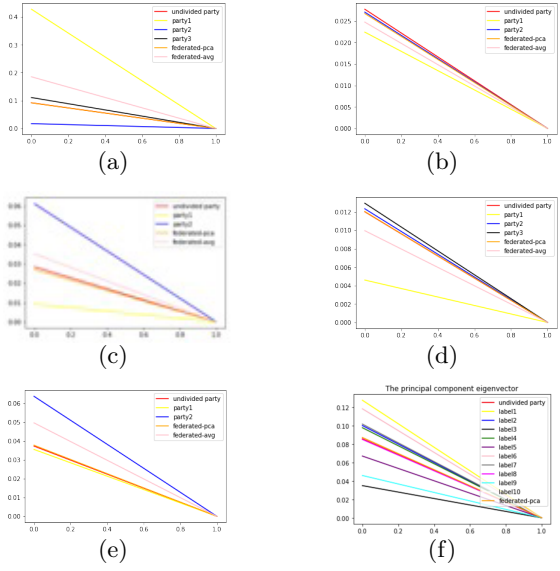


Fig. 3. The visualization of the comparative principal component eigenvector on datasets (a)Boston Housing (b)Diabetes (c)Seeds (d)Vehicles (e)Vowels (f)Fashion-MNIST

The range of bias rates between the final results of unsplit data and isolated PCA final results, federated-PCA final results are summarized in Table 3.

Besides, the final experimental improvement of the College dataset is not very obvious. Therefore, we reanalyzed the data set and checked the correlation coefficients between different features in all datasets. We found that the correlation coefficients of the first few features were too large (close to 1), so we screened these features, preprocessed this dataset, and found that the results were significantly improved after the experiment again. The comparison results are given in Figure 6.

According to the experimental results, the proposed approach has significantly helped improve the performance of the final results of almost all datasets. Compared with the higher bias rate of normal results, the bias rate range of Federated-PCA results is basically within 10% which proves the validity and value of this method. Similarly, the image dataset is also performed very well. Figure 4 clearly shows that the final image after Federated-PCA is almost the same as the final image after unsplit label image data. To further verify that the final label-wise image result extracted more features, we performed a logistic regression classification test. In contrast to the settings where each party uses the

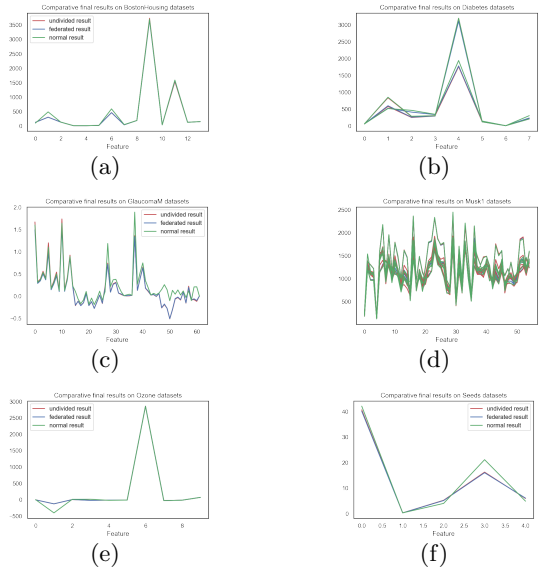


Fig. 4. The final results of comparative experiment on datasets (a)Boston Housing (b)Diabetes (c) GlaucomaM (d)Musk (e)Ozone (f)Seeds

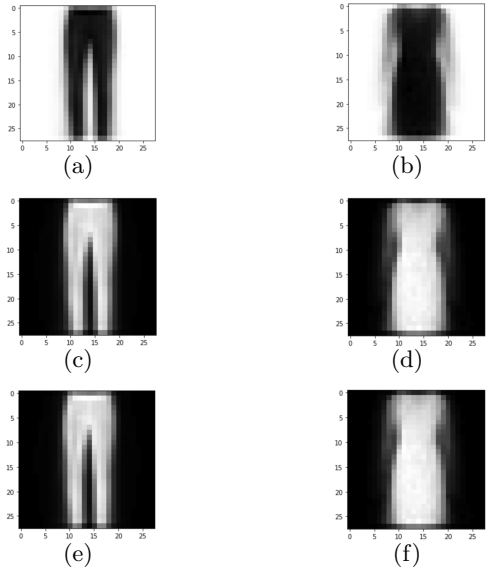


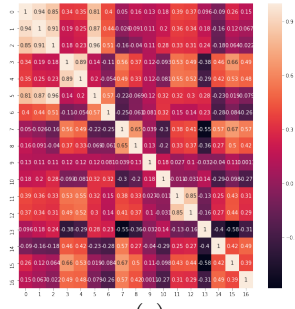
Fig. 5. The final results of comparative experiment on Fashion-MNIST datasets (a)The isolated PCA projected results on label: Trouser (b)The isolated PCA projected results on label: Dress (c)The federated-PCA projected results on label: Trouser(d)The federated-PCA projected results on label: Dress (e)The unsplit data projected results on label: Trouser (f)The unsplit data projected results on label: Dress

Table 3. Information of final result data deviation range on the ten datasets

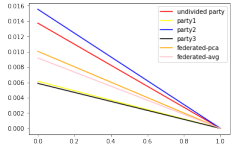
Dataset	Isolated PCA result bias rate range	Federated-PCA result bias rate range
Boston Housing	(0.193%,39.521%)	(0.013%,1.919%)
GlaucomaM	(0.288%,31.065%)	(0.006%,4.362%)
Ozone	(0.161%,79.352%)	(0.092%,8.189%)
Diabetes	(2.277%,66.445%)	(0.490%,4.743%)
Seeds	(1.623%,30.039%)	(2.564%,6.348%)
Vehicle	(0.115%,19.814%)	(0.064%,1.293%)
College	(0.598%,44.097%)	(0.985%,13.570%)
Musk	(0.061%,6.590%)	(0.042%,3.081%)
Vowel	(0.651%,36.862%)	(0.932%,1.879%)
Glass	(0.817%,5.172%)	(0.179%,0.495%)

Table 4. The classification accuracy of logistic regression on Fashion-MNIST dataset

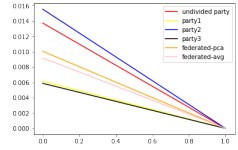
	Accuracy-full test	Accuracy-pca test
Original dataset	0.768	1
Isolated PCA result	0.336	0.4
Federated-PCA result	0.724	1
Unsplit data PCA result	0.724	1



(a)



(b)



(c)

Fig. 6. Comparison results on College dataset (a) Correlation coefficient results between different features (b)Before result (c)After result

standard PCA method independently, Table 4 shows the classification accuracy rate is greatly improved to approximately 100% after using the federated-PCA method, which will help each party further perform local training tasks.

4.3 Concluding Remarks

In this paper, based on vertical data, our proposed Federated PCA can improve the models of almost all parties. After comparative experiments, it is clear that the model results between the federated parties and the unsplit-party are close. Therefore, this method can help some data stakeholders to solve problems such as large amounts of data and shared privacy. In reality, there is a non-linear relationship between various data, and the standard PCA method is difficult to find a good representative direction, especially for the image data, most of them show a non-linear relationship. In order to further serve more complex and more dimensional image datasets, such as medical datasets, as mentioned in [11], obtaining better models by extracting features of different categories is very valuable for medical centers and hospitals. For example, a patient may go to one medical clinic for a pathology test and go to another for radiology picture archiving. Under this situation, the features of one sample are partitioned over two clinics regulated by HIPAA [2]. In the future, we will perform Federated-KPCA learning on different kinds of non-linear data. As the privacy protection of patient data is one of our first considerations, so our work will continue to share the model parameter between parties based on the privacy protection system.

During the experiment, the effect is not obvious for some structured datasets, we consider that the correlation between different features should also be the main reason for the poor final result. Later experiments will add correlation analysis in the data set preprocessing stage. Besides, considering a large number of missing data values for a certain feature in the data set, we consider whether we can directly delete this column first according to the eigenvalue calculation method proposed in [6]. The original eigenvalues are calculated from the remaining sub-matrices. This method may help us estimate the eigenvalues of missing features based on the values of other existing features.

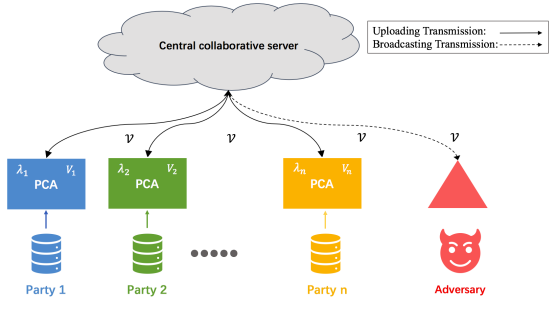


Fig. 7. The framework of the threat Federated-PCA model

The future is accompanied by many challenges, such as external adversaries are targeting at the central collaborative server and parties, the framework of the threat model is shown in Figure 7. Another future work is to improve the verification mechanism of each party, such as establishing a screening mechanism. As a result, we will make sure each party is a safe and effective participant. Besides, we can adjust the model parameters based on the results provided by each party to adjust the model parameters to further improve the performance of the model.

References

1. Agarwal, N., Suresh, A.T., Yu, F.X.X., Kumar, S., McMahan, B.: cpsgd: Communication-efficient and differentially-private distributed sgd. In: Advances in Neural Information Processing Systems. pp. 7564–7575 (2018)
2. Annas, G.J., et al.: Hipaa regulations-a new era of medical-record privacy? New England Journal of Medicine **348**(15), 1486–1490 (2003)
3. Bhowmick, A., Duchi, J., Freudiger, J., Kapoor, G., Rogers, R.: Protection against reconstruction and its applications in private federated learning. arXiv preprint arXiv:1812.00984 (2018)
4. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 1175–1191 (2017)
5. Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Yang, Q.: Secureboost: A lossless federated learning framework. arXiv preprint arXiv:1901.08755 (2019)
6. Denton, P.B., Parke, S.J., Tao, T., Zhang, X.: Eigenvectors from eigenvalues: a survey of a basic identity in linear algebra. arXiv preprint arXiv:1908.03795 (2019)
7. Du, W., Atallah, M.J.: Privacy-preserving cooperative statistical analysis. In: Seventeenth Annual Computer Security Applications Conference. pp. 102–110. IEEE (2001)
8. Geyer, R.C., Klein, T., Nabi, M.: Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557 (2017)
9. Ghazi, B., Pagh, R., Velingker, A.: Scalable and differentially private distributed aggregation in the shuffled model. arXiv preprint arXiv:1906.08320 (2019)
10. Hardy, S., Henecka, W., Ivey-Law, H., Nock, R., Patrini, G., Smith, G., Thorne, B.: Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. arXiv preprint arXiv:1711.10677 (2017)
11. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977 (2019)
12. Konečnỳ, J., McMahan, H.B., Ramage, D., Richtárik, P.: Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527 (2016)
13. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. arXiv preprint arXiv:1908.07873 (2019)
14. Liu, Y., Chen, T., Yang, Q.: Secure federated transfer learning. arXiv preprint arXiv:1812.03337 (2018)
15. McMahan, H.B., Ramage, D., Talwar, K., Zhang, L.: Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963 (2017)
16. Mishra, S.P., Sarkar, U., Taraphder, S., Datta, S., Swain, D.P., Saikhom, R., Panda, S., Laishram, M.: Multivariate statistical data analysis-principal component analysis (pca). Int J Liv Res. 2017c **7**(5), 60–78 (2017)
17. Nock, R., Hardy, S., Henecka, W., Ivey-Law, H., Patrini, G., Smith, G., Thorne, B.: Entity resolution and federated learning get a federated resolution. arXiv preprint arXiv:1803.04035 (2018)
18. Sanil, A.P., Karr, A.F., Lin, X., Reiter, J.P.: Privacy preserving regression modelling via distributed computation. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 677–682 (2004)

19. Thakkar, O., Andrew, G., McMahan, H.B.: Differentially private learning with adaptive clipping. arXiv preprint arXiv:1905.03871 (2019)
20. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 639–644 (2002)
21. Wan, L., Ng, W.K., Han, S., Lee, V.C.: Privacy-preservation for gradient descent methods. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 775–783 (2007)
22. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
23. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST) **10**(2), 1–19 (2019)