Computer Science and Information Systems 17(3):819–834 https://doi.org/10.2298/CSIS190923022O

A Homomorphic-encryption-based Vertical Federated Learning Scheme for Rick Management

Wei Ou[1], Jianhuan Zeng[2], Zijun Guo[1], Wanqin Yan[1], Dingwan Liu[1], and Stelios Fuentes[3]

[1] Department of Electronic and Information Engineering
Hunan University of Science and Engineering, Yongzhou, China
{ouwei1978430, yanwanqinqin, liudinwan}@163.com, GuoZijun0831@gmail.com
[2] Artificial Intelligence Research Center
Qianhai Institute for Innovation Research, Shenzhen, China
zengjianhuan@foxmail.com
[3] Leicester University, UK
stelios.fuentes@gmx.co.uk

Abstract. With continuous improvements of computing power, great progresses in algorithms and massive growth of data, artificial intelligence technologies have entered the third rapid development era. However, With the great improvements in artificial intelligence and the arrival of the era of big data, contradictions between data sharing and user data privacy have become increasingly prominent. Federated learning is a technology that can ensure the user privacy and train a better model from different data providers. In this paper, we design a vertical federated learning system for the for Bayesian machine learning with the homomorphic encryption. During the training progress, raw data are leaving locally, and encrypted model information is exchanged. The model trained by this system is comparable (up to 90%) to those models trained by a single union server under the consideration of privacy. This system can be widely used in risk control, medical, financial, education and other fields. It is of great significance to solve data islands problem and protect users' privacy.

Keywords: Data Security, Privacy Preservation, Federated Learning, EM Algorithm, Homomorphic Encryption.

1. Introduction

With the arrival of the era of big data, more and more industries are paying attention to artificial intelligence technology. The ability of artificial intelligence to quickly process large amounts of data and the ability to mine hidden information links between discrete data has great advantages in the financial industry. Traditional risk management has always been based on manual experience, relying on the experience of the risk managers to make decisions. However, even the most experienced risk manager cannot make the correct or optimal decision for every potential risk every time. Machine learning can make predictions of credit risk quickly, and often more accurate than the results of manual predictions [1][2][3][4]. However, machine learning and deep learning often require a large amount of high-quality raw data for training [5][6] in order to obtain models that can effectively predict or judge.

To solve the dilemma of big data, the traditional method has become a bottleneck. Simple data exchange between two enterprises is not allowed in many regulations, including the GDPR (General Data Protection Regulation). Users are the owners of the original data. Without the approval of users, enterprises cannot exchange data. Secondly, the purpose of data modeling cannot be changed before user approval. Therefore, many attempts of data exchange in the past, such as data exchange, also need great changes to comply. At the same time, the data owned by enterprises often has great potential value. Two enterprises and even the departments within one enterprise may consider the exchange of interests. Under this premise, these departments often do not simply aggregate data with other departments. It will result in data frequently appearing as silos even within the same company.

In this case, Federated Learning incarnates its advantages. Unlike conventional machine learning, Federated Learning does not require companies to exchange data with each other to implement model training. Keeping the data locally for training means that using the data will not violate privacy protection regulations, nor will it result in the disclosure of corporate trade secrets. This is of great significance for solving the problem that SMEs do not have enough high-quality data to train models and how to protect the information security of users and enterprises.

In this paper, we combine the vertical federated learning with Bayesian Machine Learning and use the homomorphic encryption algorithm to design a federated learning system which is secure, stable, and effective. With a view to solve data silos between enterprises and protect user privacy.

2. Related Works

2.1. Federated Learning

Because of Alphago's great success people naturally hope that the data-driven AI will be realized in all walks of life, but the real situation is very disappointing. In addition to a limited number of industries, there are more areas with limited data and poor quality, which is not enough to support the implementation of artificial intelligence technology. There are two main reasons: 1) There are barriers between data sources that are hard to break. In most industries, data exists in the form of silos. Due to industry competition, user privacy, complicated administrative procedures and other issues, even data integration between different departments of the same company faces many obstacles. In reality, it is almost impossible to integrate the scattered data in different places and institutions. In other words, the cost is huge. 2) With the further development of big data, it has become a worldwide trend to attach importance to user privacy and data security. Every time the public data is leaked, it will cause great concern of the media and the public. At present, all countries are strengthening the protection of data security and user privacy. The increasingly strict management of user privacy and data security will be the worldwide trend, which brings unprecedented challenges to the field of artificial intelligence. The current situation in the research and business is that the party who collects data is usually not the party who uses the data. For example, Party A

collects data, transfers it to Party B for cleaning, then transfers it to Party C for modeling, and finally sells this model to Party D for use. This form of data transfer, exchange and transaction between entities violates the GDPR and may be severely punished by the laws. Similarly, *Cybersecurity Law of the People's Republic of China* and *General Provisions of the Civil Law of the People's Republic of China* which have been implemented since 2017, also point out that network operators can not disclose, tamper with or destroy the personal information they collect, and when conducting data transactions with third parties, it is necessary to ensure that the proposed contract clearly stipulates the scope and data protection obligations of the data to be traded. The establishment of these laws and regulations challenges the traditional data processing mode of AI in different degrees.

Therefore, how to design a machine learning framework on the premise of meeting the requirements of user privacy, data security and supervision, so that the AI system can use their data more efficiently and accurately, is an important topic in the development of AI.

Federated Learning is a solution proposed by many institutions and scholars to the dilemma of data isolation and data privacy. For the privacy of mobile terminal and multi-party organization data, Google and WeBank have built their different Federated Learning frameworks. Google's Federated Learning framework is based on personal terminal devices, and AAAI Fellow Prof. Yang Qiang and WeBank subsequently proposed a systematic and general solution based on Federated Learning, which can give a solution to the difficulty of co-modeling between individuals or companies. On the premise of meeting data privacy, security and regulatory requirements, a machine learning framework is designed to enable artificial intelligence systems to use their data more efficiently and accurately. Currently, the main research directions of Federated Learning are to overcome the statistical challenges and enhance security. Prof. Yang Qiang uses the gradient descent algorithm to train a linear regression model, and the model built in combination with the homomorphic encryption algorithm is a very good case. Federated Learning is in the initial stage of rapid development. Both WeBank and Google have launched their own open source Federated Learning frameworks, FATE (Federated AI Technology Enabler) and TensorFlow Federated. In order to accelerate the popularization and implementation of "Federated Learning", WeBank submitted a proposal to IEEE Standards Association in October 2018, "Guide for Architectural Framework and Application of Federated Machine Learning" (Federated Learning Infrastructure and Application Standards), which was approved in December 2018.

(1) Basic Notion

When considering traditional machine learning, the training process can be considered as an optimization problem, defined as:

$$\min_{\omega \in \mathrm{i}^{d}} f(\omega) \tag{1}$$

$$f(\omega) \overset{\mathrm{V}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(\omega) \tag{2}$$

In above formulas, $f_i(\omega)$corresponds to the loss function; Given the parameter ω, $f_i(\omega)$ is the predicted loss at the index of data point i. First, the basic concept of Federated Learning is presented in combination with Figure 1. Federated learning [7] consists of two main components, including central training and local training, and the

K clients that make up the system are identified by index k. The process is divided into multiple communication rounds. In each round, clients train the local model synchronously using local SGD on their private data sets P_k. On the central server side, server aggregates the uploaded client parameters. Specifically, the parameter from client k is ω^k, and $k \in S$. S corresponds to (in each communication round) a participating subset which contains m clients.

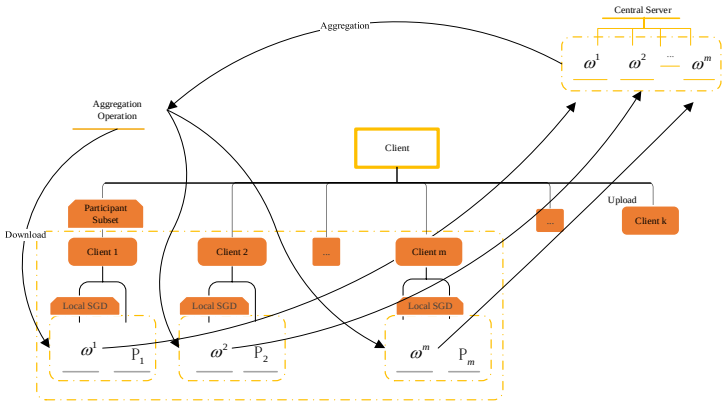

Fig. 1. Federal Learning System

For client k, the training data set owned by client k has n_k data points, and $n_k = |P_k|$. Therefore, the optimization problem under the Federated Learning can be redefined as:

$$f(\omega) = \sum_{k=1}^{k} \frac{n_k}{n} \cdot F_k(\omega) \tag{3}$$

$$F_k(\omega) = \frac{1}{n_k} \sum_{i \in P_k} f_i(\omega) \tag{4}$$

What needs to be added is that the classic assumption about training data in conventional distributed optimization, Independent and Identically Distributed Assumption (IID Assumption), training data is evenly and randomly distributed on each client, which is difficult to satisfy in the Federated Learning [8][9].

(2) Classification of Federated Learning

Matrix Di denotes the data held by each data owner i. Each row of the matrix represents a user sample, and each column represents a user's feature. At the same time, some data sets may also contain label data. We called the features space as X and the label space as Y. For example, in the financial field, the user's information is the label Y that needs to be predicted; in the sales field, the label is the user's purchase wish Y; in the education field, it is the degree of knowledge of the student. User feature X and label Y constitute the complete training data (X, Y). However, in implementation, it is often encountered that the users of different data sets or the user characteristics are not exactly the same. Specifically, taking Federated Learning with two data owners as an example, the data distribution can be divided into three cases [10][11][12]: 1) The features space $(X_1, X_2, \cdots)$ of the two data sets have a large overlap and the user space $(U_1, U_2, \cdots)$ have a small overlap. 2) The user space $(U_1, U_2, \cdots)$ overlap is larger and

the feature space $(X_1, X_2, \cdots)$ overlap is smaller. 3) The user space $(U_1, U_2, \cdots)$ and feature space $(X_1, X_2, \cdots)$ overlaps in both datasets are small Federated Learning can be divided into Horizontal Federated Learning, Vertical Federated Learning, and Federated Transfer Learning, as shown in Fig. 2.

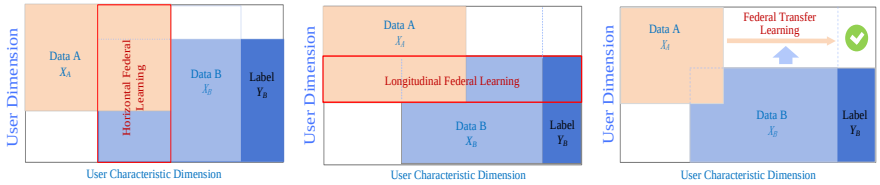

Fig. 2. Classification of Federal Learning

1) Horizontal Federated Learning

In the scenarios that data sets share the same feature space but different in users, we divide the data horizontally (the user dimension), and take the part of the data with the same feature space but not the same users for training. This approach is called Horizontal Federated Learning. For example, there are two banks in different regions. Their user groups come from their respective regions, and the intersection between them is small. However, their businesses are similar, so the features recorded are the same. And then, they can use Horizontal Federated Learning to build a model together. In 2017, Google proposed a data co-modeling solution for Android phone model updates: a Federated Learning scheme that when a single user uses an Android phone, the model parameters are continuously updated locally and uploaded to the Android cloud, so that each data owner with the same feature dimension can build a model jointly.

2) Vertical Federated Learning

In the scenarios that data sets share the same users but different in feature space, we divide the data vertically (the feature space dimension), and take the part of the data with the same users but not the same feature space for training. This method is called Vertical Federated Learning. For example, there are two different institutions, one is a bank and the other is an e-commerce company in the same place. Their users are likely to include most of the residents of the place, so the intersection of users is large. However, because banks record users' payment behaviors and credit ratings, while e-commerce companies store users' browsing and purchasing history, their feature space overlap is small. Vertical Federated Learning is a scheme that aggregates these different features in an encrypted state to enhance the ability of the model. At present, many machine learning models such as logistic regression models, tree structure models, and neural networks models have gradually been proven to be able to be built on this federated system.

3) Federated Transfer Learning

In the scenarios that the user and feature space of the two data sets have little overlap, we do not split the data, but use transfer learning instead to overcome the lack of data or labels. This method is called Federated Transfer Learning. For example, there are two different institutions, one is a bank located in China, and the other is an e-commerce company located in the United States. Due to geographical constraints, the user groups of the two institutions have very little intersection. At the same time, due to types of the

institutions are different, only a small part of the features overlap. In this case, in order to make federated learning system runs effectively, Federated Transfer Learning must be introduced to solve the problem of small unilateral data and small label samples, thereby improving the effectiveness of the model.

2.2. Homomorphic Encryption

In Federated Learning, Homomorphic Encryption is used to encrypt parameters to protect user privacy. Unlike differential privacy, the data and model will not be transmitted. So, there is no possibility of leakage at the data level, nor violation of the stringent data protection laws such as GDPR. Homomorphic Encryption [13][14][15] is a kind of encryption method that has special nature properties, this concept was first proposed in the 1970 s by Rivest et al., compared with the general encryption algorithms, homomorphic encryption can not only realize the basic cryptographic operations, but also achieve a variety of computing functions between the ciphertext [16][17]. In other words, first calculation and then decryption are equivalent to first decryption and then calculation. Gentry based on the ideal lattice for the first time in 2009 proposed the real practical Fully Homomorphic Encryption scheme (FHE). Gentry's scheme can suppress the growth of noise by performing a finite number of polynomial operations and using the compression and decryption circuit technology when performing homomorphic operations, so as to prevent the noise from growing too fast and exceeding a certain limit causes the plaintext value cannot be decrypted correctly. So far, after two stages of development, researchers have proposed a variety of fully homomorphic encryption schemes and corresponding optimization schemes. Most of the existing homomorphic encryption schemes can only support integer type homomorphic encryption, but do not support floating point types, so it cannot cover the practical application requirements. In addition, due to the inherent reasons such as algorithm construction and security guarantee, the efficiency of the algorithm and the storage occupancy of secret key and ciphertext are far from the standards of practical production and application. How to design and propose a new Fully Homomorphic Encryption scheme on the premise of ensuring the security of the Fully Homomorphic Encryption algorithm and overcoming the disadvantages of existing algorithms in noise control, execution efficiency, storage space, etc., will be an important direction of current research.

A solution proposed by Graepel to solve the problem of excessive noise caused by homomorphic encryption operation by mathematically expressing the prediction function of the model as a low-order polynomial. The research is about privacy protection in the training stage. This scheme effectively limits the number of homomorphic operations on encrypted data and limits homomorphic operations to addition and multiplication. Finally, it is applied to LM and FLD classifier and practical results are obtained. Later, David Wu et al. realized homomorphic encryption of large-scale data sets and high-dimensional data by using batch computing and CRT-based message encoding technology, and then performed linear regression and other statistical analysis on the encrypted data, and the results were suitable for the scenario of multi-source data. To some extent, homomorphic encryption only supports limited homomorphic operations, but low-order polynomials can satisfy this property. Therefore, Dowlin uses Chebyshev approximation theory to replace the nonlinear

activation function in the neural network model with a low-order polynomial function, thus realizing CryptoNets, a neural network that can process ciphertext. And through the experiment on the real data set MNIST, the rationality of the model is explained. The research is done in the classification stage, because the neural network model has better accuracy than the linear classifier, which is a good breakthrough. Bost et al. tried to process ciphertext in hyperplane decision, Naive Bayes and decision tree classifier, which were also studied in the classification stage. Since CryptoNets proposed by Dowlin did not perform well on deep neural networks, Chabanne et al. proposed improvements on this basis. The convolutional neural networks model proposed by them was more accurate than CryptoNets, which was also the first successful attempt to combine homomorphic encryption with deep neural network. In addition, Aono used additive homomorphic encryption to propose a logistic regression model that can be calculated on ciphertext. It can be seen that data based on homomorphic encryption has been applied to many common machine learning models and achieved good accuracy. However, it is worth mentioning that due to the complexity of the calculation model, the calculation time is generally much longer than that of the plaintext processing [18].

Homomorphic encryption is an encryption method that can perform any operation which can be performed on plaintext on encrypted data without decryption [16]. In other words, homomorphic encryption satisfies a specific algebraic algorithm for plaintext that is equivalent to another (possibly different) algebraic algorithm for ciphertext.

We perform some kind of processing on the encrypted data to generate a new ciphertext. The plaintext content generated by decrypting the ciphertext, is the same as the result that decrypt corresponding plaintext after encrypting operation. This method allows the ciphertext can be manipulated even when the data is encrypted, and the result of it will be the same as we expected. The process of homomorphic encryption is shown in Fig. 3. Encrypt and Decrypt represent corresponding encryption and decryption methods; the "*" operation in the plaintext space and the "#" operation in the ciphertext space are equivalent.

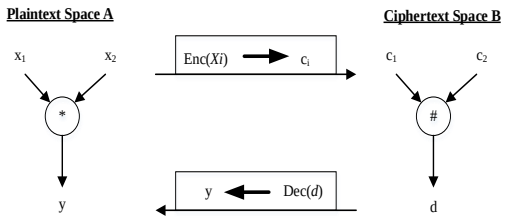

Fig. 3. Homomorphic Encryption Algorithm

Homomorphic encryption can only perform any number of times of additions or multiplications or finite additions and multiplications on the ciphertext, and its operation result is the same as the result of encrypting the plaintext directly after the corresponding operation.

Assume that an encryption scheme G is expressed as (M, C, K, E, D), where M is the plaintext space, C is the ciphertext space, K is the key space, E is the encryption

algorithm, and D is the decryption algorithm. Defines $\oplus$ as the ciphertext related operator.

Definition 1 Let P and L denote the operation, when the plaintext data set $M = \{m_1, m_2, \cdots, m_n\}, k \in K$, if: $P(E_k(m_1), E_K(m_2), \cdots, E_K(m_n)) = E_k(L(m_1, m_2, \cdots, m_n))$, says that the encryption scheme for operation L is homomorphism. The basic idea of homomorphic encryption is to achieve that after doing some operations on ciphertext we can still get the same result as doing operations on plaintext directly.

Definition 2 For any plaintext $m_i, m_j \in M$, the corresponding ciphertext is $c_i = E(m_i), c_j = E(m_j)$, and $c_i, c_j \in C$, if $(m_i + m_j) = E(m_i) \oplus E(m_j)$ or $D\left(E(m_i) \oplus E\left(m_j\right)\right) = m_i + m_j$ holds, then it is said that the encryption scheme G has the property of addition homomorphism.

Definition 3 For any plaintext $m_i, m_j \in M$, the corresponding ciphertext is $c_i = E(m_i), c_j = E(m_j)$, and $c_i, c_j \in C$, if $E(m_i m_j) = E(m_i) \oplus E(m_j)$ or $D(E(m_i) \oplus E(m_j)) = m_i m_j$ holds, the encryption scheme G is said to have multiplicative homomorphism.

Definition 4 For any plaintext $m_i, m_j \in M$, the corresponding ciphertext is $c_i = E(m_i)$, and $c_i \in C$, if $E(m_i m_j) = E(m_i) \oplus m_j$ or $D(E(m_i) \oplus m_j) = m_i m_j$ holds, then it is said that the encryption scheme G has the property of mixed multiplication homomorphism.

Definition 5 If scheme G has both the addition homomorphism and multiplication homomorphism properties, and can satisfy finite addition and multiplication ciphertext operations, then the encryption scheme G is said to be a somewhat homomorphic encryption scheme.

Definition 6 If scheme G has both addition homomorphism and multiplication homomorphism properties, and can satisfy any number of times of addition and multiplication ciphertext operations, then the encryption scheme G is said to be a fully homomorphic encryption scheme [19].

2.3. EM Algorithm

The EM algorithm [20] was formally proposed by Arthur Dempster, Nan Laird and Donald Rubin in their research paper Maximum likelihood from incomplete data via the EM algorithm in 1977. They summarized the previous EM algorithm as a special case and gave the calculation steps of the standard algorithm. After that, EM algorithm became a method to deal with incomplete observation data, which attracted much attention from all sides and was continuously studied in depth. Since 1977, there were many new applications and improvements of the algorithm [3][21]. After decades of development, the EM algorithm has been widely used to process incomplete data in medicine, engineering, business management, sociology, finance and other fields where large data volume is required. The EM algorithm is an iterative algorithm. As with most iterative algorithms, the EM algorithm requires the user to make an initial assumption about the parameters to be solved, and then continuously update the value of this set of parameters until there are no more noticeable changes. For EM algorithms, different initial parameters often lead to different results. That is, the EM algorithm cannot

guarantee that the results obtained are optimal. Theoretical analysis shows that the EM algorithm can only guarantee the locally optimal solution. In addition to the local optimal problem (or the corresponding initial sensitivity) mentioned above, the EM algorithm has another tricky problem that is the user needs to pre-set the number of gaussian members in the GMM model (each gaussian function in the gaussian mixture model is called a member, or Component). This problem is tricky because the GMM model deals with a bunch of unlabeled data, meaning that it is not clear which gaussian member is responsible for each data point. How to overcome the local extremum problem (or initial sensitivity problem) of EM algorithm and how to determine the number of gaussian members in the mixed model have been two open problems in academia. There are two simple but common approaches to local optimal problems. One is to test multiple times (select different initial parameters) and take the parameters estimated by the test with the largest likelihood function as the final result. Obviously, this is easy to do operationally but the disadvantage of being time-consuming cannot be ignored. Another approach is to use other clustering methods, such as k-means, to find good initial parameters for the EM algorithm. But the problem is that k-means has its own initial sensitivities issue. Therefore, the stability and reliability of the final results are difficult to be guaranteed. However, because k-means is very simple, this method is also more common. For the problem of how to determine the number of gaussian members, there are representative deterministic methods based on Bayesian [22] Information Criterion and relevant methods based on information theory, such as Minimal Description Length, Minimal Message Length and Akaike information criterion (AIC).

Expectation-Maximization algorithm (EM), or Dempster-Laird-Rubin algorithm, is a type of optimization algorithm that iteratively performs Maximum Likelihood Estimation (MLE) [23]. This algorithm is usually used as a replacement for the Newton-Raphson method for parameter estimation of probability models that include latent variables or incomplete-data [24]. The standard calculation framework of EM algorithm alternately consists of E-step (Expectation-step) and M-step (Maximization step). The convergence of the algorithm can ensure that the iteration approaches at least the local maximum. The EM algorithm is one of the special cases of the MM algorithm (Minorize-Maximization algorithm). There are several improved versions, including the EM algorithm using Bayesian inference [22][25], the EM gradient algorithm, and the generalized EM algorithm. Because iterative rules are easy to implement and allow for flexible consideration of latent variables, the EM algorithm is widely used to handle missing measurements of data and parameter estimation for many machine learning algorithms, including Gaussian Mixture Model (GMM) and the Hidden Markov Model (HMM).

Given independent observation data $X = \{X_1, \cdots, X_N\}$, and a probability model $f(X, Z, \theta)$ containing the hidden variable Z and parameter θ, according to MLE theory, when the likelihood of the model is maximized the optimal single-point estimate of θ in the model is given: $\theta = arg \max_{\theta} p(X|\theta)$.

$$p(X|\theta) = \int_a^b p(X, Z|\theta) dZ, \quad Z \in [a, b] \tag{5}$$

$$p(X|\theta) = \sum_{c=1}^{k} p(X, Z_c|\theta), \quad Z \in \{Z_1, \cdots, Z_k\} \tag{6}$$

Latent variables can represent missing data or any random variables that cannot be directly observed in the probability model. In the formula, the first line is the case where the latent variable is a continuous variable, and the second line is the case where the latent variable is a discrete variable. The integral or summation is also called the joint likelihood of X, Z. Without losing generality, here we use discrete variables as an example. According to the conventional method of MLE, the natural logarithm of the above formula can obtain:

$$\log p(X|\theta) = \log \prod_{i=1}^{N} p(X_i|\theta) = \sum_{i=1}^{N} \log p(X_i|\theta) = \sum_{i=1}^{N} \log[\sum_{c=1}^{k} p(X_i, Z_c|\theta)] \tag{7}$$

The above expansion considered the mutual independence of the observed data. Introduce the probability distribution q (Z) related to the latent variable, that is, the latent distribution (the latent distribution can be considered as the posterior of the latent variable to the observation data, see the E-step derivation of standard algorithms). From the Jensen inequality, the log-likelihood of the observed data has the following inequality relationship:

$$\log p(X|\theta) = \sum_{i=1}^{N} \log[\sum_{c=1}^{k} \frac{q(Z_c)}{q(Z_c)} p(X_i, Z_c|\theta)] \geq \sum_{i=1}^{N} \sum_{c=1}^{k} [q(Z_c) \log \frac{p(X_i, Z_c|\theta)}{q(Z_c)}] = L(\theta, q) \tag{8}$$

When θ, q makes the global maximum on the right side of the inequality, the θ obtained at least makes the left side of the inequality local maximum. Therefore, after expressing the right side of the inequality as $L(\theta, q)$, the EM algorithm has the following goal:

$$\hat{\theta} = \arg\max_{\theta} L(\theta, q) \tag{9}$$

$L(\theta, q)$ in the above formula is equivalent to the surrogate function in the Minorize-Maximization algorithm, which is the lower limit of the MLE optimization problem. The EM algorithm approximates the maximum of log-likelihood by maximizing the surrogate function. The EM algorithm and its improved versions are used to solve parameters of machine learning algorithms. Common examples include Gaussian Mixture Model (GMM), Probabilistic Principal Component Analysis, Hidden Markov Model (HMM) and other unsupervised learning algorithms.

3. Method

3.1. Problem Statement

A clustering problem. In the risk control case, the primary step is to do clustering for thousands of companies. It simplifies the risk control problem and increases efficiency.

We implement Vertical Federated Expectation Maximum Algorithm (Hetero-EM) to group companies into K clusters with data from different organizations.

3.2. The System

Subscript: client $m \in M$; cluster $k \in K$; sample $i \in N$; feature $j \in D$

(1) Assumptions

i. distribution assumption: assume each x_i is sampled from one of K distribution.

ii. independent assumption: assume each x_m is independent from each other.

(2) Vertical Federated Expectation Maximum Algorithm (Hetero-EM)

In vertical federated expectation maximum algorithm (Hetero-EM), we combine local distributions from clients to get aggerated updates. Instead of using Federated Averaging as Federated Neutral Network systems do, it integrates distributions on each round based on the independent assumption. Hence, the arbiter only knows the aggregated value of each user from each client, and it is hard to infer the exact information of a user, which inherent privacy. All in all, it is a Bayesian aggerated update that not only captures uncertainty but also make data more private from posterior sampling.

Algorithm Vertical Federated Expectation Maximum Algorithm for clustering

Input: M local datasets X_m, the number of clusters K
Output: global distribution π, local distribution θ, clusters assignment distribution φ

1) Build the training plan. The plan includes Hetero Mixture Models for the clustering problem, which uses the EM algorithm to do parameter updates among M clients.
2) Deploy the FL arbiter. The arbiter mainly distributes the FL plan to the clients, and receives local distributions, integrates the global distribution.
3) Deploy clients. Each client preprocesses their data and updates local distribution and uses encryption algorithms to encrypt sensitive data on local distributions.
4) The arbiter initiates communication. At this stage, the arbiter sends a signal to the client server, telling it the conditions required for the training plan, such as memory, capacity, size of collected data, etc.
5) Client response the arbiter. After receiving the signal from the arbiter, the client responds to the server and returns information (data size, time, etc.) to the arbiter.
6) The arbiter sends clients the training plan.
7) Start the federated training. The arbiter initializes parameters π_0 and θ_0. For each round t=1, 2… T:

Client m executes:

$$p_m\left(\mathrm{x}_i \middle| \theta_{k_{(t-1)}}\right)$$

$$crypto = E\left(p_m\left(\mathrm{x}_i \middle| \mu_{k_t}, \sigma^2{}_{k_t}\right)\right)$$

Send crypto to the arbiter.

Arbiter executes:

For each cluster k =1, 2…K and for each sample i = 1, 2… N

Decrypt the received crypto

$$\phi_i(\mathrm{k})_{(t)} = \frac{\pi_{k_{(t-1)}} \cdot \prod_m^M p_m\left(\mathrm{x}_i \middle| \theta_{k_{(t-1)}}\right)}{\sum_{\mathrm{k}=1}^K \pi_{k_{(t-1)}} \prod_m^M p_m\left(\mathrm{x}_i \middle| \theta_{k_{(t-1)}}\right)}$$

$$\mathrm{n}_{k_{(t)}} = \sum_{i=1}^{N} \phi_i(\mathrm{k})_{(t)}$$

$$\pi_{k_{(t)}} = \frac{\mathrm{n}_{k_{(t)}}}{n}$$

Send the aggregated distribution φ and π to M clients.

Client executes:

Get the aggregated distribution φ from the arbiter.

For each cluster k =1, 2…K:

$$\theta_{k_{(t)}} = \frac{1}{\mathrm{n}_{k_{(t)}}} \sum_{i=1}^{N} \phi_i(\mathrm{k})_{(t)} \mathrm{x}_i$$

Use local parameters to update to the artier.

8) The arbiter stop training under stop conditions and then send signal to clients.

4. Experiments

We conducted experiments for a clustering task on two datasets of companies from two organizations, where the data have the same sample IDs but own disjoint subsets of features.

4.1. Data

We train numeric Dataset A with 5000 samples and 13 features and numeric Dataset B with 5000 samples and 10 different features. These two datasets includes info about 5000 companies and open to the public @ https://gitee.com/mirrors/FATE/blob/master/examples/data/default_credit.csv. After PCA compression and scale, we have 6 features for Dataset A and 5 features for Dataset B. We have two assumptions on datasets:

i. normal distribution assumption: each x_i is sampled from Normal (μ, σ^2).

ii. independent assumption: Normal $(\mu, \sigma^2) \propto \mathrm{Normal}_A\ (\mu, \sigma^2) \cdot \mathrm{Normal}_B\ (\mu, \sigma^2)$.

Hence, on each round, client m executes:

$$\mu_{mk_{(t)}} = \frac{1}{\mathrm{n}_{k_{(t)}}} \sum_{i=1}^{N} \phi_i(\mathrm{k})_{(t)} \mathrm{x}_i \tag{10}$$

$$\sigma^2{}_{mk_{(t)}} = \left(\frac{1}{\mathrm{n}_{k_{(t)}}} \sum_{i=1}^{N} \phi_i(\mathrm{k})_{(t)} (\mathrm{x}_i - \mu_{i(t)})\ (\mathrm{x}_i - \mu_{i(t)})^T\right) \tag{11}$$

After updating the parameters of local distribution,

$$E\left(p_m\left(\mathrm{x}_i \middle|\ \mu_{k_t}, \sigma^2{}_{k_t}\right)\right) \tag{12}$$

Where,

client $m \in M$; $M = \{A, B\}$

cluster $k \in K$; number of clusters: K = 2, 3, 5

feature $j \in D$; number of features: D = D1+D2 = 6+5 =11

sample $i \in N$; number of samples: N = 5000

homomorphic encryption algorithms E: {Paillier, RSA, BFV, CKKS}

4.2. Results

As shown in Fig. 4, we train for 2, 3, 5 clusters, compared with result from centered data training. The label of dataset is K=2 for good-credit companies and bad credit companies. The f1 score for hetero-EM is 0.68 while that of centered-EM is 0.35.

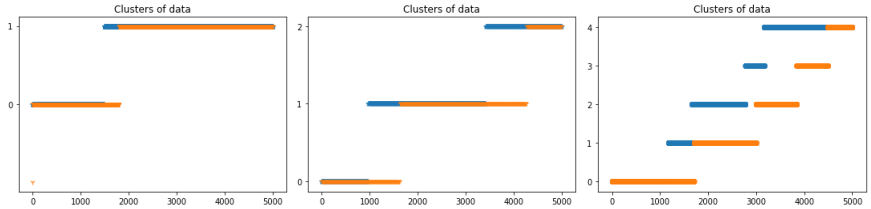

Fig. 4. Clusters of Data for K = 2, 3, 5

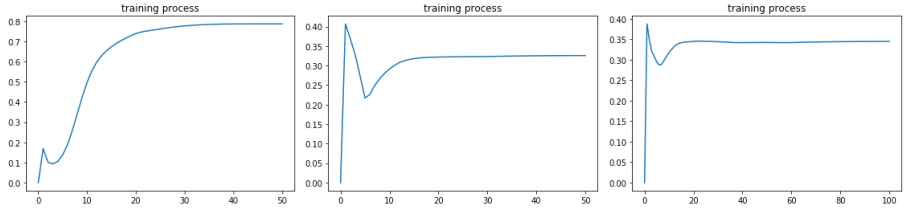

Fig. 5. The Probability of Cluster 0 During Training for K = 2, 3, 5

Table 1. Method Comparison

	Hetero-EM(GMM)	EM(GMM)
Date distribution	Decentralized	Central
K = 2; C = 95.25%	[0.21, 0.79]	[0.23, 0.77]
K = 3; C = 96.35%	[0.21, 0.48, 0.31]	[0.20, 0.31, 0.49]
K = 5; C = 88.55%	[0.24, 0.11, 0.15, 0.30, 0.20]	[0.18, 0.18, 0.26, 0.14, 0.24]

We evaluate the comparability C by the absolute distance.

$$C = (1 - |\pi_{hetero} - \pi_{center}|) * 100\% \quad (13)$$

Table 2. Homomorphic Encryption Comparison: Hetero-EM clustering for companies with K = 2; the dtype of each element is "float64"

	PHE-A	PHE-M	FHE
Method	Paillier	RSA	CKKS
Keys Pair	keySize: 2048bits	p/qSize: 1024bits	polyDegree: 8192bits
Generation	t: 1.87s	t: 0.338s	t: 0.0849s
Encryption	205ms/sample	28.9μs/sample	10.9μs/sample
Decryption	57.4ms/sample	792μs/sample	12.5μs/sample
Key Size	2048bits	2048bits	4096bits

We test different encryption algorithms on the Hetero-EM clustering for companies with K = 2. According to the table, the full homographic encryption algorithm (FHE)-CKKS is safe and the most efficient, followed by partial multiplication homographic encryption algorithm (PHE-M) and partial addition homographic encryption algorithm (PHE-M). All three encryption algorithm is safe with key pairs larger or equal to 2048 bits, but PHE-M is not efficient compared to FHE and PHE-M.

5. Conclusions

With improvements of artificial intelligence and arrival of the era of big data, contradictions between data sharing and data privacy are becoming a big problem. Considering of user privacy and trade secrets, it is difficult for enterprises to exchange their raw data for co-model building. In this paper, we develop a hetero-EM federated ML based on the homomorphic encryption algorithm. Each client is assumed to provide a local distribution, which is modeled through our framework. We test three encryption algorithms during communications. Besides, we design an inference strategy that allows us to integrate the global distribution in a single communication step without complex data pooling in the server. We then demonstrate the efficacy and efficiency of our Method on federated learning problems simulated from two private financial datasets.

The further works are as follows: 1) To build an improved federated learning system that is compatible to other Bayesian methods. This will make federated learning system more flexible; 2) To explore the intrinsic strategy of data spite in homomorphic encryption. This will make it more efficient and time-saving; 3) To construct a federated learning model based on Blockchain, protect user privacy and data security better.

References

1. Li, Z.: Research on Financial Risk Control Model and Algorithm Based on Machine Learning. North China University of technology, Beijing. (2019)
2. Zhang, L.: Research on Network Loan Credit Risk Prediction Model Based on Machine Learning Algorithm. Lanzhou University, Lanzhou. (2019)
3. Gan, L.: Research on Credit Risk Prediction Model Based on Machine Learning Technology – a Case Study of an Internet Financial Company. Beijing Jiaotong University, Beijing. (2017)
4. Gao, H., Xu, Y., Yin, Y., Zhang, W., Li, R., Wang, X.: Context-aware Qos Prediction with Neural Collaborative Filtering for Internet-of-Things Services. IEEE Internet of Things Journal(IoT-J), 7(5), 4532-4542. (2020)
5. Zhao, J.: Research on Credit Risk Prediction Method Based on Big Data Financial Cloud Platform. Chang'an University, Xi'an. (2017)
6. Ma, X., Gao, H., Xu, H., Bian, M.,: An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing. EURASIP Journal on Wireless Communications and Networking, 2019(1), 1-19. (2019)
7. Konečný, J., McMahan, B., Ramage, D.: Federated Optimization: Distributed Optimization Beyond the Datacenter. Mathematics, 2(1), 115. (2015)
8. Liu, M., Liao, B., Ding, L., Xiao, L.: Performance Analyses of Recurrent Neural Network Models Exploited for Online Time-Varying Nonlinear Optimization. Computer Science and Information Systems, 13(2), 691–705. (2016)
9. Lasek, P., Gryz, J.: Density-Based Clustering with Constraints. Computer Science and Information Systems, 16(2), 469–489. (2019)
10. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated Machine Learning: Concept and Applications. ACM Transactions on Intelligent Systems, 10(2), 12.1-12.19. (2019)
11. Yang, Q.: AI and Data Privacy Protection: The Way to Federated Learning. Journal of Information Security Research, 5(11), 961-965. (2019)
12. Gao, H., Kuang, L., Yin, Y., Guo, B., Dou, K.: Mining Consuming Behaviors with Temporal Evolution for Personalized Recommendation in Mobile Marketing Apps. ACM/Springer Mobile Networks and Applications (MONET), 1-16. (2020)
13. Shi, J.: Parallel Algorithm of Fully Homomorphic Encryption over Floating-Point Based on Spark. Nanjing University of Posts and Telecommunications, Nanjing. (2018)
14. He, W.: Research on Key Technologies of Privacy-Preserving Machine Learning Based on Homomorphic Encryption. University of Electronic Science and Technology of China, Chengdu. (2019)
15. Wang, J.: Homomorphic Encryption Secure Access Control Algorithm and Cloud Computing Research Based on Hyperelliptic Curve Encryption. Shaanxi University of Science and Technology, Shaanxi. (2019)
16. Xia, C.: Research of Homomorphic Encryption Technology and Application. Anhui University, Anhui. (2013)
17. Gao, H., Liu, C., Li, Y., Yang, X.: V2VR: Reliable Hybrid-Network-Oriented V2V Data Transmission and Routing Considering RSUs and Connectivity Probability. IEEE Transactions on Intelligent Transportation Systems, 1-14. (2020)
18. Meng, S.: A Survey of Machine Learning Based on Homomorphic Encryption. Computer Knowledge and Technology, 15(5), 182-183. (2019)
19. Feng, C.: Research on Related Algorithms of Fully Homomorphic Encryption. Shandong University, Shandong. (2015)
20. Qiu, T.: EM Algorithm and Its Application Research Based on Gaussian Mixture Models. University of Electronic Science and Technology of China, Chengdu. (2015)
21. Yang, X., Zhou, S., Cao, M.: An Approach to Alleviate the Sparsity Problem of Hybrid Collaborative Filtering Based Recommendations: The Product-Attribute Perspective from User Reviews. MOBILE NETWORKS & APPLICATIONS, 25(2), 376-390. (2020)

22. Zheng, Y.: Research of Large-Scale Engineering Projects' Schedule Risk Research Based on Bayesian Network. Southwest Jiaotong University, Chengdu. (2014)
23. Cao, J.: Greedy EM Algorithm Based on MapReduce Framework. Anhui University of Science and Technology, Anhui. (2018)
24. Xu, T.: The Research of EM Algorithm in Incomplete Monitoring Data Processing. Chengdu University of Technology, Chengdu. (2017)
25. A, M.: Research and Application on Naive Bayes Classification Algorithm. Dalian University of Technology, Dalian. (2014)

Wei Ou (corresponding author) is a lecturer at the Department of Electronic and Information Engineering, Hunan University of Science and Engineering. He received the B.S. degree from the Air Force Engineering University, in 2000, and the M.S. degree in the National University of Defense Technology in 2005, and the Ph.D. degree in the National University of Defense Technology, in 2013. His current research interests include cryptography, cyber security, AI security and Blockchain technology.

Jianhuan Zeng is a software engineer at the Artificial Intelligence Research Center, Qianhai Institute for Innovation Research. She received the B.S. degree from the Sun Yat-sen University, in 2017, and the M.S. degree in Columbia University in 2019. Her current research interests include cyber security and AI security.

Zijun Guo is currently with the Department of Electronic and Information Engineering, Hunan University of Science and Engineering. His current research interests include AI security and Blockchain technology.

Wanqin Yan is currently with the Department of Electronic and Information Engineering, Hunan University of Science and Engineering. Her current research interests include cyber security and AI security.

Dingwan Liu is currently with the Department of Electronic and Information Engineering, Hunan University of Science and Engineering. Her current research interests include cryptography and AI security.

Dingwan Liu is currently with the Department of Electronic and Information Engineering, Hunan University of Science and Engineering. Her current research interests include cryptography and AI security.

Stelios Fuentes is currently with Leicester University, UK. His research interests include distributed computing, recommendation systems, and big data.

Received: September 23, 2019; Accepted: March 13, 2020