Effective Matrix Factorization for Recommendation with Local Differential Privacy*

Hao Zhou[1,2], Geng Yang[1,2], Yahong Xu[1,2], and Weiya Wang[1,2]

[1] College of Computer Science and Software, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu 210023, China
[2] Big Data Security and Intelligent Processing Lab, Nanjing 210003, China

Abstract. With the continuous upgrading of smart devices, people are using smartphones more and more frequently. People not only browse the information they need on the Internet, but also more and more people get daily necessities through online shopping. Faced with a variety of recommendation system, it becomes more and more difficult for people to keep their privacy from being collected while using them. Therefore, ensuring the privacy security of users when they use recommendation system is increasingly becoming the focus of people. This paper summarizes the related technologies. A recommendation algorithm based on collaborative filtering, matrix factorization as well as the randomized response is proposed, which satisfies local differential privacy(LDP). Besides, this paper also discusses the key technologies used in privacy protection in recommendation system.In addition, This paper includes the algorithm flow of recommendation system. Finally, the experiment proves that our algorithm has higher accuracy while guaranteeing user privacy.

Keywords: Local differential privacy · Matrix factorization · Recommender system · Randomized Response.

1 Introduction

Faced with a huge amount of item information, it is more and more difficult for people to find what they are interested in and need. The advent of the recommendation system reduces the difficulty for users to choose. Recommendation system can not only locate items with high correlation quickly according to the information provided by users but also make an additional recommendation in other aspects. To provide better user experience, recommendation system usually

* the National Natural Science Foundation of China (61572263, 61502251, 61602263, 61872197), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX18 0891), the Natural Science Foundation of Jiangsu Province (BK20161516, BK20160916), the Postdoctoral Science Foundation Project of China (2016M601859), the Natural Research Foundation of Nanjing University of Posts and Telecommunications (NY217119).

collects relevant information of users actively and get users' profiles by mining relevant information[19]. This kind of collecting and analyzing behavior has the potential risk of privacy leakage. On the one hand, the recommendation system may actively make use of user information for profit; on the other hand, the user data maintained by the recommendation system also has the risk of being attacked and leaked.

When people use the recommendation system, the recommendation system can accurately predict users' privacy information by mining users' relevant information[20]. Many existing studies have found that recommendation systems can obtain users' relevant privacy information by mining user information. For example, the location-based recommendation system can judge the user's residence, company, travel frequency, and even the user's travel purpose by locating the user's location and the time spent in each location. Although there are already methods for protecting all the items and scoring information for each user, this method requires the user to implement large matrix multiplication locally, so that the space complexity and time complexity of the implementation will increase with the total items linearly. As a result, it is clearly difficult for users to achieve in their local equipment. Our goal is to improve the accuracy and speed of recommendation while reducing time and space consumption to improve the practicability of the recommendation system based on protecting users' privacy information.

The purpose of this paper is to ensure the recommendation quality of recommendation while satisfying the LDP and reducing the space and time cost. In this paper, a new gradient descent matrix factorization algorithm that satisfies differential privacy is proposed. We use the LDP solution proposed by Nguyen to protect the private data of each user[6]. We also use the LDP solution proposed by Qin et al[3] to reduce the number of related items, which greatly reduces the disturbance error caused by a large number of items.

Our main contributions of this paper are as follows. First, we proposed a new algorithm that protects all user items and scoring data. Secondly, we have greatly reduced the number of items related to users, and thus significantly reduce the relevant dimensions. As a result, space and time consumption of the algorithm is significantly reduced. Thirdly, by adding a new gradient correction parameter, the quality of the recommendation system has been significantly improved. To sum up, our algorithm guarantees high recommendation quality and needs less space and time costs while protecting items and ratings.

2 Preliminary

2.1 LDP

Local differential privacy is a popular privacy protection method based on ϵ-differential privacy. This means that before sending personal privacy data to the data collector, the user first disturbs the data locally to satisfy the ϵ-differential privacy[16], and then uploads the disturbed information to the data collector.

Generally, if a randomized algorithm f satisfies ϵ-differential privacy, f satisfies the following inequalities for any two neighboring datasets D And D' and any possible output o:

$$Pr[f(D) \in o] \leq e^{\epsilon} \times Pr[f(D') \in S] \tag{1}$$

Generally, it is difficult for an opponent to be confident that the output o obtained by using algorithm f comes from either D or D'[9]. So, for the individual user, using this way to process the data, when the data is abused, he can plausibly deny. Among them, ϵ is also called the privacy budget[8], which is mainly used to control the strength of privacy protection. When the privacy budget is very small, it means stronger privacy protection. In this paper, we consider two data sets, in which each recorded information is a pair of items that the user has scored. However, Shin et al[1], Hua et al[2], and Shen et al[9] all consider that both neighboring data sets D and D' contain all item information, even if the user has not scored the corresponding item[3]. Different definitions of adjacent data sets will lead to different time and space costs and affect the final accuracy. Our algorithm is more realistic for it only considers the related items and has lower space, time consumption as well as higher accuracy while satisfying the privacy protection for each user.

One of the important concepts of differential privacy is sequential combinability[3], that is, for all algorithm f_i of a series of randomized algorithms, they satisfy ϵ_i-differential privacy, then for the whole sequence of f_i, it satisfies $\sum_{i=1} \epsilon_i$-differential privacy. Early studies[7] have shown that for a given privacy budget ϵ, users can divide ϵ into several parts, and each part can release perturbation information[6], the overall process is to satisfy the ϵ-differential privacy.

2.2 Existing LDP Solution

Because LDP only requires locally perturbed of privacy data to satisfy the ϵ-differential privacy. Therefore, theoretically, every user can apply LDP to their private information. However, in reality, it is often the data collector who can obtain the user's real data, and only disturbs the user's private information when it is released. If the user uses LDP to disturb his information locally and then publishes the proceed information, the recommendation system cannot get the real exact data of the user, therefore the final information released by the system can only be the data set of all users disturbed information[14]. This ensures privacy protection for user data. The following is a summary of the existing LDP methods, which have received extensive attention recently.

RandomizedResponse. The randomized response algorithm (RR) is based on LDP. Generally, RR refers to whether or not a user answers a judgmental question, such as 'have you eaten today?'. He can reasonably deny the answer. RR controls the intensity of privacy protection by adjusting the probability of real and false answers. That is to say, if the probability of true answers is set to be smaller, the intensity of privacy protection will be higher. To use RR in LDP, we set the probability of getting the true value to p[15]. According to

the existing works, when p satisfies the following values, the RR satisfies the ϵ-differential privacy:

$$p = \frac{e^{\epsilon}}{1 + e^{\epsilon}} \tag{2}$$

RR can only be used to answer questions with binary answers. But it is the cornerstone of solving more complex problems.

RAPPOR. RAPPOR can deal with more complex questions, especially when the answer is non-binary. RAPPOR is mainly used to estimate the frequency of items. Generally, suppose there are n users and m item categories, and each user u_i has an item v_j. The purpose of the data collector is to collect the occurrence frequency of each item category. In RAPPOR, user u_i represents the ownership of the item v_j by uploading a vector of m bit length. In this m bit vector, all the bits except v_j-th are zero. Then, the user uses RR independently for each bit in the m-bit vector[3]. The specific values will be described below. Data collectors collect vectors sent by all users to calculate unbiased estimates of the occurrence of each item in m item categories.

To get the value of p, the concept of sensitivity in differential privacy needs to be used in RAPPOR. Generally speaking, for any function F, sensitivity Δ_f can be defined as

$$\Delta_f = max_{DD'} \left\| F(D) - f(D') \right\|_1 \tag{3}$$

D' and D are adjacent data sets, and $\|\bullet\|$ denotes ζ_1 of a vector. Since there is only 1 bit which value is one and all the others are zero, the maximum value of $\left\| F(D) - f(D') \right\|_1$ is 2. Therefore, the sensitivity is 2. According to the existing papers, when p satisfies the following values, RAPPOR can satisfy ϵ-differential privacy:

$$p = \frac{e^{\frac{\epsilon}{2l}}}{e^{\frac{\epsilon}{2l}} + 1} \tag{4}$$

Generally, l is much smaller than m. In the existing recommendation system, the gradient matrix of all items is uploaded. As a result, the maximum ζ_1 is $2m$ for the vector subtraction of two m bits and the sensitivity is $2m$. This paper assumes that users own l items. By using the RAPPOR method for l items owned by users, we can protect the related items owned by users to satisfy ϵ-differential privacy, while greatly reducing the sensitivity. Besides, the space complexity and time complexity of the algorithm are significantly reduced.

2.3 Matrix Factorization

This paper is based on the standard collaborative filtering recommendation algorithm. We assume that there are n users scoring m items (movies, etc.). We assume that the matrix is $R^{n \times m}$, and the element r_{ij} whose value is not zero indicates that the user i scored the item j[10]. Let the set of valid scoring subscripts $D \subset \{1, \ldots, n\} \times \{1, \ldots, m\}$, which represents the user/item pair. Then the total number of valid scores M can be expressed by $|D|$[11]. The scoring matrix is generally very sparse, which results in M being much smaller than nm, especially when both n and m are large.

With a given training set, the recommendation system can predict scores of items that have not been scored by the user. Matrix factorization is one of the most popular methods for prediction because of its high accuracy and easy implementation. In the matrix factorization, each user is represented by a d-dimensional vector, also called a personal profile[1]. At the same time, each item is also represented by a d-dimensional vector, also known as the profile of the item. Then, the relevance of the item to the user can be represented by the inner product of the two vectors[12]. Thus, for users and items, the corresponding hidden factor vector forms are $U = u_{\{1:n\}}$ and $V = v_{\{1:m\}}$[13], respectively. In this paper, the user profile is represented by u_i, $u_i \in R^d, 1 \leq i \leq n$, and the profile of the item is represented by $v_j \in R^d$,$1 \leq j \leq m$ and their values are solved by minimizing the regularized mean square error

$$arg \quad min = \frac{1}{M} \sum_{r_{ij} \in D} (r_{ij} - u_i^T v_j)^2 + \lambda_u \sum_{i-1}^{n} \|u_i\|^2 + \lambda_v \sum_{j=1}^{n} \|v_j\|^2 \tag{5}$$

Where $\frac{1}{M} \sum_{r_{ij} \in D} (r_{ij} - u_i^T v_j)^2$ is a loss function that measures the distance between two matrices, and $\lambda_u \sum_{i-1}^{n} \|u_i\|^2 + \lambda_v \sum_{j=1}^{n} \|v_j\|^2$ is a regular factor used to constrain parameters to avoid overfitting, Where λ_u ,λ_v are normal numbers. The obtained u_i and v_j can predict the relevance of the unrated item to the user by calculating their inner product $u_i^T \times v_j$.

We use the stochastic gradient descent (SGD) to minimize the formula (5). Using the SGD, U and V can be calculated by using u_i and v_j as follows:

$$v_j^t = v_j^{t-1} - \gamma_t \{\bigtriangledown_{v_j} + 2\lambda_v v_j^{t-1}\} \tag{6}$$

$$u_j^t = u_i^{t-1} - \gamma_t (\bigtriangledown_{u_i} + 2\lambda_u u_i^{t-1}\} \tag{7}$$

Where u_j^t and v_j^t are the value of u_i and v_j at t iterations, γ_t is a positive number, which represents the learning rate when the number of iterations is t. $\bigtriangledown_{u_i}$,$\bigtriangledown_{v_j}$ are gradients of u_i and v_j, respectively, which can be obtained from the following equation:

$$\bigtriangledown_{u_i} = -\frac{2}{M} \sum_{r_{ij} \in R} y_{ij} v_j (r_{ij} - u_i^T v_j) \tag{8}$$

$$\bigtriangledown_{v_j} = -\frac{2}{M} \sum_{r_{ij} \in R} y_{ij} u_i (r_{ij} - u_i^T v_j) \tag{9}$$

Since each user does not rate all items, in reality, the number of rated items is much smaller than the total items. Therefore, in this paper, the user first filters out the possible types of graded items, and then only updates the related items' gradients locally, while ignoring the others.

3 A New Differentially Private Matrix Factorization Algorithm

3.1 System Model

The experimental environment of this paper is assumed to be that the recommendation system is untrusted, and the user does not want his private information to be obtained by the recommendation system. Our goal is to allow the system and individual users to get more accurate recommendations using smaller space, time and communication costs.

Figure 1 shows how our system works. In our system, we first perturb each user's scored items to satisfy ϵ_1-differential privacy. Then upload the disturbed data to the recommendation system. Because l is much smaller than m, the sensitivity of the gradient matrix is much less than $2m$, and the communication cost will be greatly reduced. After obtaining the relevant item, the user calculates the gradient matrix $\bigtriangledown_V$ of 1 items. Then, using the dimensionality reduction method proposed by Shin et al[1], the object gradient matrix $\bigtriangledown_V$ is projected into the low-dimensional space to further compress the communication cost. After perturbed the data is sent to the recommendation system. Then, the server updates V by calculating the item gradient matrix of all users and sends the updated V to each user. In this way, after k iteration, the user can obtain the correlation between items and himself by calculating $u_i^T \times v_j$.

Since M cannot be obtained locally, the approximate calculation is generally performed by replacing M with the number of users n. The number of iterations k, the number of users n, the privacy budget ϵ, the regular coefficients λ_v and λ_u, as well as the learning rate γ_t are all given by the recommendation system.

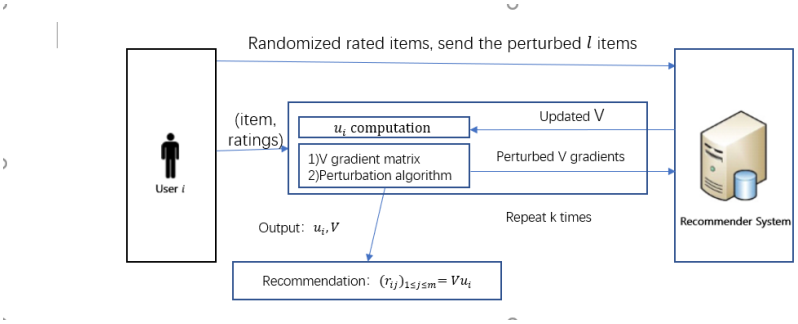

Fig. 1. Overview of Our Recommender System.

3.2 A New Solution for Protecting Items and Ratings

Since a user always wants to upload data about all items, and the total number of items m is generally very large[1], the sensitivity of V is very large. However,

the number of items related to the user is much smaller than the total number of items. Therefore, we assume the number of rated items is l rather than m and ensure the rated items which satisfy ϵ_1-differential privacy. Then the randomized gradient of the l items is uploaded to the server. Thus, the communication cost changes from $O(m)$ to $O(l)$, noting $l \ll m$.

In the algorithm private-GD-DR, we first assume that the number of each user's related item is l. If the number of items actually scored by the user is greater than l, we select l items randomly from the scored items; otherwise, we select one item from the unscored items until the total number is up to l[3]. Each user u_i selected item is represented by a vector l_i of length l bits, the value of one item being scored is 1, the value of the unrated value is 0, and the item number of the corresponding position is recorded by the vector m_i. Then, each bit in the vector l_i is perturbed using the RAPPOR method to obtain l_i^*, and the corresponding item number of the element having a value of 0 in the perturbed vector is replaced with the remaining unattached item label. The resulting m_i^* after the disturbance is sent to the server. In the following paper, m_i' is a complement of m_i.

Theorem 1. *The rating item selection described above satisfies ϵ_1-differential privacy.*

Proof. It is assumed that l_{i1} and l_{i2} are two sets of binary items, and they are all equal in length l. Assume that F is the perturbation method for the item to be uploaded in the algorithm private-GD-RAPPOR, and any possible output of F is l_i. We have

$$\begin{aligned}
\frac{\Pr\left[F\left(l_{i1}\right)=l_i\right]}{\Pr\left[F\left(l_{i2}\right)=l_i\right]} &\leq \frac{max_{l_{i1}}\Pr\left[F\left(l_{i1}\right)=l_i\right]}{max_{l_{i2}}\Pr\left[F\left(l_{i2}\right)=l_i\right]} \\
&= \frac{\prod_{j=1}^{l}\frac{e^{\frac{\varepsilon_1}{2l}}}{e^{\frac{\varepsilon_1}{2l}}+1}}{\prod_{j=1}^{l}\frac{1}{e^{\frac{\varepsilon_1}{2l}}+1}} \\
&\leq e^{\frac{\varepsilon_1}{2l}*l} \\
&< e^{\varepsilon_1}
\end{aligned}$$

Therefore, the theorem is proved.

3.3 Accuracy Improvement via Dimension Reduction

After one user u_i gets the m_i^*, he uses m_i^* to calculate the ∇_V. At the same time, ∇_U is calculated using all the related items. As a result, we can further reduce the dimension of user data.

Let $q \ll l$, let Φ be a $q * l$ random matrix whose element φ_{kj} is a Bernoulli distribution with mean 0 and variance $\frac{1}{q}$. Φ is shared between the user and the recommendation system, and the user i does not upload the item gradient ∇_V^i

Algorithm 1 Private-GD-DR

Require: l_i, m_i, m'_i,1 =< i <= n,positive integer q, predefined iteration number k, and privacy parameter ϵ, rated set D.

Ensure: Item profile matrix $V \in R^{m\times d}$

Generate a $q\times l$ random matrix Φ whose entries are drawn from Gaussian distribution with mean 0 and standard deviation $\frac{1}{\sqrt{q}}$ and send Φ to users,$\nabla^{\star}_{B}$ is the pseudo inverse matrix of Φ
Initialize U, V and a counter $iter = 0$, $array\mathrm{m} \in R^{n\times l}$
for $i = 1$; $i < n$; $i++$ **do**
  for $j = 1$; $j < l$; $j++$ **do**
    Draw T *Bernoiulli* $(\frac{e^{\frac{\epsilon}{2l}}}{e^{\frac{\epsilon_1}{2l}}+1})$
    if T =1 **then**
      $m^{*}_{ij} = m_{ij}$
    else
      select p uniformly at random from $\{1, 2, ..., m-l\}$
      $m^{*}_{ij} = m'_{ip}$
    end if
  end for
  $array\mathrm{m}_i = m^{*}_{i}$
end for
while $iter \le k$ **do**
  Initialize $\nabla^{\star}_{B} \in \{0\}^{m\times d}$
  for $i = 1$; $i < n$; $i++$ **do**
    Initialize $x^{\star}_{i} \in \{0\}^{q\times d}$
    Derive $\bigtriangledown^{i}_{V} = \{-2u_i(r_{ij} - u^{T}_{i}v_j)\}_{j\in array\mathrm{m}_i,(i,j)\in D}$
    Compute $x_i = \Phi\bigtriangledown^{i}_{V}$
    Sample s uniformly at random from $\{1, 2, ..., q\}$
    Sample p uniformly at random from $\{1, 2, ..., d\}$
    If$(x_i)_{s,l} \notin [-1, 1]$, project $(x_i)_{s,l}$ onto $[-1, 1]$
    Draw T *Bernoiulli* $(\frac{(x_i)_{s,l}(e^{\frac{\epsilon_2}{k}}-1)+e^{\frac{\epsilon_2}{k}}+1}{2(e^{\frac{\epsilon_2}{k}}+1)})$
    if T =1 **then**
      $(x^{\star}_{i})_{s,p} = qd\frac{e^{\frac{\epsilon_2}{k}}+1}{e^{\frac{\epsilon_2}{k}}-1}$
    else
      $(x^{\star}_{i})_{s,p} = -qd\frac{e^{\frac{\epsilon_2}{k}}+1}{e^{\frac{\epsilon_2}{k}}-1}$
    end if
    for $j = 1$; $j < l$; $j++$ **do**
      $q = array\mathrm{m}_{[i][j]}$
      Compute $\nabla^{\star}_{B[q][p]} = \nabla^{\star}_{B[q][p]} + \nabla^{\dagger}_{B[:][l]}(x^{\star}_{i})_{s,p}$
    end for
  end for
  Compute $\nabla^{\star}_{B} = \nabla^{\star}_{B}/n$ and send $\nabla^{\star}_{B}$ to users
  $iter = iter + 1$
  Get $\bigtriangledown^{i}_{U}$ from (8),and $u_i = u_i - \gamma_t\{\bigtriangledown^{i}_{U} + 2\lambda_u u_i\}$
  $V = V - \gamma_t\{(\bigtriangledown^{\star}_{V} - \eta^{\star}_{V}) + 2\lambda_u V\}$
end while
return V.

but uploads $\nabla_B^i(\nabla_B^i = \Phi\nabla_V^i)$[4]. Before uploading the data, it is only necessary to add noise to the method proposed by Jingyu Hua et al, and finally, send $\nabla_B^{i,*}$ to the server. The server restores the sparse matrix by using a sparse recovery algorithm. The recommendation system will calculate the restored data and feedback the results to each user. In this way, after k times of iteration, the recommendation system and users can obtain the final updated V[17], and the updated u_i is obtained only by user i.

Theorem 2. *The graded item gradient update method described above satisfies ϵ_2-differential privacy.*

Proof. Suppose ∇_B^i and $\nabla_B^{i'}$ are two arbitrary gradient matrices of user i, let ∇_B^i and $\nabla_B^{i'} \in [-1,1]^{q\times l}$. Suppose M is the perturbation method for the gradient of the item in the algorithm private-GD-RAPPOR, and any possible output of M is $\nabla_B^{i',*}$. We have

$$\begin{aligned}\frac{\Pr\left[M\left(\nabla_B^{i,*}\right)=v|\nabla_B^i\right]}{\Pr\left[M\left(\nabla_B^{i,*}\right)=v|\nabla_B^{i',*}\right]} &\leq \frac{max_{l_{i1}}\Pr\left[M\left(\nabla_B^{i,*}\right)=v|\nabla_B^i\right]}{max_{l_{i2}}\Pr\left[M\left(\nabla_B^{i,*}\right)=v|\nabla_B^{i',*}\right]}\\ &= \frac{e^{\varepsilon_2/k}-1+e^{\varepsilon_2/k}+1}{-e^{\frac{\varepsilon_2}{k}}+1+e^{\frac{\varepsilon_2}{k}}+1}\\ &= e^{\varepsilon_2/k}\end{aligned}$$

Therefore, the V obtained for each iteration satisfies $\frac{\varepsilon_2}{k}$-differential privacy. So, the V obtained after k iterations satisfies ε_2 -differential privacy. Thus, the theorem is proved. It can be seen from the sequence composability that the total algorithm satisfies ε-differential privacy[3], where $\varepsilon = \varepsilon_1 + \varepsilon_2$.

Because the privacy budget $\frac{\varepsilon}{k}$ allocated for each iteration decreases linearly with the number of iterations, the deviation caused by each disturbance to increase linearly with k. To reduce the influence of the number of iterations, we add a learning correction parameter f_k of the item gradient. From the above analysis and experimental verification, when $f_k = \frac{1}{k}$, the deviation will be significantly reduced. As shown in Table 1, when the related items are reduced from m to d, the time complexity is greatly reduced.

Table 1. Time Cost at Each Iteration.

	user	server
Hua et al.[2]	$O(md)$	$O(nmd)$
Xiao et al.[1]	$O((md)\log(m))$	$O((mnd)\log(m))$
Ours	$O((ld)\log(l))$	$O((nld)\log(l))$

4 Experiment and results

The database tested in this paper is two movie data sets, which are MovieLens and LibimSeTi[1]. The MovieLens dataset contains 20M rating information from 138,493 people for 26,744 movies. The scale for this data set is from 0.5 to 5. The LibimSeTi dataset contains 135,359 ratings for 135,359 people on 26,509 movies. The other parameter settings are the same as Shin et al. We compare Shin et al, Hua et al, and our algorithm on two data sets from five aspects.

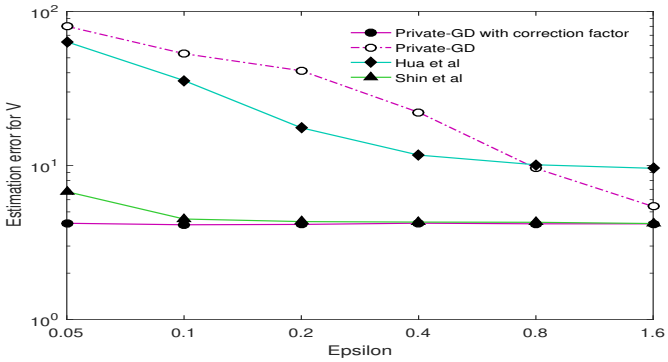

Fig. 2. On MovieLens,the estimation errors $\|V^* - V\|_{max}$ for each algorithm.

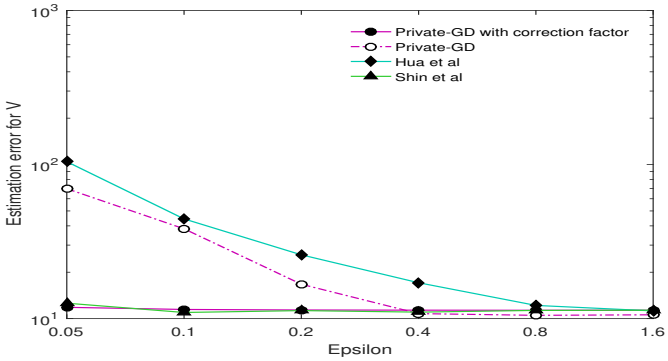

Fig. 3. On LibimSeTi,the estimation errors $\|V^* - V\|_{max}$ for each algorithm.

First, we compare the difference between the contour matrix of the items obtained by each algorithm. Figure 2 Figure 3 shows the estimated results of each algorithm after ten iterations on the MovieLens and the LibimSeTi data

sets. The estimate of Hua et al decreases linearly with an increase of ε but is significantly larger than other algorithm results. Experimental results obtained by Shin's algorithm also decreases with an increase of ε, but it is basically stable after reaching 4.19 due to the addition of the learning correction parameter $\frac{1}{k^2}$. When the modified learning parameters are not added, the estimation difference is higher than that of Shin et al lower than the Hua et al algorithm. However, when the learning correction parameter $\frac{1}{k}$ is added, our algorithm performs better when the privacy budget is lower.

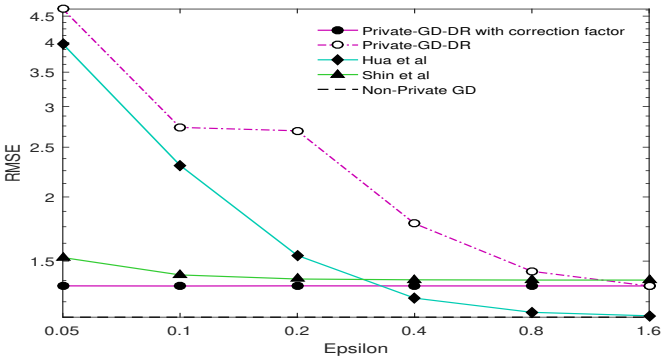

Fig. 4. On MovieLens,the prediction RMSEs for each algorithm.

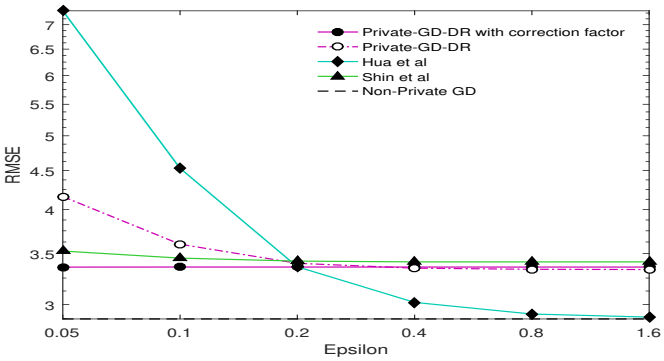

Fig. 5. On LibimSeTi,the prediction RMSEs for each algorithm..

In addition, in order to compare the mean square error of the results, we analyze the mean square error results for each algorithm after ten iterations, as shown in Figure 4 and Figure 5. For the two data sets, our algorithm is superior

to the algorithms of Hua et al and Shin et al when ε is small. At the same time, Hua et al satisfy $m\varepsilon$-differential privacy, and our algorithm satisfies ε differential privacy. Since our algorithm adds a correction parameter $\frac{1}{k}$ at each iteration, the mean square error does not increase significantly with k when k is large. Because the Libimseti data set has a larger scale, the resulting mean square error is larger than the MovieLens.

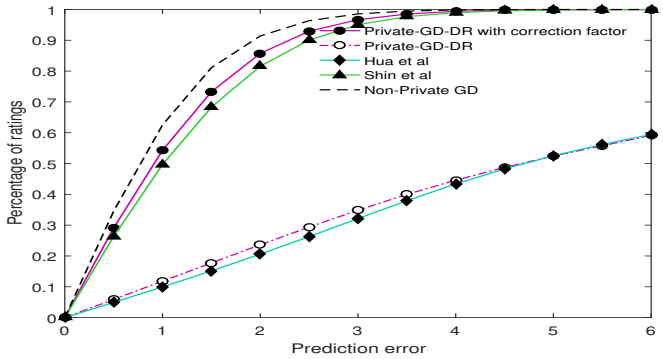

Fig. 6. On MovieLens,the prediction errors $\left|\widehat{r_{ij}} - r_{ij}\right|$ for each algorithm.

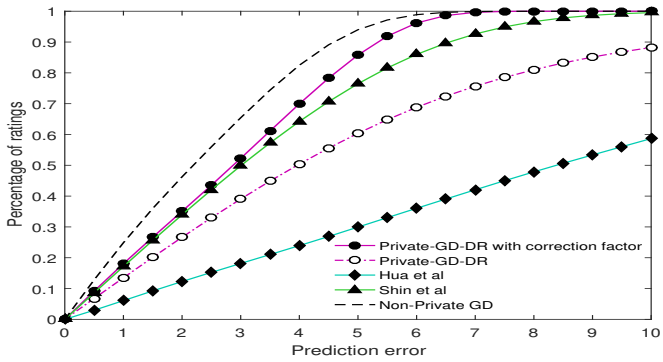

Fig. 7. On LibimSeTi,the prediction errors $\left|\widehat{r_{ij}} - r_{ij}\right|$ for each algorithm.

To test the accuracy of algorithms prediction, one data set is divided into ten parts, and their CDF (joint allocation function) is calculated separately. Nine of the ten data sets were used as training samples and the last one was used as a test sample. The prediction error $|\widehat{r_{ij}} - r_{ij}|$ of the test sample is finally calculated by 10 iterations of each training sample. $\widehat{r_{ij}}$ is the system prediction

score, r_{ij} is the true score. Figure 6 and Figure 7 show the test results of the two data sets for estimation.

To evaluate the accuracy of the recommendation, we calculate the f-score of each user's recommended top-10 items, as shown in Figures 8 and 9. In the data set MovieLens, the f-score increases with the increase of privacy budget, and when the privacy budget is small, our algorithm recommends more accuracy. In the data set LibimSeTi, when the privacy budget is small, The quality of our algorithm and Shin's algorithm recommendation are very stable, but our algorithm has higher accuracy than Shin's algorithm. The reason can be seen in Figures 4 and 5. The percentage of predicted errors in the MovieLens increases with the increase of the privacy budget faster than in the data set LibimSeTi, so the resulting f-score changes are more obvious.

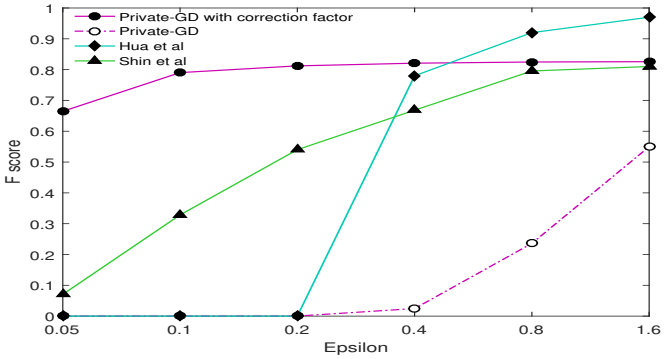

Fig. 8. On MovieLens,the F-score for each algorithm.

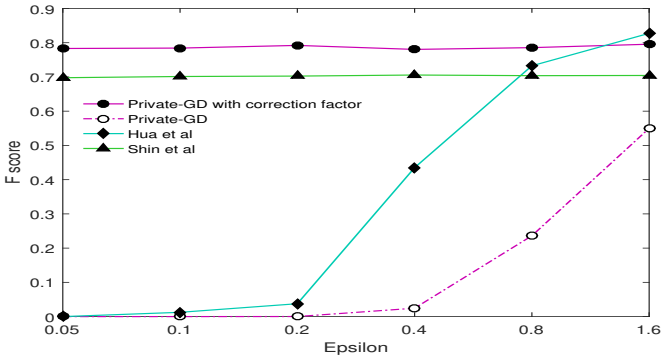

Fig. 9. On LibimSeTi,the F-score for each algorithm.

Finally, to evaluate the influence of the iteration number, it can be analyzed by calculating the RMSE of each algorithm, as shown in Figures 10. On the MovieLens dataset, the algorithms iteratively calculate 1, 2, 3, 4, 5, 10, 20, and 50 times, respectively, with a fixed privacy budget of 0.1. With the increase of the privacy budget, our algorithm with a suitable learning rate parameter of this algorithm can converge faster than other algorithms, thus the curve drops faster.

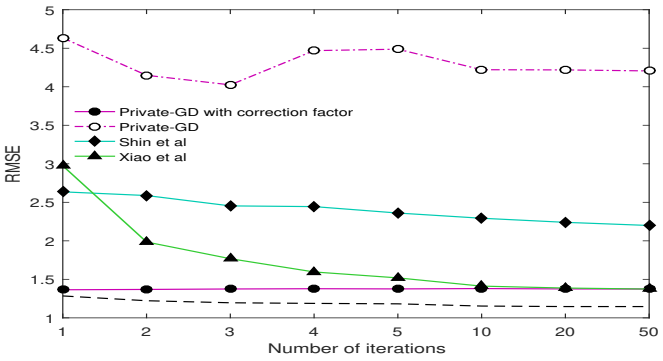

Fig. 10. On MovieLens,the prediction RMSEs of each algorithm after k iterations when ϵ=0.1.

5 Conclusion

The matrix factorization algorithm we propose sharply reduces the space and time complexity while ensuring the protection of user-related items and ratings as well as the quality of the recommendation.

References

1. H. Shin, S. Kim, J. Shin and X. Xiao, "Privacy Enhanced Matrix Factorization for Recommendation with Local Differential Privacy," in IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 9, pp. 1770-1782, 1 Sept. 2018.
2. J. Hua, C. Xia, and S. Zhong. Differentially private matrix factorization. In IJCAI, pages 1763–1770, 2015.
3. Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren. Heavy Hitter Estimation over Set-Valued Data with Local Differential Privacy. In CCS, 2016.
4. R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin. Private spatial-dataaggregationinthelocalsetting.InICDE,pages289–300, 2016.
5. G. G. Fanti, V. Pihur, and U. Erlingsson. Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. PoPETS, 3, 2016.

6. T. Nguyen, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin, Collecting and analyzing data from smart device users with local differential privacy. In IEEE ICDE arXiv:1907.00782 [cs.DB], 2019.
7. Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren. Heavy Hitter Estimation over Set-Valued Data with Local Differential Privacy. In CCS, 2016.
8. Y. Shen and H. Jin, EpicRec: Towards practical differentially private framework for personalized recommendation. In CCS, pages 180–191, 2016.
9. Y. Xin and T. Jaakkola. Controlling privacy in recommender systems. In NIPS, pages 2618–2626, 2014.
10. R. Bassily and A. D. Smith. Local, private, efficient protocols for succinct histograms. In STOC, pages 127–135, 2015.
11. G. Fanti et al. Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. PoPETS, issue 3, 2016, 2016.
12. T. Nguyen et al. Collecting and analyzing data from smart device users with local differential privacy. arXiv preprint arXiv:1606.05053, 2016.
13. X. He, A. Machanavajjhala, and B. Ding. Blowfish privacy: Tuning privacy-utility trade-offs using policies. In SIGMOD, pages 1447–1458, 2014.
14. D. Zheng and Y. Xiong, "A Unified Probabilistic Matrix Factorization Recommendation Algorithm," 2018 International Conference on Robots & Intelligent System (ICRIS), Changsha, 2018, pp. 246-249.
15. W. Liu, B. Wang and D. Wang, "Improved Latent Factor Model in Movie Recommendation System," 2018 International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, 2018, pp. 101-104.
16. H. Sun, B. Dong, H. W. Wang, T. Yu and Z. Qin, "Truth Inference on Sparse Crowdsourcing Data with Local Differential Privacy," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 488-497.
17. X. Zhao, Y. Li, Y. Yuan, X. Bi and G. Wang, "LDPart: Effective Location-Record Data Publication via Local Differential Privacy," in IEEE Access, vol. 7, pp. 31435-31445, 2019.
18. L. Lv, Z. Zhang and L. Zhang, "A Periodic Observers Synthesis Approach for LDP Systems Based on Iteration," in IEEE Access, vol. 6, pp. 8539-8546, 2018.
19. N. Li, W. Qardaji, D. Su, and J. Cao. Privbasis: Frequent itemset mining with differential privacy. PVLDB, 5(11):1340–1351, 2012.
20. F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In SIGMOD, pages 19–30, 2009.