Personalized Recommendation Algorithm for Mobile Based on Federated Matrix Factorization

Junjie Jia and Zhipeng Lei*
School of Computer Science and Engineering, Northwest Normal University, Lanzhou, China
*Corresponding author: zhipenglei@nwnu.edu.cn

Abstract. There is a problem that the amount of users’ preference data on the mobile is small, and users are unwilling to disclose the preference data for the recommendation system about mobile users, so the server can’t centrally train a large amount of users’ preference data for a personalized recommendation. This paper proposes a personalized federated matrix factorization algorithm by introducing a federated matrix factorization model. The algorithm introduces users’ and items’ biases to modify the predictive rating model on each mobile; At the same time, conformity is introduced to give different weights to the preference data. In the case that the preference data does not leave the mobile, but the user preference data of the multiparty mobiles is shared, the multi-party mobiles and the server jointly train personalized matrix factorization model. The experimental results show that the algorithm in this paper still has high recommendation accuracy under the premise of correct updating in the federated matrix factorization model that uses bias and conformity.

Keywords: Personalized recommendation, federated matrix factorization, mobile, bias, conformity.

1. Introduction
With the rapid development of mobile Internet and communication technology, people have got rid of the shackles of fixed terminals. By carrying portable smartphones, wearable devices, tablet computers, and other mobile devices, People can conveniently work, communication, social interaction, shopping, and other activities on the Internet anytime and anywhere. Data on activities is collected by mobile devices in real-time and regularly sent to third parties, such as service providers or institutions, then using data mining for market analysis or user recommendation. As a classic application of the machine learning model, the user recommendation system for mobile devices effectively solves the problem of information overload through a personalized recommendation of users by a third party. Through mobile devices for in-depth mining, this recommendation system collects a large amount of historical behavior data of users and analyzes the preferences and needs of different users by using mathematical tools, which can achieve accurate push to users. Such as product recommendation, advertising, etc. But at the same time, it may also disclose users' privacy, such as user's personal information, health status, geographical location, etc.
It can be seen that there are privacy risks in recommending for mobile users. On the one hand, the third-party data server protects the privacy of the collection and analysis process of group users. Some researchers have proposed using encryption methods [3, 4] and differential privacy methods [5, 6] to protect the privacy of the recommendation system, but the preference data of mobile users still leaves the local area, and there is a risk of user privacy leakage. On the other hand, the traditional recommendation system is based on the support of large-scale databases. In the real situation, the amount of data on the mobile of a single user is small, and many users are unwilling to disclose personal privacy. It is difficult to achieve the purpose of recommendation through centralized training of large-scale user data. To solve this problem, Google proposed the concept of federated learning in 2016 [7]. Under the federated learning mechanism, each participant (user) has the same identity and status, and a shared data strategy can be established to update the model parameters by controlling the communication and calculation between the participants. Because the data is not transferred, it will not leak user privacy or affect data specifications [8].

At present, with the continuous in-depth research of federated learning technology, there are gradually some explorations in applying federated learning to the recommendation system. Ammad et al. proposed a Federated Collaborative Filtering (FCF) recommendation method in 2019 [9]. In FCF, personal rating data is stored on the user client, and the local client calculates the embedded gradient of users and items. Users are embedded and maintained in local user clients and are directly updated using the local gradient on each client. Then, based on the most commonly used matrix factorization [10] and factorization machine [11] algorithms in current recommendation systems, WeBank proposed federated matrix factorization (FedMF) [12], federated factorization machine (FedFM) [24], and other federated recommendation algorithms. Tao Qi et al. proposed a Federated News Recommendation (FedNewsRec) method in 2020 [13], the user behavior on the news platform (website or application) is stored on the local user’s device and not uploaded to the server. The server is used to maintain the news recommendation model and update it through model gradients from a large number of users. Adrian Flanagan et al. proposed a federated multi-view matrix factorization (FED-MVMF) method in 2020, which is the first federated multi-view matrix factorization method with auxiliary information sources [14].

The FedMF method uses matrix factorization technology and adopts a distributed learning and homomorphic encryption scheme. The preference data (rating) of each user is kept locally, using the traditional stochastic gradient descent method [15], the encryption gradient is continuously updated interactively between each participant and the third-party server, and finally, the minimum loss function is satisfied. And get each participant’s prediction rating for the project to achieve the purpose of recommending participants. FedMF overcomes the privacy problems of traditional matrix factorization recommendation systems that leak user preference data and feature vectors. It can ensure that user information does not leave the local area, and it can make corresponding recommendations to users based on preventing privacy leakage. For mobile users, how to improve recommendation accuracy and make accurate recommendations to users while protecting personal privacy has become an urgent problem that needs to be resolved.

This paper proposes a personalized Federated Matrix Factorization (PER-FedMF) recommendation algorithm. This algorithm considers two issues: (1) There are individual differences in the rating standards of mobile users; (2) There are abnormal rating preferences for mobile users. Therefore, the main contributions of this paper are as follows: (1) Using the FedMF recommendation algorithm to introduce bias and conformance correction parameters on the mobile to eliminate user rating differences and abnormal ratings. On the premise of protecting personal privacy, it guarantees the recommendation accuracy of the FedMF model; (2) It is proved that the correction parameters of bias and conformity are updated correctly in the FedMF model.
2. Related

2.1. Federated matrix factorization

The goal of the algorithm in the recommendation system is to explore the connections between users and content products. Yang et al. [16] summarized federated recommendation algorithms into three categories, including horizontal federated recommendation algorithm (product-based federated recommendation), vertical federated recommendation algorithm (user-based federated recommendation), and migration federated recommendation. Horizontal federal recommendation mainly solves the problem of how to build collaboratively a recommendation system when participants have a large number of the same products and different user groups. For example, the federation of data between branches of the same movie company in different regions. Vertical federal recommendation mainly solves the problem of how to build collaboratively a recommendation system when the participants (institutions) have a large number of the same users and different products. For example, a federation of news recommendation services providers and video recommendation service providers, or a federation between recommendations service providers and user data providers. The migration federation recommendation mainly solves the problem of how to share collaborative experience to build a recommendation system when the participants have not many users and products. This paper focuses on the application scene in Figure 1 (different mobile users but the same products). For example, the products (movies) faced by users on a certain movie website are the same, but mobile users are different. Therefore, this paper mainly uses the horizontal federated matrix factorization recommendation algorithm as the basis to modify the recommendation accuracy. The concepts of additive homomorphic encryption, horizontal federated matrix factorization scene, and algorithm processes involved are as follows.

2.1.1. Additive homomorphic encryption. Additive homomorphic encryption (HE) is an encryption method commonly used in federated matrix factorization to protect user privacy through the exchange of encryption parameters. Homomorphic encryption allows any third party to operate the encrypted data without prior decryption. This paper uses a classic additive encryption scheme Paillier [22]: The additive homomorphic encryption has the following properties: (1) Encrypted numbers can be multiplied by non-encrypted scalars; (2) Encrypted numbers can be added together; (3) Encrypted numbers can be added to non-encrypted scalars. The encryption formula is generally as follows:

\[ E(x_i) + E(x_j) = E(x_i + x_j) \quad \forall x_i, x_j \in X \] (1)

Where E is the encryption algorithm and X is the collection of all data. It usually consists of the following functions: key generation, encryption process, decryption process, ciphertext addition operation, decryption, to get the added ciphertext.

2.1.2. Horizontal federation matrix factorization scene. The horizontal federated matrix factorization algorithm can be divided into two scenes according to the different rating data parties [23]. The first scene is that the rating data party is a mobile user of the recommendation system. The second scene is that the rating data party is an institution with a large amount of user data (such as movie websites, e-commerce websites). The specific scenes are as follows:

(1) The data party is the mobile user of the recommendation system

As shown in Figure 1, taking three mobile users as an example, the mobile users of the recommendation system have their rating data. And the user is a node, each user saves his row in the rating matrix locally. This row includes the user's ratings for different items, and when the data is not available on the mobile, its rating data is trained in the federated recommendation model in collaboration with other users.

(2) The data party is an institution with a large amount of user data
As shown in Figure 2, two institutions as an example, the rating data owned by different institutions are about the same item, but the user groups of the two institutions are different. Each data party has data in all the columns of several rows of the rating matrix, these rows correspond to the user groups owned by different data parties. For example, user rating data from different regions of the same movie company. In this case, each data party has a large number of users' rating data. In the case that the rating data does not come out of the local data party, these data parties can still use their data to collaboratively train the federated recommendation model.

![Figure 1. Diagram of the mobile user.](image1)

![Figure 2. Diagram of an institution with a large amount of user data.](image2)

2.1.3. Algorithm process. The loss function of the FedMF algorithm is based on the loss function of the matrix factorization, which has been widely used to mine the interaction between users and commodities. Funk [17] pointed out that the rating dataset of m users on n items in the original data can be transformed into a user-item rating matrix r. The original rating matrix r is decomposed into a user matrix p and an item matrix q with dimensions $m \times k$ and $n \times k$, in which the hidden feature k is mathematically expressed as $r \approx p \cdot q^T$. Each row in p indicates how much each user likes the different attributes of the item, each row in q represents the weight of the attribute or feature in the item [18].

The predicted rating $\hat{r}_{i,j}$ of user i for item j can be obtained by the formula $\hat{r}_{i,j} = p_{i,k} \cdot q_{j,k}^T$, where $p_{i,k}$ and $q_{j,k}$ denote the i-th row of the potential factor matrix p and the j-th row of q respectively. For the update of user matrix q and item matrix q, this paper adopts the stochastic gradient descent method [15]. The measurement standard for the two is determined by the minimized loss function $e_{i,j}$.

$$e_{i,j}^2 = \min_{p,q} \sum_{i=1}^{m} \sum_{j=1}^{n} (r_{i,j} - \hat{r}_{i,j})^2 + \frac{\beta}{2} (\|p\|_F^2 + \|q\|_F^2)$$

(2)
The loss function represents how close the predicted rating is to the original rating, where $\beta$ is a regularization parameter and $\beta/2 (||p||^2 + ||q||^2)$ is a regularization term to prevent the model from overfitting.

This paper mainly focuses on the horizontal federated matrix factorization of a scene (1). As shown in Figure 1, assuming that each participant is a mobile user, each participant’s data is the mobile user’s historical profile or historical information (user’s rating data on items), assuming that the federal recommendation system has $x$ mobile users as $S_1, S_2, \ldots, S_x$. The 1st to the $x$-th mobile user has $m_1, m_2, \ldots, m_x$ user groups respectively (the number of users in each user group is 1), and $m_1 + m_2 + \cdots + m_x = m$. To protect the privacy of participants, the FedMF algorithm isolates user rating data to prevent the server from directly accessing user rating data. Minimize the loss function on the mobile and the server to update the user matrix $p$ and the item matrix $q$ to minimize the difference between the predicted rating $\hat{r}_{h,i,j}$ and the real rating $r_{h,i,j}$ on the mobile user. The matrices $p$ and $q$ can be determined by minimizing the loss function (3). Then the loss function of the federated matrix factorization is as formula (3):

$$
e_{ij}^h = \min \sum_{r \in S_i} \sum_{j \in P(r)} \left(r^h_{r,j} - \hat{r}_{r,j}^h\right)^2 + \lambda ||q||_2^2 + \mu \sum_{k=1}^x ||p^k||_2^2, h \in (1,x)$$

Where $S_i$ represents the historical dataset of the $i$-th user, $p^h$ represents the user matrix of the $h$-th mobile, $r^h_{r,j}$ represents the real rating of the $h$-th mobile user, $\hat{r}_{r,j}^h$ represents the predicted rating of the $h$-th mobile user, $\lambda ||q||_2^2 + \mu \sum_{k=1}^x ||p^k||_2^2$ represents the regularization item of the item matrix $q$, and the $h$-th mobile user matrix to prevent overfitting.

The update of the user matrix and the item matrix uses the federated matrix factorization algorithm FedMF [6] for training. The general model framework is shown in Figure 3. Assuming that the user and the server have realized the generation and distribution of the key, the server has the public key $pk$, the user has the same private key $sk$, the public key $pk$ is shared by the server and all the users, and the private key $sk$ is only on the user the specific training process is as follows:

1. Initialize the parameters. Including the user matrix $p$ on the client and the item matrix $q$ on the server, the server encrypts $q$ with the public key $pk$, get the encrypted item matrix $\text{PEnc}(q, pk)$, where $\text{PEnc}$ is the encryption process of Algorithm Paillier;

2. The server provides an encrypted $C_q$ matrix for all clients to download;

3. i. The user downloads $C_q$ from the server, decrypts it with the private key $sk$ to obtain $q = \text{PDec}(sk, C_q)$, where $\text{PDec}$ is the decryption process of Paillier the algorithm. Then the client derives the $p$ in the loss function to obtain the $p$ gradient $\nabla p^t = 2\gamma \sum_{j \in (1,j)} q^h_{i,j} \left(r^h_{r,j} - \hat{r}_{r,j}^h\right)$. The update of $p$ is $p^t_{i,j} = p^{t-1}_{i,j} - \gamma \nabla p^t_{i,j}$, where $h \in (1,x)$, $t$ represents the number of iterations, $\gamma$ represents the learning rate, and $q_{i,j}$ represents the $j$-th row of the item matrix, $D^*$ represents the batch training data from the $h$-th client;

ii. The client derives $q$ to obtain the gradient $\nabla q^t_{j,i} = 2\gamma \sum_{j \in (1,j)} p^h_{i,j} \cdot$ of the vector $q_{j,i}$.

iii. The public key $pk$ encrypts $G^t_{h,i,j}$, get $C_{G^t_{h,i,j}} = \text{PEnc}(G^t_{h,i,j}, pk)$, Send $C_{G^t_{h,i,j}}$ to the server;
(4) The server receives the $C_{q_{ij}}$ sent by each client, update the encryption item matrix $C_{q_{ij}} = (1 - 2\gamma)C_{q_{ij}} + \sum_{k=1}^{d}C_{q_{k}j}$. Repeat steps (2), (3), (4) until the loss function (3) converges, and achieve the entire training process.

In the federated matrix factorization recommendation, how to fully mine the user’s interest preference from the user’s rating of the item is a key issue of the algorithm model. In actual scenes, some users have stricter ratings of items and higher requirements for the quality of products, and the ratings are lower than other more forgiving users. Such individual differences in user rating standards seriously affect the accuracy of the matrix factorization training model to predict ratings. Paterek et al. [19] added user and item deviations in matrix factorization to more accurately mine the interaction between users and items. Koren et al. proposed BiasMF [10], which improved the performance of the basic matrix factorization model by introducing deviation (bias) terms between users and items. According to the above literature, this paper proposes a formula for adding user bias and item bias to correct the forecast rating in the matrix factorization process.
\[ \hat{r}_{ij} = p \cdot q^T + b_i + b_j \]  

(4)

Where \( b_i \) represents the bias of user \( i \), and \( b_j \) represents the bias of item \( j \).

2.3. Rating conformity
High-quality rating data of user items can also help federated recommendation algorithms mine users’ interest preferences and establish user interest models more accurately. Literature[18] believes that each user’s preference for items generally satisfies a fixed range which is called a rating trend. Abnormal ratings that deviate from the rating trend will affect the accuracy of the predicted ratings. All rating data in the traditional federated matrix factorization model are treated equally, each rating has the same weight. If the user ratings deviating from his rating trend in certain specific environments (misoperation or emotional fluctuations, etc.), the rating continues to maintain the same weight as other ratings, it will reduce the accuracy of the federated matrix factorization model for user feature learning. Therefore, it is necessary to further mine user and item interaction data, improve the quality of rating data, and reduce the impact of abnormal points in rating data. This considers whether the mobile rating data in the federated matrix factorization model meets the user rating trend. This paper defines the degree of compliance with the user rating trend as conformity. According to the literature [18], this paper defines the user rating conformity \( \omega_{ij} \) to measure the degree of deviation of the \( i \)-th user’s rating of the \( j \)-th item from the user’s rating trend. The formula for conformity with user ratings in this paper is as follows:

\[ \omega_{ij} = 1 + \tanh(\min\left(1, \frac{1}{|r_{ij} - \omega_i + \sigma|} \right)) \]  

(5)

Where \( r_{ij} \) is the real rating, \( \omega_i \) is the average rating of user \( i \), and \( \omega_{ij} \) is the rating trend of user \( i \). \( \sigma \) is a minimum value. Here, the value \( \sigma \) is set to 0.0001 to prevent the rating value and the average value from being equal, causing the denominator to be zero. \( r_{\text{max}} \) is the highest rating value, to prevent too high conformity. \( \omega_{ij} \) has added a bias item of 1, to ensure that the lowest rating conformance is 1 to avoid the occurrence of 0. \( \tanh \) is a normalized function, to prevent the conformity difference between the ratings from being too large, and aggravating the sparsity of the data.

3. Federated matrix factorization model fusion bias and conformity
The traditional federated matrix factorization algorithm applies the idea of federated learning to the traditional matrix factorization algorithm. Under the premise that the third-party server does not collect user preference data, it can still make accurate recommendations to target users. In the whole process, the user preference data never leaves the local area, and all preference data can be shared. It plays a role in protecting the privacy of user preferences. However, these algorithms use the traditional matrix factorization model without considering the bias and conformity of the rating data. In this section, the algorithm proposed in this paper will be described in detail. The implementation process of the algorithm is divided into two stages. In the first stage, the personalized correction parameters of bias and conformance are added to the traditional matrix factorization model, and a personalized matrix factorization (PER-MF) algorithm is proposed; In the second stage, the idea of federated learning is introduced into the personalized matrix factorization algorithm, and a personalized federated matrix factorization algorithm is proposed.

3.1. Personalized Matrix Factorization
Because the matrix factorization algorithm does not fully mine the user's interest preferences and the conformity of the rating data, it is not suitable for actual recommendation scenarios. In this section,
this paper adds the proposed personalized correction parameters to the loss function of the matrix factorization model based on the user's predicted rating and the conformity of the rating data to obtain the loss function of PER-MF algorithm.

\[ e_{ij}^2 = \min_{p,q} \sum_{i=1}^{m} \sum_{j=1}^{n} (r_{ij} - p \cdot q^T - b_i - b_j)^2 + \frac{\beta}{2} (w_1 \| p \|_2^2 + \| q \|_2^2) \]  

(6)

The steps of the proposed PER-MF algorithm are as follows:
1. Calculate the average rating of each user called \( \omega_i \) based on all user preference data;
2. According to the user's average rating, using formula (5) to obtain the conformity \( \omega_{r_{ij}} \) of each rating data;
3. User preference data is transformed into user-item rating matrix \( r \);
4. Randomly initialize user matrix \( p \), project matrix \( q \), user bias \( b_i \), project bias \( b_j \);
5. According to formula (4) modeling, the prediction rating of model training is obtained;
6. Calculating the error between the predicted rating \( \hat{r}_{ij} \) and the real rating \( r_{ij} \) according to the loss function (6);
7. Experiment with the stochastic gradient descent method to update the element values in \( p \) and \( q \) and the element values in \( b_i \) and \( b_j \);
8. Repeat steps (5) and (6) until the loss function converges.

After the server collects user preference data and uses PER-MF algorithm to recommend products to users, there is a risk of leaking user preference data. How to share the preference data of multiple mobile terminals without the preference data leaving the mobile. At the same time, the PER-MF algorithm is used for recommendation by joint multi-party mobile terminals and servers, so the PER-FedMF algorithm is further proposed.

### 3.2. Personalized Federation Matrix Factorization

Based on the PER-MF algorithm, through the introduction of the idea of federated learning, when the third-party server does not collect user preference information, the mobile can make accurate product recommendations to users by using the PER-MF algorithm. Therefore, the PER-FedMF algorithm proposed in this paper is obtained. The data source of the algorithm is distributed, rather than stored on a third-party server. The data source of the algorithm is distributed, rather than stored on a third-party server. Interaction data (ratings) between users and items and user personal data are only available on mobile, while project features are stored and shared on a third-party server. The algorithm framework is proposed for the federated matrix factorization of fusion bias and conformity personalized parameters, which is suitable for most recommendation fields. After the introduction of federated learning, it is necessary to jointly train a personalized matrix factorization model on the mobiles of all parties and the server. Therefore, based on formula (6), the loss function of this algorithm is obtained through federated learning:

\[ e_{ij}^2 = \min_{p,q} \sum_{k=1}^{h} \sum_{i,j=1}^{n} a_{ij} \left( r_{ij}^k - p^k \cdot q^k - b_i - b_j \right)^2 + \lambda_1 \| q \|_2^2 + \mu_i \| b_i \|_2^2 + \mu_j \| b_j \|_2^2 \]

\[ + \lambda_2 \sum_{k=1}^{h} \left( \| p^k \|_2^2 + \mu_i \| b_i^k \|_2^2 \right) \]

(7)

Where \( \lambda_1, \lambda_2, \mu_i, \mu_j \) represent regularization parameters, \( b_i \) represents the bias term of the user \( i \) whose rating is corrected on the \( h \)-th mobile, and \( b_j \) represents the bias term of the global corrected rating. \( \lambda_1 \| q \|_2^2 \) represents the regularization item of the global item matrix, \( \mu_i \| b_i \|_2^2 \) represents the
regularization item of the global item bias $b_p$ and $\sum_{h=1}^{H} (\lambda_p \| p_h \|_2^2 + \mu_p \| b_p \|_2^2)$ represents the regularization item of the user matrix and user bias of the h-th mobile. The regularization term is to prevent the model from overfitting. Compared with the traditional FedMF algorithm, how to update the personalized correction parameters of the user, item bias, and rating data conformity in the PER-FedMF algorithm proposed is one of the main contributions of this paper.

The algorithm in this paper uses the traditional stochastic gradient descent method [15] to train $p, q, b, b_j$ on each mobile and server. The generation and distribution of public and private keys are the same as the FedMF algorithm model. The framework of the algorithm in this paper is shown in Figure 4, and the pseudo-code is shown in Algorithm 1. The specific steps of the algorithm are as follows:

1. According to the modeling of the rating conformity in section 2.3, the rating dataset of the mobile is preprocessed to obtain the rating trend $\omega_i$ of user $i$ and the conformity of the rating data $\omega_{h,i}$ and update the dataset of each mobile as Ratings.

2. Initialize the parameters, including the user bias $b_i$ on the mobile and the item bias $b_j$ on the server. The server uses the public key $pk$ to encrypt $b_j$ and obtains the encrypted project offset $C_h = PEnc(b_j, pk)$, the initialization of user matrix $p$ and item matrix $q$ is the same as FedMF algorithm.

3. The server provides the latest encrypted $C_h$ matrix and $C_{ij}$ vector for all mobiles to download;

4. The mobile terminal downloads the encrypted $C_h$ vector from the server and decrypts it to obtain $b_j$. The download and decryption of the matrix are consistent with FedMF.

(i) The mobile obtains the derivation of $b_i$ in the loss function to obtain the gradient $\nabla_{b_i} = 2\omega_{i,j}\sum_{j \neq i, j \neq i} q_{i,j}^T (r_{i,j} - \hat{r}_{i,j})$ of the vector $b_i$, the formula of $\hat{r}_{i,j}$ is: $\hat{r}_{i,j} = p^k \cdot q_j^T + b_i + b_j$;

(ii) The mobile obtains the derivation of $b_j$ in the loss function to obtain the gradient $\nabla_{b_j} = 2\omega_{i,j}\sum_{j \neq i, j \neq j} (r_{i,j} - \hat{r}_{i,j})$ of $b_j$, and the update of $b_i$ is: $b_i^{h+1} = b_i^{h} - \gamma \nabla_{b_i}$;

(iii) Derivation of loss function q to obtain the gradient $G_{k,j}^{r_i} = 2\omega_{k,j} \sum_{(j,i) \neq i} p_{i,j}^k (r_{i,j} - \hat{r}_{i,j})$ of vector $q_j$, The bi in the loss function is derived to obtain the gradient $G_{k,j}^{b_i} = 2\omega_{k,j} \sum_{(j,i) \neq i} (r_{i,j} - \hat{r}_{i,j})$ of the vector $b_i$. Where $p_{i,j}^k$ represents the i-th row of the h-th mobile user matrix;

(iv) Encrypt gradient $G_{k,j}^{r_i}$ and gradient $G_{k,j}^{b_i}$ to get $C_{k,j}^{r_i} = PEnc(G_{k,j}^{r_i}, pk)$ and $C_{k,j}^{b_i} = PEnc(G_{k,j}^{b_i}, pk)$. Send $C_{k,j}^{r_i}$ and $C_{k,j}^{b_i}$ to the server;

(v) After the server receives A and B from each mobile Update the encrypted item matrix $C_{k,j} = (1 - 2\gamma \lambda)C_{k,j} + \sum_{h=1}^{H} C_{h,j}$ and the encrypted item bias $C_{k,j} = (1 - 2\gamma \lambda)C_{k,j} + \sum_{h=1}^{H} C_{h,j}$. Repeat steps (3), (4), (5) until the loss function (7) converges, and the entire training process will be achieved.

Algorithm 1 PER-FedMF: Personalized Federated Matrix Factorization

| Input: $r, \omega_i, \omega_{h,i}$ |
| Output: $\hat{r}$ |
| **FL Mobile** |

1. Mobile initializes $p, b_j$
2: while True do
3: Download latest $p, b$
4: Compute $p_i$ using $p_i^{h,t} = p_i^{h,t-1} - \nabla p_i$
5: Compute $b_j$ using $b_j^{h,t} = b_j^{h,t-1} - \nabla b_j$
6: Compute $q$ gradients $G_{k,j}^{t-1}$ using $G_{k,j}^{t-1} = 2\omega_{q,j} \sum_{i(i,j)\in D_k} p_i^{h,t-1} (r_{ij} - \hat{r}_{ij}^q)$
7: Compute $b_j$ gradients $\tilde{G}_{k,j}^{t-1}$ using $\tilde{G}_{k,j}^{t-1} = 2\omega_{b,j} \sum_{j(i,j)\in D_k} (r_{ij} - \hat{r}_{ij}^b)$
8: Encrypt $G_{k,j}^{t-1}$ using $C_{G_{k,j}^{t-1}} = PEnc(G_{k,j}^{t-1}, pk)$
9: Encrypt $\tilde{G}_{k,j}^{t-1}$ using $\tilde{C}_{G_{k,j}^{t-1}} = PEnc(\tilde{G}_{k,j}^{t-1}, pk)$
10: Transmit $C_{G_{k,j}^{t-1}}$ and $\tilde{C}_{G_{k,j}^{t-1}}$ → FL Server
11: end while

FL Server
1: Server initializes $q, b$
2: while True do
3: Receive $C_{G_{k,j}^{t-1}}$ and $\tilde{C}_{G_{k,j}^{t-1}}$
4: Compute $q$ using $C_{q,j} = (1 - 2\lambda)C_{q,j} + \sum_{h=1}^{\hat{r}_{ij}} C_{G_{k,j}^{t-1}}$
5: Compute $b_j$ using $C_{b_j} = (1 - 2\lambda)C_{b_j} + \sum_{k=1}^{\hat{r}_{ij}} C_{G_{k,j}^{t-1}}$
6: Transmit $C_{q,j}$ and $C_{b_j}$ → FL Mobile
7: end while

According to the training process of PER-FedMF algorithm, during the execution of the algorithm, only the model parameters are alternately updated between the mobiles and the server. The user's original rating information does not leave the mobile, but the rating information can still be shared, which not only plays a role in protecting user privacy but also makes accurate recommendations. Compared with the FedMF algorithm, the algorithm in this paper considers and proves that the user, item bias and rating conformity parameters are correctly encrypted, transmitted, and updated between the mobile and the server, and it improves the accuracy of the recommendation. The rating conformity parameter is always transmitted throughout the model training process along with the square of the difference between the real rating and the predicted rating in the loss function (7); The user bias is only updated on the local mobile, the gradient of the item bias is updated on the mobile, and the update is encrypted and sent to the server, and the item bias parameters are updated on the server.

4. Experiment and analysis
This section will describe the datasets, experimental parameter settings, measurement indicators, evaluation methods, experimental results, and analysis used in the experiments in some detail in this paper.

4.1. Dataset
In this experiment, we select a representative MovieLens real movie rating dataset [21], which contains 100,000 rating information of 9724 movies produced by 610 users. Each rating ranges from 1
to 5, and each user has rated at least 20 movies. The sparsity of this dataset is 98.3%. We randomly select 17 data from each user rating data of the MovieLens dataset as the test set, and the final training set accounts for about 90% of the dataset. Where the number of ratings is 90466, and the test set accounts for about 10%. The number of ratings is 10370. To reduce the influence of randomness, each experiment was repeated 3 times, and the average value of the experiment was taken as the experimental result.

4.2. Evaluation index

Recommendation accuracy is a key indicator to measure the pros and cons of recommendation algorithms. This paper uses two common evaluation indicators, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate the accuracy of the proposed algorithm. MAE represents the average value of the absolute error between the predicted rating and the true rating. The smaller the MAE value, the higher the accuracy. It is defined as follows:

$$MAE = \frac{1}{n} \sum_{i,j} |\hat{r}_{ij} - r_{ij}|$$  \hspace{1cm} (8)

RMSE represents the square root of the partial variance between the predicted rating and the true rating and the ratio of the number of predictions n. RMSE measures the accuracy of the recommendation system more harshly. The smaller the RMSE, the higher the accuracy. It is defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i,j} (\hat{r}_{ij} - r_{ij})^2}$$  \hspace{1cm} (9)

Where n is the number of predicted ratings.

4.2.1. Verify that the personalized parameters are updated correctly. To explore the correct update of the personalized parameters of bias and conformity in the FedMF model, specific measurement is made from the RMSE and MAE values of the training set and test set. It can be seen from Figure 5 that in the last 5 iterations of the model convergence, the RMSE value of the PER-FedMF algorithm is slightly higher than that of PER-MF. After calculating the RMSE value of the last 5 times, the average recommendation accuracy of the PER-FedMF algorithm is 0.05% lower than that of the PER-MF algorithm. It can be seen from Table 1 that the RMSE and MAE values of the PER-FedMF algorithm on the test set are slightly higher than those of PER-MF. After calculation, the recommended accuracy of PER-FedMF algorithm is 0.03% and 0.04% lower than that of PER-MF algorithm. The reason for the decrease in recommendation accuracy is that in the process of distributed training between the mobile terminals of all parties and the server, there must be some loss of accuracy when the gradients of all parties are aggregated by the server. Experiments show that the recommendation accuracy of the algorithm in this paper is far less than 0.1%, indicating that the performance of the algorithm is almost lossless, which proves that the personalized correction parameters of bias and compliance are correctly updated in the FedMF model; While protecting the privacy of user preferences, it is still possible to fully mine user preference data on the mobile.
4.2.2. Algorithm recommendation performance. To explore the influence of the personalized parameters of bias and conformity on the results of federated recommendation, the recommendation quality of the algorithm in this paper is specifically measured in terms of the number of iterations, epochs, learning rate, and hidden features.

(1) Analysis of the impact of epochs on algorithm performance:

It can be seen from Figure 6 that under the same number of iterations, PER-FedMF has a smaller RMSE value than FedMF, indicating that PER-FedMF has better performance than FedMF. Recommendation accuracy is highest when the number of iterations is small, because of the early stage of model training, user and item bias have a greater impact on the prediction rating. PER-FedMF approaches convergence faster than FedMF, and the convergence effect is better because the bias term is considered in the model training process while reducing the influence of abnormal preference data and further improving the prediction rating.

(2) Analysis of the impact of learning rate $\gamma$ on algorithm performance:
Here, the implicit feature factor $k$ is set to 10, and other parameters remain unchanged. The range of learning rate $\gamma$ is 0.0001 to 0.001, and the RMSE value is used to judge the accuracy of the PER-FedMF algorithm. It can be seen from Figure 7 that under the same learning rate $\gamma$, the RMSE value of PER-FedMF on the test set is smaller than that of FedMF, indicating that PER-FedMF has a better-recommended performance. Especially when the learning rate $\gamma$ is small, the PER-FedMF recommendation accuracy improvement effect is more obvious. This is because when the learning rate $\gamma$ is small, the model has not yet reached convergence during the 150 iterations. The user, item bias, and conformity have a greater impact on model training. With the gradual increase of the learning rate between the value ranges, the performance improvement of the algorithm reaches a stable state relative to FedMF, because the degree of correction of the predicted rating becomes stable during the model convergence process of the personalized correction parameter. In the case of a certain number of iterations, a small learning rate may prevent the model from converging to the minimum value, and a large learning rate may cause the model to float back and forth near the minimum value, neither of which can reach the optimum. Therefore, the appropriate learning rate should be selected according to the actual situation.

(3) Analysis of the impact of implicit feature $k$ on algorithm performance

Here, the learning rate $\gamma$ is set to 0.0002, and other parameters remain unchanged. The range of hidden feature $k$ is selected as 10-50, and the accuracy of PER-FedMF is judged by MAE. It can be seen from Figure 8 that under the same hidden feature $k$, the MAE value of PER-FedMF is smaller than that of FedMF, indicating that PER-FedMF has a better-recommended performance. When the hidden feature $k$ is small, the performance improvement effect of PER-FedMF recommendation is more obvious, because the model does not fully mine the preference data at this time. At this time, adding users, item bias, and conformity can mine user preference data to a greater extent. As the value of $K$ increases, the degree of performance improvement of PER-FedMF is gradually decreasing, because as the value of $k$ increases, FedMF's mining of user preference data becomes more and more sufficient, and the degree of correction of the prediction rating by the personalized correction parameter is reduced, but within the selected $k$ value range, the performance of the PER-FedMF algorithm is always optimal.

After the above-mentioned experimental results and analysis, this paper proves that the personalized correction parameters of the user, item bias, and conformity are correctly updated in the FedMF model. It is verified that the user preference data does not leave the mobile, and the PER-FedMF model with almost lossless recommendation accuracy can still be trained; Compared with the traditional FedMF algorithm, the recommendation performance of the PER-FedMF algorithm in this paper is better and is more suitable for real recommendation scene.

![Figure 7. The impact of learning rate $\gamma$ on performance.](image-url)
5. Conclusions
This paper proposes a mobile personalized recommendation algorithm PER-FedMF based on federated matrix factorization. The algorithm introduces the personalized correction parameters of the user, item bias, and rating conformity into the federated matrix factorization model, reconstructs the predictive rating model, and distinguishes the quality of rating data. Under the condition that the user rating data on the mobile does not leave the local area, the relationship between users and items is fully and accurately mined. Through experiments on the benchmark movie dataset, it is shown that PER-FedMF is better than the current FedMF model in recommendation accuracy and is more suitable for real recommendation scenarios. But in the process of PER-FedMF model training, the time efficiency is low. How to improve the time efficiency of PER-FedMF model will be a direction for our further research in the future.

Acknowledgments
This work was financially supported by The Higher Education Innovation Ability Enhancement Project of Gansu Province (2019A-006) fund. Junjie Jia’s Email: 654205992@qq.com. Zhipeng Lei’s Email: 512861984@qq.com.

References
[1] Ricci, F., Rokach, L., Shapira, B., Recommender systems: introduction and challenges, J. (eds) Recommender Systems Handbook. Springer, Boston, MA, 2015, pp: 1-34.
[2] Wang Wei, and Zhou Gang. A collaborative filtering recommendation algorithm based on improved Bhattacharyya coefficient, J/OL. Computer Application Research: 1-4, https://doi.org/10.19734/j.issn.1001-3695.2019.08.0521, 2020.
[3] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh. Privacy-preserving matrix factorization, C. Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS’13), Association for Computing Machinery, New York, NY, USA, 2013: 801-812.
[4] J. Canny. Collaborative filtering with privacy, C. Proceedings 2002 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 2002: 45-47.
[5] Arnaud Berliz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. Applying differential privacy to matrix factorization, C. In Proceedings of the 9th ACM Conference on Recommender Systems (RecSys ’15), Association for Computing Machinery, New York, NY, USA, 2015: 107-114.
[6] Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the Netflix Prize contenders, C. In Proceedings of the 15th ACM SIGKDD
international conference on Knowledge discovery and data mining (KDD '09), Association for Computing Machinery, New York, NY, USA, 2009: 627-636.

[7] Jakub Konecný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence, OL. http://arxiv.org/abs/1610.02527, 2016.

[8] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data, C. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR 54, 2017: 1272-1282.

[9] Muhammad Ammad, Elena Ivannikova, Suleiman A Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. Federated collaborative filtering for privacy-preserving personalized recommendation system, OL. http://arxiv.org/abs/1901.09888, 2019.

[10] Koren Y, Bell R, and Volinsky C. Matrix Factorization Techniques for Recommender Systems, J. Computer, 2009, 42 (8): 30-37.

[11] Steffen Rendle. Factorization Machines with libFM, J. ACM Trans. Intell. Syst. Technol, 2012, 3 (3): 1-22.

[12] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure Federated Matrix Factorization, OL. http://arxiv.org/abs/1906.05108, 2019.

[13] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang and Xing Xie. Privacy-Preserving News Recommendation Model Training via Federated Learning, OL. https://arxiv.org/abs/2003.09592, 2020.

[14] Adrian Flanagan, Were Oyomno, Alexander Grigorievskiy, Kuan Eeik Tan, Suleiman A. Khan, and Muhammad Ammad-Ud-Din. Federated Multi-view Matrix Factorization for Personalized Recommendations, OL. https://arxiv.org/abs/2004.04256, 2020.

[15] Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent, C. Proceedings of COMPSTAT'2010, Physica-Verlag HD, 2010: 177-186.

[16] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated Machine Learning: Concept and Applications, J. ACM Trans. Intell. Syst. Technol, 2019, 10 (2): 2157-6904.

[17] Funk, S. Netflix Update: Try This at Home, OL. http://sifter.org/simon/journal/2006121-1.html, 2006.

[18] Wu Z, Tian H, Zhu X, Wang S. Optimization Maxtrix Factorization Recommendation Algorithm Based on Rating Centrality, C. the 3th International Conference on Data Mining & Big Data. Springer, cham, 2018.

[19] Paterek, A. Improving regularized singular value decomposition for collaborative filtering, C. Proceedings of KDD cup and workshop, 2007.

[20] Peng Hongwei, Jin Yuanyuan, Lv Xiaoying, and Wang Xiaoling. A context-aware POI recommendation algorithm based on matrix factorization, J. Chinese Journal of Computers, 2019, 42 (08): 1797-1811.

[21] F. Maxwell Harper and Joseph A. Konstan. The MovieLens Datasets: History and Context, J. ACM Trans. Interact. Intell. Syst, 2016, 5 (4): 1-19.

[22] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes, C. International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, 1999: 223-238.

[23] YANG Q. Federated recommendation, EB/OL.2019. https://cn.fedai.org/research/conferences/the-28th-international-joint-conferences/the-28th-international-joint-conference-on-artificial-intelligence-ijcai-2019.

[24] CHAI D. Vertical secure federated matrix factorization, R. Hong Kong, China, 2019.