Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1975719, 14 pages
https://doi.org/10.1155/2017/1975719

Research Article

New Collaborative Filtering Algorithms Based on SVD++ and Differential Privacy

Zhengzheng Xian,[1,2] Qiliang Li,[1] Gai Li,[3] and Lei Li[1]

[1]*School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, China*
[2]*Guangdong University of Finance, Guangzhou, Guangdong, China*
[3]*Shunde Polytechnic, Foshan, Guangdong, China*

Correspondence should be addressed to Zhengzheng Xian; xianzhengzheng@126.com

Received 28 November 2016; Revised 5 February 2017; Accepted 19 February 2017; Published 19 March 2017

Academic Editor: Kaoru Ota

Collaborative filtering technology has been widely used in the recommender system, and its implementation is supported by the large amount of real and reliable user data from the big-data era. However, with the increase of the users' information-security awareness, these data are reduced or the quality of the data becomes worse. Singular Value Decomposition (SVD) is one of the common matrix factorization methods used in collaborative filtering, which introduces the bias information of users and items and is realized by using algebraic feature extraction. The derivative model SVD++ of SVD achieves better predictive accuracy due to the addition of implicit feedback information. Differential privacy is defined very strictly and can be proved, which has become an effective measure to solve the problem of attackers indirectly deducing the personal privacy information by using background knowledge. In this paper, differential privacy is applied to the SVD++ model through three approaches: gradient perturbation, objective-function perturbation, and output perturbation. Through theoretical derivation and experimental verification, the new algorithms proposed can better protect the privacy of the original data on the basis of ensuring the predictive accuracy. In addition, an effective scheme is given that can measure the privacy protection strength and predictive accuracy, and a reasonable range for selection of the differential privacy parameter is provided.

1. Introduction

The Internet has been widely used since the birth of Web 2.0, and the human lifestyle has been greatly changed. When a user opens a shopping website or a mobile terminal application, a very enthusiastic recommender system will list some commodities in which he or she may be interested based on the purchase history record, browser footprint, evaluation information, and so forth. Today, there are numerous intelligent applications such as those. If the value of implicit feedback information such as historical browsing data, historical rating data, and the evaluation timestamp can be fully exploited, the predictive accuracy could be improved further. The Singular Value Decomposition (SVD) model [1] is a kind of common collaborative filtering method to provide personalized recommendation services, and the predictive accuracy can be improved by considering the user and item bias information. As a derivative model of SVD, the SVD++ model [2–4] achieves better recommendation accuracy by adding implicit feedback information, such as movies that a user has evaluated, and the specific value of the score does not matter for this kind of information.

While the Internet has brought much convenience to users, their daily medical, transportation, purchase, and Internet browsing information, which is neglected by the users themselves, will all be recorded to become data resources for Internet companies to identify further business opportunities and benefits. Meanwhile, there is also a risk of leakage of personal privacy information because the information is collected. In recent years, the issue of leakage of personal privacy information triggered by the Internet has arisen frequently. For example, in the Netflix Prize competition, the Netflix Corporation released a dataset through anonymous processing. However, researchers from

the University of Texas were able to deduce the real Netflix users by linking the rating and timestamp in this dataset with public information on Internet Movie Database (IMDB). As another example, in 2012, an American college student was recognized as homosexual by his roommate. His roommate used a network to search for the frequency of access to homosexual forums and websites. Collaborative filtering based on items that are related in a transaction performed by a user will lead to the increase in similarity with this user's previous commodity transactions. Thus, an attacker can track similar commodity lists related to the target user (attack target) and then determine what is a new commodity. When a similar commodity appears in these lists, the attacker can deduce the item to be added to the target user's records. Thus, what can be obtained through indirect derivation of the personal privacy information is increasingly considered.

In 2006, Dwork [5] proposed differential privacy (DP), and it can solve the issues of leakage of personal privacy information by relating to the background knowledge mentioned above. It has a very strict definition and has nothing to do with background knowledge, so it can fundamentally solve the defects of the traditional privacy protection model and is an effective way to remove the possibility of leakage of personal privacy information from the data source. Although DP has been researched for 10 years, the major research achievements are academic theories. The Apple corporation has always claimed that the user's privacy should be the top priority. This year, at the Worldwide Developers Conference (WWDC2016), Apple proposed the application of DP to collect and analyse user data from the keyboard, Spotlight, and Notes in iOS 10. Its goal is to ensure that the Quality of Service (QoS) [6] will not be affected and that the user's personal information will not be leaked. This measure opens up new pioneering work on DP in the application layer.

Today, it is quite urgent in the field of data mining to improve QoS and ensure the security of personal privacy information, eliminating users' worries and providing true and reliable data in order to guarantee the production of effective knowledge and rules [7, 8].

The contributions of our work are summarized as follows. First, we propose three new methods that apply differential privacy to SVD++ through gradient perturbation, objective-function perturbation, and output perturbation. Second, rigorous mathematical proofs are given to ensure that they all maintain the differential privacy. Third, we compare the predictive accuracies obtained by our differential privacy algorithms for SVD++ with those of the same methods for SVD and related methods in the literature on two real datasets and the method of objective perturbation for SVD++. Results show that our methods obtain better results in terms of balancing privacy and prediction. Finally, we propose a scheme for selection of DP protection parameter ε in order to balance the strength of privacy and the predictive accuracy, and a reasonable range of DP parameter ε could be obtained by this scheme.

The remainder to the paper is organized as follows. Section 2 surveys some works related to private-preserving in recommender systems. Section 3 introduces the SVD++ model and DP model. Section 4 presents the three new methods, which apply DP to SVD++ using gradient perturbation, objective-function perturbation, and output perturbation. Section 5 presents the experimental evaluation of each method on two real datasets. Finally, Section 6 summarizes the key aspects of our work and briefly addresses the directions for future work.

2. Related Work

The privacy protection of recommender systems became a popular research topic when Canny [11] proposed that the recommender not use the user's data for financial benefit in 2002. It is a hot topic in research to apply DP to personalized collaborative filtering technology since DP is considered to be the best privacy protection technology. McSherry and Mironov [12] applied DP to collaborative filtering first, and the main idea of the paper was to use the Laplace mechanism to compute a differential private item-to-item covariance matrix, which was used to find neighbours and compute the SVD recommendation. However, it seems unreasonable that there is less contribution to the covariance when a user's buying activity increases. Zhu et al. [13] addressed the privacy issues in the context of neighbourhood based CF methods by proposing a Private Neighbour Collaborative Filtering (PNCF) algorithm. Hua et al. [14] first proposed that recommenders who are not trusted should be prevented from using a user's ratings, while allowing the user to leave or join in the matrix factorization (MF) process and then realizing DP protection by disturbing the objective function of MF. Liu et al. [15] proposed a method that applied DP to Bayesian posterior sampling by Stochastic Gradient Langevin Dynamics (SGLD), thus avoiding the influence of the Gaussian noise on the whole parameter space. Zhu and Sun [16] proposed Differentially Private Item-Based Recommendation and Differentially Private User-Based Recommendation and designed a low-sensitivity metric to measure the similarities between both items and users. Yan et al. [17] proposed a socially aware algorithm called DynaEgo to improve the performance of privacy-preserving collaborative filtering. DynaEgo utilizes the principle of DP as well as the social relationships to adaptively modify the users' rating histories to prevent exact user information from being leaked. Javidbakht and Venkitasubramaniam [18] proposed using DP as a metric to quantify the privacy of the intended destination, and optimal probabilistic routing schemes are investigated under unicast and multicast paradigms. Balu and Furon [19] proposed using sketching techniques to implicitly provide DP guarantees by taking advantage of the inherent randomness of the data structure, and this approach is well suited for large-scale applications. Berlioz et al. [9] applied DP to the latent factor model for each step of MF; however, they did not provide rigorous mathematical proofs and need to do some preprocessing of the raw data; thus, the experimental results showed that a large DP parameter is needed to obtain good predictive accuracy.

Chaudhuri et al. [20] proposed general techniques to produce privacy-preserving approximations of classifiers learned via (regularized) Empirical Risk Minimization (ERM). They

proposed an output perturbation and objective-function perturbation based DP model but these methods were applied to logistic regression and SVM in [20]. Based on the above works, the SVD++ model, which is a derivative model of SVD, is the research object, and three new algorithms that apply DP to SVD++ using gradient perturbation, objective-function perturbation, and output perturbation are proposed. To improve the predictive accuracy, SVD++ considers the related information of the user and item. The theoretical proofs are given and the experiment results show that the new private SVD++ algorithms obtain better predictive accuracy, compared with the same DP treatment of traditional MF [9] and SVD.

The DP parameter is the key to the privacy protection power, but in the current study, it was selected by experience. Finally, an effective trade-off scheme is given that can balance the privacy protection and the predictive accuracy to a certain extent and can provide a reasonable range for parameter selection.

3. Preliminaries

3.1. SVD++ Model. The "user-item" rating matrix is the core data used by the recommender system. MF is a good method of predicting the missing ratings in collaborative filtering. In brief, MF involves factorizing a sparse matrix and finding two latent factor matrices: the first is the user matrix to indicate the user's features (i.e., the degree of preference of a user for each factor) and the other is the item matrix, which indicates the item's features (i.e., the weight of an item for each factor). The missing ratings are then predicted from the inner product of these two factor matrices.

Let $R_{n\times m}$ be a rating matrix containing the ratings of n users for m items. Each matrix element r_{ui} refers to the rating of user u for item i. Given a lower dimension d, MF factorizes the raw matrix $R_{n\times m}$ into two latent factor matrices: one is the user-factor matrix $P_{n\times d}$ and the other is the item-factor matrix $Q_{d\times m}$. The factorization is done such that R is approximated as the inner product of P and Q (i.e., $\widetilde{R}_{n\times m} = P_{n\times d}\times Q_{d\times m}$), and each observed rating r_{ui} is approximated by $\widetilde{r}_{ui} = q_i^T \cdot p_u$ (also called the predicted value). However, $q_i^T \cdot p_u$ only captures the relationship between the user u and the item i. In the real world, the observed rating may be affected by the preference of the user or the characteristics of the item. In other words, the relationship between the user u and the item i can be replaced by the bias information. For instance, suppose one wants to predict the rating of the movie "Batman" by the user "Tom." Now, the average rating of all movies on one website is 3.5, and Tom tends to give a rating that is 0.3 lower than the average because he is a critical man. The movie "Batman" is better than the average movie, so it tends to be rated 0.2 above the average. Therefore, considering the user and movie bias information, by performing the calculation $3.5 - 0.3 + 0.2 = 3.4$, it is predicted that Tom will give the movie "Batman" a rating of 3.4. The user and item bias information can reflect the truth of the rating more objectively. SVD is a typical factorization technology (known as a baseline predictor in some works in the literature). Thus, the predicted rating is changed to

$$\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot p_u, \tag{1}$$

where μ is the overall average rating and b_u and b_i indicate the observed deviations of user u and item i, respectively.

The goal of a recommender system is to improve the predictive accuracy. In fact, the user will leave some implicit feedback information, such as historical browsing data, and historical rating data, on Web applications as long as any user has rated item i, no matter what the specific rating value is. To a certain extent, the rating operation already reflects the degree of a user's preference for each latent factor. Therefore, the SVD++ model introduces the implicit feedback information based on SVD; that is, it adds a factor vector ($y_j \in R^f$) for each item, and these item factors are used to describe the characteristics of the item, regardless of whether it has been evaluated. Then, the user's factor matrix is modelled, so that a better user bias can be obtained. Thus, the predictive rating of the SVD++ model is

$$\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot \left(p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j \right), \tag{2}$$

where $R(u)$ is the number of items rated by user u.

To obtain the optimal P and Q, the regularized squared error can be minimized as follows. The objective function of the SVD++ model is

$$\min_{P,Q} \sum_{r_{ui}\in R} \left[r_{ui} - \mu - b_u - b_i - q_i^T \cdot \left(p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j \right)^2 + \lambda \left(b_u^2 + b_i^2 + \|p_u\|^2 + \|q_i\|^2 \right) \right], \tag{3}$$

where λ is the regularization parameter to regularize the factors and prevent overfitting.

With regard to b_u, b_i, and $\sum y_j$, two methods can be used [1]: fast empirical likelihood estimation (i.e., formula (4)) and Stochastic Gradient Descent (SGD). Considering the rate of convergence and the influence of the error in each iteration, the first method is used in this paper.

$$b_i = \frac{\sum_{u\in R(i)} (r_{ui} - \mu)}{\lambda_1 + |R(i)|},$$

$$b_u = \frac{\sum_{i\in R(u)} (r_{ui} - \mu - b_i)}{\lambda_2 + |R(u)|} \tag{4}$$

$$\sum_{j\in R(u)} y_j = \frac{\sum_{j\in R(u)} I(r_{uj} > 0)}{\lambda_3 + |R(u)|}.$$

In formula (4), when $r_{uj} > 0$, the value of $I(r_{uj} > 0)$ will be 1; otherwise, it will be 0. In addition, averages tend to zero using the regularization parameters λ_1, λ_2, and λ_3, which are determined by cross-validation.

SGD and Alternating Least Squares (ALS) are two common optimization algorithms used to solve the objective function (formula (4)). The SGD algorithm is a combination of randomness and optimization and does not need to calculate the exact value but uses unbiased estimation.

Stochastic Gradient Descent. Let e_{ui} represent the error between the true and the predicted values (i.e., $e_{ui} = r_{ui} - \tilde{r}_{ui}$). p_u is any element of the user matrix P, q_i is any element of the item matrix Q, and the error of SVD++ can be expressed as $e_{ui} = r_{ui} - (\mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2} \sum_{j \in R(u)} y_j))$. In SGD, the factors are learned by iteratively evaluating the error e_{ui} for each rating r_{ui}, and the user and item vectors are updated by taking a step in the direction opposite to the gradient of the regularized loss function. Then, the updating rules for both p_u and q_i can be formulated as follows:

$$\begin{aligned} p_u &\longleftarrow p_u + \gamma (e_{ui} q_i - \lambda p_u), \\ q_i &\longleftarrow q_i + \gamma (e_{ui} p_u - \lambda q_i), \end{aligned} \tag{5}$$

where constant γ is the learning rate and can determine the rate of error minimization.

Alternating Least Squares. In ALS, the optimization problem can be solved iteratively. One latent matrix (say P) in each iteration is fixed and then the objective function of SVD++ (formula (3)) is converted into a convex optimization problem, where the solution (say Q) can be found efficiently. Similarly, another latent matrix can be found in the same way. Finally, these steps are repeated until convergence is achieved.

3.2. Differential Privacy. The privacy protection of the collaborative filtering algorithm needs not only to reduce the risk of leaking the private information from the original data but also to ensure the availability of data. DP defines an extremely strict attack model and provides a rigorous, quantitative representation and proof of the risk of leakage of private information. The amount of background knowledge that the attacker has does not matter since DP protects information of the user's potential privacy by adding noise in order to prevent the attacker from inferring the user's protected information even if the attacker knows other information. The attacker does not know whether certain user information exists in the original dataset. Because DP can result in recommendation results not related to the information in the original dataset, DP is applied to the recommender system based on collaborative filtering to prevent indirect deduction of personal private information.

Definition 1 (ε-differential privacy). Given any two adjacent "user-item" rating matrices $R_{n\times m}$ and $R'_{n\times m}$, which differ by at most one score, if any possible output result S ($S \in \text{Range}(A)$) satisfies formula (6), the random algorithm A provides ε-differential privacy.

$$\Pr\left[A\left(R_{n\times m}\right) \in S\right] \leq \exp(\varepsilon) \times \Pr\left[A\left(R'_{n\times m}\right) \in S\right], \tag{6}$$

where $\Pr[\cdot]$ is the probability that private information will be disclosed and is controlled by the randomness of algorithm A; it is independent of the background knowledge of the attacker. Parameter ε is used to indicate the strength of privacy protection, where a smaller value indicates a higher strength of privacy protection. In addition, the two rating matrices differ by at most one score and can also be understood as two matrices that differ by at most one record of a user.

The key technology of DP protection is to add noise that satisfies the Laplace or exponent mechanism [21]. The former is applied to the results for numerical protection and the latter is applied for nonnumerical protection. The amount of noise is related to the function's sensitivity and the privacy protection parameter ε. The sensitivity of the function is that the maximum difference in the output results comes from two datasets that differ by only one record. The sensitivity is divided into global sensitivity and local sensitivity. The former is determined by the function itself and different functions will have different global sensitivities. The latter is determined by the specific given dataset and the function itself. The formal definition of global sensitivity, the Laplace mechanism, and the two composition properties of DP are given as follows.

Definition 2 (global sensitivity). Given any two adjacent "user-item" rating matrices $R_{n\times m}$ and $R'_{n\times m}$ that differ by at most one score, for any function $f : (R_{n\times m}, I) \to \mathbb{R}$, the L_k-global sensitivity of function f is

$$\text{GS}_f = \max_{R,R'} \left\| f(R, i) - f\left(R', i\right) \right\|_k, \tag{7}$$

where d is the dimension of function f, $f(R, i)$ is the predicted value of item i, and $\| \cdot \|_k$ denotes the L_k-norm.

If the global sensitivity of the function is too large to compute the average, median, and so forth, enough noise must be added to protect the privacy, but this will lead to the reduction in the availability of data. To address this problem, Nissim et al. [22] proposed the local sensitivity. In this paper, global sensitivity is adopted because the sensitivity of our function is small.

Dwork et al. [21] demonstrated that the Laplace mechanism could be used to obtain ε-differential privacy. The main idea is to add noise sampled from a Laplace distribution with a calibrated scale b. The probability density function of the Laplace distribution with mean 0 and scale b is

$$f(x \mid b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right). \tag{8}$$

In this paper, it is denoted as lap(b).

Theorem 3. *Given any two adjacent "user-item" rating matrices $R_{n\times m}$ and $R'_{n\times m}$ that differ by at most one score, for any function $f : (R_{n\times m}, I) \to \mathbb{R}$ (its global sensitivity is GS_f), if the random noise $Y \sim Lap(\mathrm{GS}_f/\varepsilon)$, and the algorithm A satisfy*

$$A(R, i) = f(R, i) + Y, \tag{9}$$

the algorithm A provides ε-differential privacy.

This work also relies on the K-norm mechanism [23], which makes it possible to calibrate noise to the L_2-sensitivity of the evaluated function.

In this paper, the outputs of the new privacy algorithms are all numerical, so the Laplace mechanism is used to achieve DP.

Composition. Usually, a complex privacy-preserving problem requires DP protection technology to be applied multiple times. In this case, in order to ensure that the privacy protection level of the whole process is controlled within the budget given by the privacy protection parameter ε, two important composition properties of DP itself are required. One is the sequential composition property, and the other is the parallel composition property [21]. The sequential composition property ensures that multiple random algorithms are distributed in a DP budget (like $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$), and each algorithm maintains ε_i-differential privacy. For the same dataset, the composition algorithm of these algorithms will maintain the sum of the total privacy budget DP (i.e., it will maintain $(\sum_i \varepsilon_i)$-differential privacy). The parallel composition property means that, for a disjoint dataset, the composition algorithm of these algorithms will maintain the maximum total privacy budget DP (i.e., it will maintain $(\max \varepsilon_i)$-differential privacy).

4. Privacy-Preserving SVD++

The intuitive idea is that, after using traditional MF to solve this problem, there should be some latent features that determine how a user rates an item. However, if an attacker has some background knowledge, he or she can obtain the user's private data from the original rating matrix. For example, an attacker can infer that a user likes certain types of movies, but the user does not want other people to know this. Thus, our goal is to protect the raw rating matrix by using DP reasonably. The main idea of SVD++ is to analyse the user's preference for each factor and the extent to which the film contains the various factors from the observed ratings and some implicit feedback from users and then to predict the missing score. In this paper, considering the fact that SVD can obtain good predictive accuracy, we apply DP to SVD++ flexibly. Similarly, to the traditional MF, the SVD++ process can also be divided into the following four stages:

(i) Inputting of the original rating matrix

(ii) SVD++ factorization process by SGD or ALS

(iii) Outputting of the user characteristic matrix and the item characteristic matrix

(iv) Rating prediction (i.e., recommendation)

In [9, 10], DP was applied to these four stages and it was necessary to perform some preprocessing of the original matrix. The work of [10] was an extension of [9], and several algorithms in these two works are the same. Compared with [9, 10], our algorithms have three advantages. The first is that our algorithms do not perform any preprocessing with DP in order to ensure the availability of the original data. The second is that our algorithms adopt SVD++ to achieve MF because the SVD++ model considers the user and item biases and implicit feedback information of users in order to improve the recommendation accuracy. The third is that the objective perturbation of ALS for SVD++ comes from the idea of [20] and obtains better experimental results on two datasets than [9, 10].

4.1. SGD with Gradient Perturbation for SVD++. SGD with gradient perturbation for SVD++ applies DP to the error of each iteration in the SGD optimization algorithm. For a detailed description of the process, see Algorithm 1.

For Algorithm 1, a few explanatory points need to be stated as follows:

(1) To constrain the effect of noise, the obtained error can be to a range (in our experiments, we let $e_{\max} = 2$ and $e_{\min} = -2$ due to the experimental rating being between 1 and 5).

(2) The number of gradient descent iterations k should be given in advance.

(3) According to the sequential composition property of DP, the noise at each iteration is calibrated to maintain (ε/k)-differential privacy so that the overall SVD++ maintains ε-differential privacy after k iterations.

Theorem 4. *Given the differential privacy parameter ε and the maximum value ($r_{\max}$) and minimum value ($r_{\min}$) in the "user-item" rating matrix, set $\Delta = r_{\max} - r_{\min}$ and let the rating error in each iteration be $e_{ui} = r_{ui} - \tilde{r}_{ui}$ (r_{ui} is the raw rating and $\tilde{r}_{ui}$ is the predictive rating). If the noise vector is $v(b) \propto \exp(-\varepsilon\|b\|/(\Delta k))$, then Algorithm 1 provides ε-differential privacy after k iterations.*

Proof. First, the error ($e_{ui} = r_{ui} - \tilde{r}_{ui}$) and the global sensitivity of the error ($\mathrm{GS}_{e_{ui}}$) have the largest difference between ratings, so $\mathrm{GS}_{e_{ui}} = r_{\max} - r_{\min}$.

Second, in k iterations, if the differential privacy is ε, then the budget allocated at each iteration should be ε/k.

Third, b is a noise vector that is added to e_{ui} in each iteration and its probability density is $v(b) \propto \exp(-\varepsilon\|b\|/(\Delta k))$. According to the Laplace mechanism, the new error becomes $e'_{ui} = e_{ui} + \mathrm{Lap}(\mathrm{GS}_{e_{ui}}/(\varepsilon/k)) = e_{ui} + \mathrm{Lap}(\Delta k/\varepsilon)$. Therefore, the error in each iteration maintains (ε/k)-differential privacy.

Finally, according to the sequential composition property of DP, Algorithm 1 provides $((\varepsilon/k) * k)$-differential privacy (i.e., it provides ε-differential privacy) after k iterations. □

4.2. Private-Preserving ALS for SVD++. Two new approaches were proposed in [20], namely, objective perturbation and

Input: $R_{n\times m} = \{r_{ui}\}$ – "user-item" rating matrix
d – number of factors
γ – learning rate
λ – regularization parameter of SVD++ objective function
λ_1, λ_2 and λ_3 – regularization parameters for computing the item bias, user bias, and implicit feedback factor
k – number of gradient descent iterations
$e_{\max}$ and $e_{\min}$ – upper and lower bounds on the per-rating error
ε – differential privacy parameter
Output: Latent factor matrices $P_{n\times d}$ and $Q_{d\times m}$
(1) Initialize the random latent factor matrices P and Q
(2) **for** k iterations **do**
(3) **for** each r_{ui} **do**
$b_i = \dfrac{\sum_{u\in R(i)}(r_{ui}-\mu)}{\lambda_1 + |R(i)|}$, $b_u = \dfrac{\sum_{i\in R(u)}(r_{ui}-\mu-b_i)}{\lambda_2 + |R(u)|}$
(4) $\sum\limits_{j\in R(u)} y_j = \dfrac{\sum_{j\in R(u)} I(r_{uj}>0)}{\lambda_3 + |R(u)|}$
(5) $\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2} \sum\limits_{j\in R(u)} y_i)$
(6) $e_{ui} = r_{ui} - \widetilde{r}_{ui}$
(7) $e'_{ui} = e_{ui} + b$
(where $\nu(b) \propto \exp(-\varepsilon\|b\|/\Delta k)$ and $\Delta = r_{\max} - r_{\min}$)
(8) Clamp e'_{ui} to $[e_{\min}, e_{\max}]$
(9) update $p_u : p_u \leftarrow p_u + \gamma(e'_{ui} q_i - \lambda p_u)$
(10) update $q_i : q_i \leftarrow q_i + \gamma(e'_{ui} p_u - \lambda q_i)$
(11) **end for**
(12) **end for**
(13) **return** $P_{n\times d}$ and $Q_{d\times m}$

Algorithm 1: SGD with gradient perturbation for SVD++ (DPSS++).

output perturbation using DP for the design of privacy-preserving algorithms, and then they were applied to logistic regression and SVM. Specifically, experimental results showed that the results of objective perturbation are optimal when balancing privacy protection and predictive accuracy. In this subsection, this approach is applied to the ALS optimization algorithm of SVD++. Algorithm 2 describes the process of ALS objective perturbation and Algorithm 3 describes the process of ALS output perturbation.

In the SVD++ model, considering the user's bias, the item's bias, and the rating information to which the user has contributed in which the user has taken part, then the predicted rating is changed to

$$\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot \left(p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j \right) \tag{10}$$

(see Section 3.1). The basic principle of ALS for solving SVD++ can be seen in Section 3.1. According to the principle of ALS, the raw objective function (formula (3)) becomes two convex optimization problems as follows:

$$\begin{aligned} J_Q(p_u, R) &= \sum_{R_u} (r_{ui} - \widetilde{r}_{ui})^2 + n_u \lambda \|p_u\|_2^2, \\ J_p(q_i, R) &= \sum_{R_u} (r_{ui} - \widetilde{r}_{ui})^2 + n_i \lambda \|q_i\|_2^2, \end{aligned} \tag{11}$$

where R_u and R_i are subsets of raw R and

$$\begin{aligned} R_u &= \{r_{vi} \in R \mid v = u\}, \\ n_u &= |R_u|, \\ R_i &= \{r_{uv} \in R \mid v = i\}, \\ n_i &= |R_i|. \end{aligned} \tag{12}$$

Then, the main idea of Algorithm 2 is to add noise to the objective function; that is,

$$\begin{aligned} J_Q^{\text{priv}}(p_u, R) &= J_Q(p_u, R) + \frac{1}{n} b^T p_u, \\ J_P^{\text{priv}}(q_i, R) &= J_P(q_i, R) + \frac{1}{n} b^T q_i, \end{aligned} \tag{13}$$

where b is a noise vector with d components and d is the number of features of P or Q. To solve the convex optimization problem, the idea of ERM [20] is used. So, from formula (13), we can obtain

$$p_u^{\text{priv}} = \arg\min_{p_u} J_Q^{\text{priv}}(p_u, R) + \frac{1}{2}\Delta \|p_u\|^2, \tag{14}$$

$$q_i^{\text{priv}} = \arg\min_{q_i} J_P^{\text{priv}}(q_i, R) + \frac{1}{2}\Delta \|q_i\|^2. \tag{15}$$

According to Algorithm 2 of [20], the regularization terms $(1/2)\Delta\|p_u\|^2$ and $(1/2)\Delta\|q_i\|^2$ avoid overfitting after

Input: $R_{n\times m} = \{r_{ui}\}$ – "user-item" rating matrix
d – number of factors
N – total number of ratings
λ – regularization parameter of SVD++ objective function
λ_1, λ_2 and λ_3 – regularization parameters for computing the item bias, user bias, and implicit feedback factor
k – number of gradient descent iterations
ε – differential privacy parameter
C – the parameter for computing the slack term
Output: Latent factor matrices $P_{n\times d}$ and $Q_{d\times m}$
(1) Initialize random latent factor matrices P and Q:
(2) **for** k iterations **do**
(3) **for** each r_{ui} **do**

$$b_i = \frac{\sum_{u\in R(i)}(r_{ui}-\mu)}{\lambda_1 + |R(i)|},\ b_u = \frac{\sum_{i\in R(u)}(r_{ui}-\mu-b_i)}{\lambda_2 + |R(u)|}$$

(4) $$\sum_{j\in R(u)} y_j = \frac{\sum_{j\in R(u)} I(r_{uj} > 0)}{\lambda_3 + |R(u)|}$$

(5) $$\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j)$$

(6) **for** each user u, when given matirx Q, **do**
(7) let $\varepsilon' = \varepsilon - \log(1 + 2C/N\lambda + C^2/N^2\lambda^2)$
(8) if $\varepsilon' > 0$ then $\Delta = 0$
(9) else $\Delta = C/N(e^{\varepsilon/4} - 1) - \lambda$, and $\varepsilon' = \varepsilon/2$
(10) Generate random noise vector b with pdf

$$\nu(b) \propto \exp\left(-\frac{\varepsilon' \|b\|}{2}\right)$$

(11) Compute $p_u^{\text{priv}} = \arg\min_{p_u} J_Q^{\text{priv}}(p_u, R) + (1/2)\Delta\|p_u\|^2$
(12) **end for**
(13) **for** each item i, when given matrix P **do**
(14) Omit (the same as (7)~(10))
(15) Compute $q_i^{\text{priv}} = \arg\min_{q_i} J_P^{\text{priv}}(q_i, R) + (1/2)\Delta\|q_i\|^2$
(16) **end for**
(17) **end for**
(18) **end for**
(19) **return** $P_{n\times d}$ and $Q_{d\times m}$

Algorithm 2: ALS with objective perturbation for SVD++ (DPSAObj++).

perturbation, where Δ is determined by the privacy parameter ε and the slack term parameter C.

The ALS objective functions for SVD++ are convex and differentiable, so they satisfy the application conditions of Algorithm 2 of [20]. In this paper, our Algorithm 2 describes the DP protection process of ALS objective perturbation to solve for the latent factors of SVD++.

Regarding Algorithm 2, a few explanatory points should be stated as follows:

(1) First, to deduce and compute the value of parameter C in steps (7) and (9), the value of C is set to 2. The specific deduction process is similar to the deduction applied in logistic regression (Corollary 4) and SVM (Corollary 6) from [20].

(2) To solve for the values of p_u and q_i after objective perturbation, that is, to solve for the partial derivatives of formulas (14) and (15), respectively, where n indicates the number of users and m indicates the number of items in the raw matrix, the key steps are as follows.

When $\forall 1 \le u \le n$ and $1 \le k \le d$, we can obtain

$$\frac{1}{2}\frac{\partial p_u^{\text{priv}}}{\partial p_{uk}} = \sum_i \Bigg(\mu + b_u + b_i + q_i^T \left(p_u + |R_u|^{-1/2} \sum_{j\in R(u)} y_j \right) - r_{ui} \Bigg) q_{ik} + \lambda n_u p_{uk} + \frac{1}{N} b_k + \frac{1}{2}\Delta p_{uk}. \tag{16}$$

Then, we have

$$\frac{1}{2}\frac{\partial p_u^{\text{priv}}}{\partial p_{uk}} = \frac{1}{2}\left(\frac{\partial p_u^{\text{priv}}}{\partial p_{u1}}, \ldots, \frac{\partial p_u^{\text{priv}}}{\partial p_{ud}}\right) = p_u \left[Q^T Q + \left(\lambda n_u + \frac{1}{2}\Delta\right) I \right]$$

Input: $R_{n\times m} = \{r_{ui}\}$ – "user-item" rating matrix
d – number of factors
λ – regularization parameter of SVD++ objective function
λ_1, λ_2 and λ_3 – regularization parameters for computing the item bias, user bias, and implicit feedback factor
k – number of gradient descent iterations
ε – differential privacy parameter
Output: Latent factor matrices $P_{n\times d}$ and $Q_{d\times m}$
(1) Initialize random latent factor matrices P and Q:
(2) **for** k iterations **do**
(3) **for** each r_{ui} **do**

$$b_i = \frac{\sum_{u\in R(i)}(r_{ui}-\mu)}{\lambda_1 + |R(i)|},\ b_u = \frac{\sum_{i\in R(u)}(r_{ui}-\mu-b_i)}{\lambda_2 + |R(u)|}$$

(4) $$\sum_{j\in R(u)} y_j = \frac{\sum_{j\in R(u)} I(r_{uj} > 0)}{\lambda_3 + |R(u)|}$$

(5) $$\tilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j)$$

(6) **for** each user u, when given matrix Q, **do**
(7) Generate random noise vector b with pdf
(8) $$f(b) \propto \exp\left(-\frac{\varepsilon \|b\|}{2k} \cdot \frac{n_u \lambda}{2q_{\max}\Delta r}\right)$$
(9) $$p_u(R, Q) \longleftarrow \arg\min_{p_u} J_Q(p_u, R) + b$$
(10) **end for**
(11) **for** each item i, when given matrix P **do**
(12) Generate random noise vector b with pdf
(13) $$f(b) \propto \exp(-\frac{\varepsilon \|b\|}{2k} \cdot \frac{n_i \lambda}{2p_{\max}\Delta r})$$
(14) $$q_i(R, P) \longleftarrow \arg\min_{q_i} J_P(q_i, R) + b$$
(15) **end for**
(16) **end for**
(17) **end for**
(18) **return** $P_{n\times d}$ and $Q_{d\times m}$

Algorithm 3: ALS with output perturbation of SVD++ (DPSASOut++).

$$+\left(|R_u|^{-1/2} \sum_{j\in R(u)} y_j\right) Q^T Q - \left(R_u - \mu - b_u - b_i\right) Q + \frac{1}{N}\mathbf{b}, \tag{17}$$

where $n_u = |R_u|$, $R_u = \{r_{vi} \in R \mid v = u\}$, and I is a $d \times d$ identity matrix.

Then, fixing Q and solving $\partial p_u^{\text{priv}} / \partial p_{uk} = 0$, we have

$$p_u = \left(R_u Q - b_u Q - b_i Q - \mu Q - \left(|R_u|^{-1/2} \sum_{j\in R(u)} y_j\right) Q^T Q - \frac{1}{N}\mathbf{b}\right) \times \left[Q^T Q + \left(\lambda n_u + \frac{1}{2}\Delta\right) I\right]^{-1}. \tag{18}$$

Similarly, given a fixed P, when $\forall 1 \le i \le m$, we can solve Q as follows:

$$q_i = \left(R_i P - b_u P - b_i P - \mu P - \left(|R_i|^{-1/2} \sum_{j\in R(i)} y_j\right) P^T P - \frac{1}{N}\mathbf{b}\right) \times \left[P^T P + \left(\lambda n_i + \frac{1}{2}\Delta\right) I\right]^{-1}, \tag{19}$$

where $n_i = |R_i|$, $R_i = \{r_{uv} \in R \mid v = i\}$.

Theorem 5. *Given the differential privacy parameter ε and the parameter for computing the slack term C, if $\|p_u\|^2$, $\|q_i\|^2$, and the loss functions of ALS are convex and differentiable, Algorithm 2 provides ε-differential privacy.*

Proof. Our Algorithm 2 satisfies the application condition of Algorithm 2 in [20], which was proven to provide ε-differential privacy; thus our Algorithm 2 also provides ε-differential privacy.

Another privacy-preserving ALS algorithm of SVD++ is the ALS output perturbation method, which is shown in Algorithm 3.

In the objective function of ALS (i.e., formula (11)), each user vector p_u and item vector q_i can be obtained by solving the following risk minimization problem:

$$\begin{aligned} p_u(R,Q) &= \arg\min_{p_u} J_Q(p_u,R), \\ q_i(R,P) &= \arg\min_{q_i} J_P(q_i,R). \end{aligned} \tag{20}$$

The main idea of Algorithm 3 is that it guarantees DP by adding a random noise vector b to the output of $p_u(R,Q)$ and $q_i(R,P)$. □

Regarding Algorithm 3, a few explanatory points should be stated as follows:

(1) $p_{\max}$ and $q_{\max}$ are the upper bounds on $\|p_u\|^2$ and $\|q_i\|^2$, respectively; $\Delta r = r_{\max} - r_{\min}$. Because $p_u(R,Q)$ and $q_i(R,P)$ are the L_2-sensitivity values, their global sensitivities can be obtained as $\mathrm{GS}p_u = 2q_{\max}\Delta r/n_u\lambda$ and $\mathrm{GS}q_i = 2p_{\max}\Delta r/n_i\lambda$.

(2) According to the Laplace mechanism, for a fixed matrix Q, a random noise vector b with the pdf $f(b) \propto \exp(-\varepsilon\|b\|/2k \cdot n_u\lambda/2q_{\max}\Delta r)$ is generated. For a fixed matrix P, a random noise vector b with the pdf $f(b) \propto \exp(-\varepsilon||b||/2k \cdot n_i\lambda_1/2p_{\max}\Delta r)$ is generated.

(3) For the ALS objective function of SVD++ (formula (11)), we have Corollary 6 and Theorem 7 as follows.

Corollary 6. *Let r_{ui} refer to the rating of user u for item i. The predictive rating in SVD++ is $\tilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2}\sum_{j\in R(u)} y_j)$. $N(\cdot) = \|p_u\|^2$ is differentiable and 1-strongly convex and the loss function $\ell = (r_{ui} - \tilde{r}_{ui})^2$ is convex and differentiable with $|\ell'(\cdot)| \le 1$. Then, the L_2-sensitivity of $J_Q(p_u,R)$ is at most $2q_{\max}\Delta r/n_u\lambda$.*

Proof. Let there be two rating matrices that differ in the value of the last entry:

$$R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nm} \end{pmatrix}, \quad R' = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r'_{nm} \end{pmatrix}. \tag{21}$$

Moreover, let

$$\begin{aligned} G(p_u) &= J_Q(p_u,R), \\ g(p_u) &= J_Q(p_u,R') - J_Q(p_u,R), \\ p_{u1} &= \arg\min_{p_u} J_Q(p_u,R), \\ p_{u2} &= \arg\min_{p_u} J_Q(p_u,R'), \\ g(p_u) &= \left(r'_{ui} - \tilde{r}_{ui}\right)^2 - \left(r_{ui} - \tilde{r}_{ui}\right)^2. \end{aligned} \tag{22}$$

Second, due to the convexity of ℓ and the 1-strongly convexity of $N(\cdot) = \|p_u\|^2$, $G(p_u) = J_Q(p_u,R)$ is $n_u\lambda$-strongly convex.

In addition, due to the differentiability of $N(\cdot) = \|p_u\|^2$ and ℓ, $G(p_u)$ and $g(p_u)$ are also differentiable at all points. Then, we have

$$\begin{aligned} \nabla g(p_u) &= -2\left(r'_{ui} - \tilde{r}_{ui}\right)q_i + 2\left(r_{ui} - \tilde{r}_{ui}\right)q_i \\ &= 2q_i\left(r_{ui} - r'_{ui}\right) = 2q_i\Delta r. \end{aligned} \tag{23}$$

Then, the equation $\|\nabla g(p_u)\| = 2\Delta r\|q_i^T\| \le 2q_{\max}\Delta r$ can be obtained. Hence, the L_2-sensitivity of $J_{Q(P_u,R)}$ is less than or equal to $2q_{\max}\Delta r/n_u\lambda$. The proof now follows by an application of Lemma 1 of [20].

Similarly, the L_2-sensitivity of $q_i(R,P)$ is at most $\mathrm{GS}q_i = 2p_{\max}\Delta r/n_i\lambda$. □

Theorem 7. *Let r_{ui} refer to the rating of user u for item i. The predictive rating in SVD++ is $\tilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2}\sum_{j\in R(u)} y_j)$. $N(\cdot) = \|p_u\|^2$ and $N(\cdot) = \|q_i\|^2$ are differentiable and 1-strongly convex and the loss function $\ell = (r_{ui} - \tilde{r}_{ui})^2$ is convex and differentiable with $|\ell'(\cdot)| \le 1$. Then, Algorithm 3 provides ε-differential privacy.*

Proof. The proof of Theorem 7 follows from Corollary 6 and [20].

(1) According to the proof of Corollary 6, if the conditions on $N(\cdot) = \|p_u\|^2$ and the loss function ℓ hold, the L_2-sensitivity of $J_Q(p_u,R)$ with the regularization parameter $n_u\lambda$ is at most $2q_{\max}\Delta r/n_u\lambda$.

(2) When $\|b\|$ is picked from the distribution $\nu(b) = (1/\alpha)e^{-\beta\|b\|}$, where $\beta = n_u\lambda\varepsilon/2q_{\max}\Delta r$, for a specific vector $b_0 \in \mathbb{R}^d$, the density at b_0 is proportional to $e^{-\beta\|b_0\|}$.

(3) Let $R_{n\times m}$ and $R'_{n\times m}$ be any two rating matrices that differ in the value of the last entry. Then, for any p_u, we have $g(p_u \mid R)/g(p_u \mid R') = \nu(b_1)/\nu(b_2) = e^{-(n_u\lambda\varepsilon/2q_{\max}\Delta r)(\|b_1\|-\|b_2\|)}$, where b_1 and b_2 are the corresponding noise vectors and $g(p_u \mid R)$ ($g(p_u \mid R')$, resp.) is the density of the output of Algorithm 3 at p_u when the input is R (R', resp.).

(4) If p_{u1} and p_{u2} are the respective solutions to nonprivate regularized $J_Q(\cdot)$ when the inputs are R and R', then $b_1 - b_2 = p_{u1} - p_{u2}$. From Corollary 6 and using the triangle inequality, $\|b_1\| - \|b_2\| \le \|b_1 - b_2\| \le \|p_{u1} - p_{u2}\| \le 2q_{\max}\Delta r/n_u\lambda$.

Moreover, by symmetry, the densities of the directions of b_1 and b_2 are uniform. Therefore, by construction, $\nu(b_1)/\nu(b_2) \le e^{\varepsilon}$.

(5) When fixing the latent matrix P and optimizing Q, the proof process is similar. Thus, according to the

definition of DP, Algorithm 3 provides ε-differential privacy. □

5. Experiments

5.1. Experiment Datasets. In the experiments, two datasets are used to verify that our algorithms fit not only a single kind of dataset. One dataset is a MovieLens-1M dataset from http://grouplens.org/datasets/movielens/. The other is a partial Netflix dataset (called Netflix-1M in this paper) that was captured from http://www.netflixprize.com/, which was constructed to support participants in the Netflix Prize. Some statistical properties of the selected MovieLens-1M and the Netflix-1M datasets are shown in Table 1.

5.2. Evaluation Measurement and Experimental Settings. As a frequently used methodology in machine learning and data mining, tenfold cross-validation to train and evaluate the performance of our algorithms is used. The validation datasets are divided into training and test sets with an 80/20 ratio. Then, the Root Mean Square Error (RMSE) metric is used to measure the accuracy of the predicted ratings $\tilde{r}_{ui}$. The smaller the RMSE, the more accurate the prediction is. The RMSE is computed by RMSE $= \sqrt{\sum_R (r_{ui} - \tilde{r}_{ui})^2 / |R|}$, where $|R|$ denotes the number of effective ratings; the ratings here are valid, and missing scores are not included. Considering the possible discrepancies resulting from the addition of noise, the final RMSE is averaged across multiple runs.

The selection of the parameters in each algorithm is introduced briefly.

(i) Except for Figure 4, the number of factors was set to $d = 5$.

(ii) The learning rate was set to $\gamma = 0.001$.

(iii) The regularization parameter of SVD++ was set to $\lambda = 0.125$ by cross-validation.

(iv) The number of iterations was set to $k = 20$ when the error variety is less than 0.0001.

(v) To compare with [9], the values of $p_{\max}$ and $q_{\max}$ in Algorithm 3 were set to the same values as in [9]; that is, $p_{\max} = 0.4$ and $q_{\max} = 0.5$.

(vi) The regularization parameters used to compute the user bias, item bias, and implicit feedback information were set to $\lambda_1 = 10$, $\lambda_2 = 25$ and $\lambda_3 = 10$, respectively, by referring to [1].

5.3. Experimental Results and Comparison

5.3.1. Experimental Results and Analysis. The meanings of the notation used to present the experimental results are shown in Table 2.

The work of [10] was an extension of [9], and several of the same algorithms are used in the two papers. Algorithm 4 of [9] and Algorithm 4 of [10] are the same (called differentially private SGD in the two papers), and Algorithm 5 of [9] and Algorithm 6 of [10] are the same (called differentially private ALS with output perturbation in the two papers).

Table 1: Statistical properties of the two datasets.

Property	MovieLens-1M	Netflix-1M
Users	6040	4996
Movies	3952	3999
Density	4.19%	0.19%
Average rating	3.5816	3.5956
Variance rating	1.2479	1.2208

Figure 1 shows how the results of our three algorithms compare with their baselines (without DP protection) on the two datasets.

From Figure 1, the RMSEs of the proposed algorithms did not deviate from their baselines. On the whole, the results of our algorithms for the MovieLens-1M dataset are better than for the Netflix-1M dataset, because the training samples of the Netflix-1M dataset are fewer and sparser than those of the MovieLens-1M dataset. Thus, it can be concluded that the predictive accuracy is closely related to the dataset size and scarcity, even when carrying out processing by DP. Particularly in Figure 1(b), the predictive accuracy of the ALS perturbation (Algorithms 2 and 3) becomes poor when $\varepsilon < 0.01$ and the ALS output perturbation performs worse than the other algorithms. This is mainly because it perturbs the latent factor matrices after decomposition, and the smaller the value of ε, the more noise added; as a result, the inner product of the two latent factors deviates greatly from its true value. In addition, the two ALS perturbation algorithms are better than the SGD gradient perturbation algorithm (Algorithm 1) when $\varepsilon > 0.01$, even though they were both processed by DP. Particularly, the ALS objective perturbation obtains the best predictive accuracy on the MovieLens-1M dataset, regardless of whether the privacy parameter ε is large or small; that is, the results of this approach processed by DP are the most stable. This is because the update at each iteration of SGD is significantly related to the error and each iteration of ALS is directly related to the training dataset, which means that the ALS method itself is better than SGD.

To increase the predictive accuracy, as the derivative model of SVD, SVD++ introduces implicit feedback information, such as which movies a user has evaluated in the past. Figure 2 shows the results of comparing SVD++ with SVD using three DP protection algorithms. From Figure 2, it can be seen that SVD++ provides a slightly higher advantage over SVD when using the three DP protection algorithms. Overall, the RMSE of ALS with objective perturbation is optimal, especially when $\varepsilon > 0.01$.

In addition, Figure 3 shows the results of our algorithms compared with those of the correlative algorithm of [9] on the two datasets.

In [9], Berlioz et al. also proposed SGD perturbation (called PSGD in our experiments) and ALS output perturbation (called PALS). However, they needed to do some DP preprocessing of the input matrix. In fact, preprocessing of the original input matrix, that is, adding noise to it, will affect the result of SVD++. However, our algorithms not only omit the preprocessing steps but also obtain better prediction accuracies on the two test datasets (from Figure 3). Particularly, the advantage of our ALS with objective perturbation is more obvious. Furthermore, from Figure 3, it

Table 2: The meanings of the notation used to present the experimental results.

Name	Meaning
SGDBase++	Without DP protection, no preprocessing, SGD for SVD++
ALSBase++	Without DP protection, no preprocessing, ALS for SVD++
PSGD	Algorithm 4 of [9] or Algorithm 4 of [10], with preprocessing, SGD for MF
PALS	Algorithm 5 of [9] or Algorithm 6 of [10], with preprocessing, ALS for MF
DPSS	No preprocessing, SGD gradient perturbation for SVD (refer to our Algorithm 1)
DPSAObj	No preprocessing, ALS objective perturbation for SVD (refer to our Algorithm 2)
DPSAOut	No preprocessing, ALS output perturbation for SVD (refer to our Algorithm 3)
DPSS++	Our Algorithm 1, no preprocessing, SGD gradient perturbation for SVD++
DPSAObj++	Our Algorithm 2, no preprocessing, ALS objective perturbation for SVD++
DPSAOut++	Our Algorithm 3, no preprocessing, ALS output perturbation for SVD++

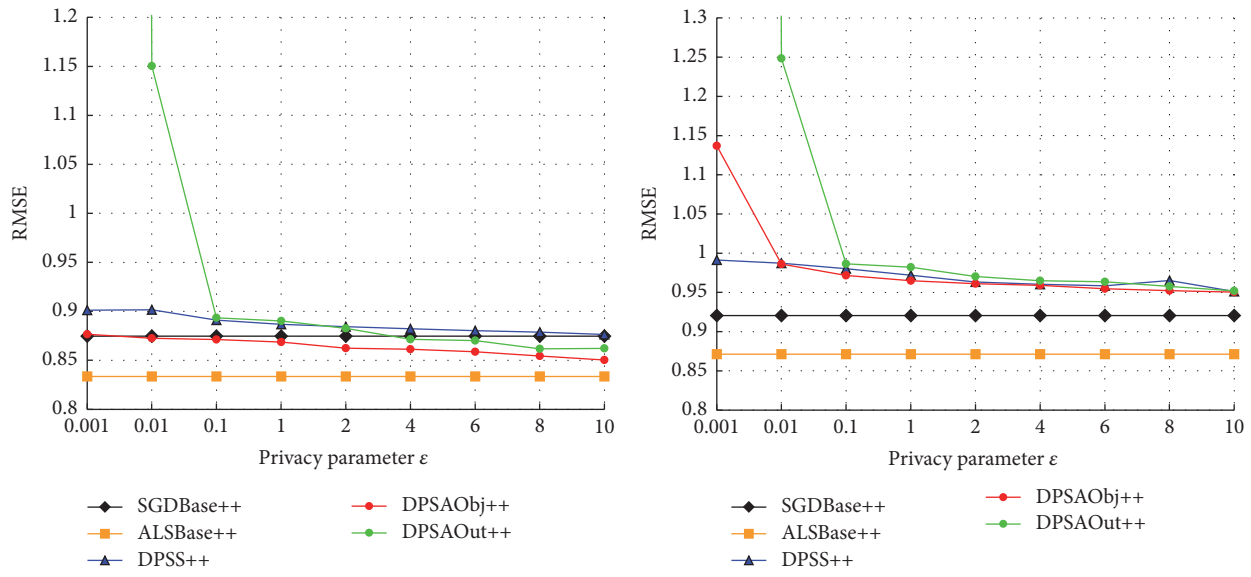

Figure 1: Comparison of the algorithm results with their respective baselines.

is worth noting that their algorithms cannot achieve better prediction accuracy when the value of ε is larger (up to 20). Moreover, the value of ε is too large and would be unreasonable according to the meaning of DP.

In addition, not only are the recommendation results of SVD++ better than those of SVD on a real dataset but also the predictive accuracy will be improved with an increase in the number of features (also called factors) in SVD and SVD++ [24]. To verify that our DP protection algorithms still have this characteristic, Figure 4 shows the relationship between the predictive accuracy and the number of factors after performing SGD gradient perturbation and ALS objective perturbation for SVD and SVD++.

In summary, the three DP algorithms that we have proposed for SVD++ can protect the privacy of the original data on the basis of ensuring the predictive accuracy. In particular, the ALS objective perturbation for the SVD++ algorithm gives a better trade-off between privacy and recommendation accuracy.

5.3.2. A Selection Scheme for DP Parameter ε. In DP applications, the strength of privacy protection depends on the parameter ε, but it is equally important to ensure the predictive accuracy when DP is applied to collaborative filtering, so a scheme for selection of DP protection parameter ε is proposed in order to balance the strength of privacy protection and the predictive accuracy. The specific steps are described as follows.

Step 1. Determine the recommended target user u.

Step 2. Compute the recommended-item set (in this paper, a movie set is used) to the user u from two aspects. Let S_1 be the recommended-item set after performing a certain DP process, and let S_2 be the recommended-item set without performing any DP process.

Step 3. Compute the intersection of the two recommended-item sets obtained in the second step, and denote it as $S = S_1 \cap S_2$.

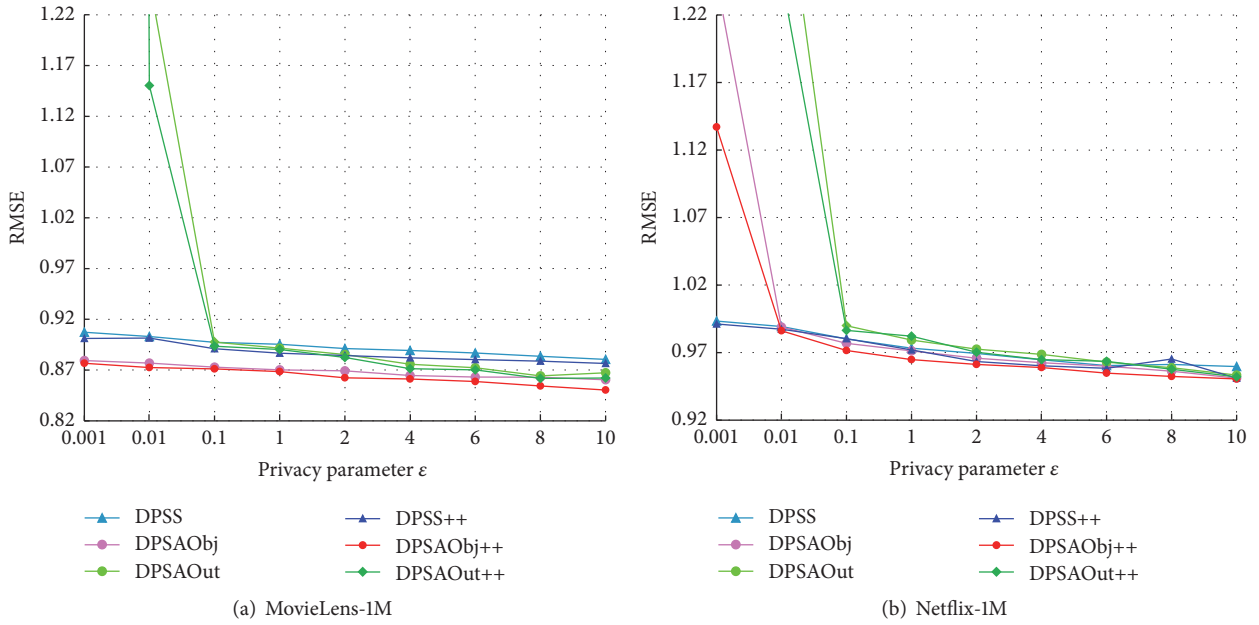

Figure 2: Comparison of SVD++ with SVD using three DP protection algorithms.

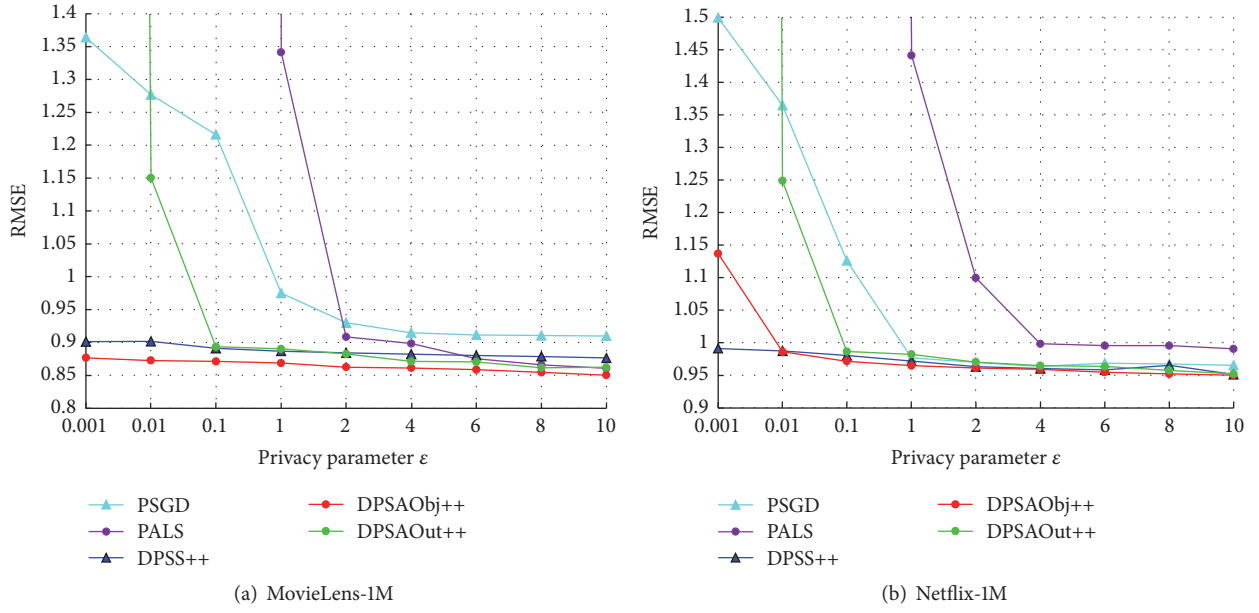

Figure 3: Comparison of our algorithms with the correlative algorithm of [9].

Step 4. If N is the total number of recommended-item sets, obtain a percentage: $P = S/N * 100\%$. The greater P is, the smaller the influence of predictive accuracy is, and the value of ε should be reasonable at this time.

This scheme can only provide a reasonable range for DP parameter ε. Normally, if this percentage is less than 20%, the recommended results are considered to be seriously affected, even though the privacy protection is very strong. On the other hand, if this percentage is more than 80%, the power of privacy protection is thought to be too weak, even though the recommendation results are better. Therefore, the value of DP parameter ε is reasonable when this percentage is between 20 and 80%. To verify this scheme, the ALS DP processes of SVD, SVD++, and the correlation algorithm of [9] (PALS) are compared, and Figure 5 shows the impact of DP parameter ε on the MovieLens-1M dataset. Each parameter in this experiment is still set in accordance with the description given in Section 5.2. In addition, the number of recommended-movie sets is set to 30 and the recommended user is selected randomly. At the same time, the result is the average value of ten runs because of the randomness of Laplace noise.

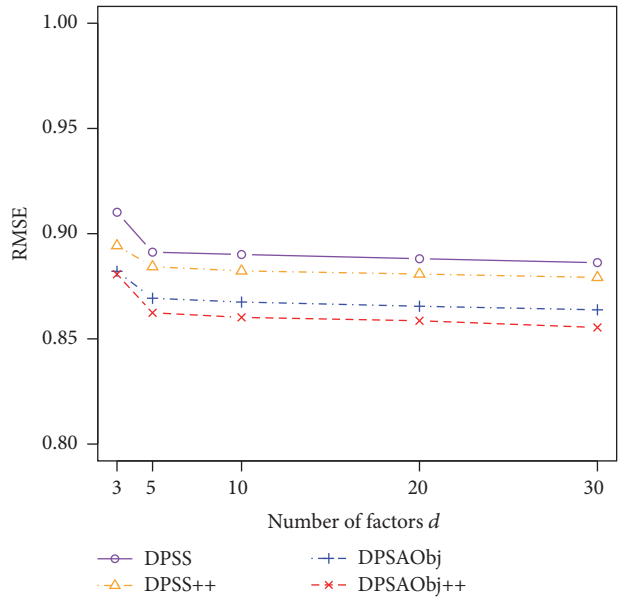

Figure 4: The relationship between the accuracy of recommendation and the number of factors.

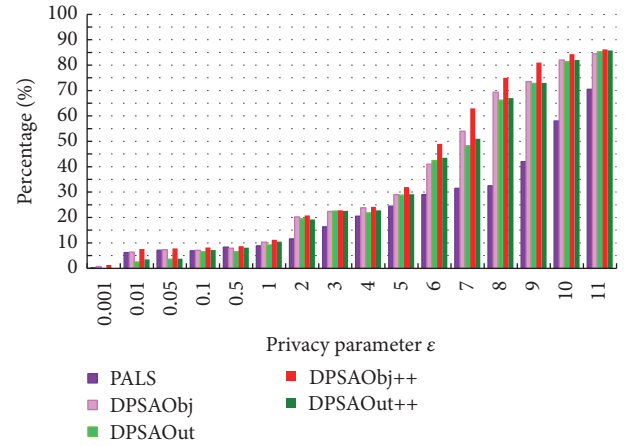

Figure 5: Comparison of the impacts of privacy parameter ε on the recommendation results.

From Figure 5, it can be concluded that the impacts of the privacy parameter ε on the recommendation results of the three new algorithms (especially Algorithm 2) are smaller than those for Algorithm 2 from [9] and SVD, which carries out the same process using DP. For our two algorithms, the coincidence degree of the recommended-movie set is found to be between 20% and 80% when the value of the privacy parameter ε is between 2 and 11. In other words, the values of ε in this percentage range can balance the privacy strength and predictive accuracy better.

6. Discussion

Currently, the services provided by the Web are richer and more colourful. While data providers can obtain convenient personalized services and Web businesses can thus obtain more profits, which is a win-win situation. However, the leakage of personal privacy information has become a very worrying problem for many users. A variety of Internet records on users, film scores, the purchase of goods, and other information provide attackers with a certain background knowledge and personal privacy information can be derived indirectly. Therefore, in order to protect the private information of the original data on the basis of ensuring the predictive accuracy, we proposed three new methods that apply differential privacy to SVD++ through gradient perturbation, objective-function perturbation, and output perturbation. Rigorous mathematical proofs are given to ensure that all three methods maintain the differential privacy. According to experimental verification and comparison with DP privacy-preserving based on SVD and [15] on two real datasets, our new algorithms for SVD++ give better experimental results, especially the approach of ALS objective perturbation for SVD++ (Algorithm 2), which obtained better results in terms of balancing privacy and prediction. A scheme for the selection of DP parameters is finally proposed, and it can obtain a reasonable range for the DP parameter, balancing privacy, and recommendation accuracy.

Recommender systems and the field of data mining require healthy development and are inseparable from the protection of privacy in in-depth research. In the future, a more in-depth study of the following aspects can be expected.

(i) Relative parameter tuning for SVD++: typically, SVD++ parameters, such as the number of factors, the regularization parameter, and the learning rate, are tuned to increase prediction accuracy, while preventing overfitting and ensuring convergence.

(ii) More effective selection of DP parameter ε: in this paper, only the selection interval of ε is provided, but it is hard to determine the optimal ε. After all, the Laplace noise itself is random.

(iii) Comparison of other collaborative filtering or recommender algorithms: in this paper, the new approach is the application of DP to the optimal algorithms of SVD++. To extend the application of DP, other collaborative filtering or recommender algorithms could be studied and compared with one another in terms of their recommender effects.

(iv) Multiple evaluation measurements might be used to verify the new algorithms.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is sponsored in part by the Natural Science Foundation of Guangdong Province (nos. 2014A030313662 and 2016A030310018) and College Students' Science and Technology Innovation Fund of Guangdong Province (no. G2016Z08).

References

[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, Springer, Berlin, Germany, 2010.

[2] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[3] B. Mehta, T. Hofmann, and W. Nejdi, "Robust collaborative filtering," in *Proceedings of the 1st ACM Conference on Recommender Systems (RecSys '07)*, pp. 49–56, Minneapolis, Minn, USA, October 2007.

[4] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 426–434, Las Vegas, Nev, USA, August 2008.

[5] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP '06)*, pp. 1–12, Venice, Italy, July 2006.

[6] K. Su, L. L. Ma, B. Xiao, and H. Q. Zhang, "Web service QoS prediction by neighbor information combined non-negative matrix factorization," *Journal of Intelligent and Fuzzy Systems*, vol. 30, no. 6, pp. 3593–3604, 2016.

[7] Q. Liu, Q. Wu, Y. Zhang, and X. Wang, "Recommendation-based third-party tracking monitor to balance privacy with personalization," in *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS '14)*, pp. 1472–1474, Scottsdale, Ariz, USA, November 2014.

[8] P. Dandekar, N. Fawaz, and S. Ioannidis, "Privacy auctions for recommender systems," *ACM Transactions on Economics and Computation*, vol. 2, no. 3, pp. 1–22, 2014.

[9] A. Berlioz, A. Friedman, M. A. Kaafar, R. Boreli, and S. Berkovsky, "Applying differential privacy to matrix factorization," in *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*, pp. 107–114, Vienna, Austria, September 2016.

[10] A. Friedman, S. Berkovsky, and M. A. Kaafar, "A differential privacy framework for matrix factorization recommender systems," *User Modeling and User-Adapted Interaction*, vol. 26, no. 5, pp. 425–458, 2016.

[11] J. Canny, "Collaborative filtering with privacy," in *Proceedings of the IEEE Symposium on Security and Privacy (S and P '02)*, pp. 45–57, Berkeley, Calif, USA, May 2002.

[12] F. McSherry and I. Mironov, "Differentially private recommender systems: building privacy into the netflix prize contenders," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 627–635, Paris, France, July 2009.

[13] T. Q. Zhu, G. Li, Y. L. Ren, W. L. Zhou, and P. Xiong, "Differential privacy for neighborhood-based collaborative filtering," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '13)*, pp. 752–759, ACM, Ontario, Canada, August 2013.

[14] J. Hua, C. Xia, and S. Zhong, "Differentially private matrix factorization," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI '15)*, pp. 1763–1770, Buenos Aires, Argentina, July 2015.

[15] Z. Liu, Y.-X. Wang, and A. J. Smola, "Fast differentially private matrix factorization," in *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*, pp. 171–178, Vienna, Austria, September 2015.

[16] X. Zhu and Y. Sun, "Differential privacy for collaborative filtering recommender algorithm," in *Proceedings of the 2nd ACM International Workshop on Security and Privacy Analytics (IWSPA '16)*, pp. 9–16, New Orleans, La, USA, March 2016.

[17] S. Yan, S. Pan, W. Zhu, and K. Chen, "DynaEgo: privacy-preserving collaborative filtering recommender system based on social-aware differential privacy," in *Information and Communications Security*, vol. 9977 of *Lecture Notes in Computer Science*, pp. 347–357, Springer International, Cham, Switzerland, 2016.

[18] O. Javidbakht and P. Venkitasubramaniam, "Differential privacy in networked data collection," in *Proceedings of the Annual Conference on Information Science and Systems (CISS '16)*, pp. 117–122, Princeton, NJ, USA, March 2016.

[19] R. Balu and T. Furon, "Differentially private matrix factorization using sketching techniques," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '16)*, pp. 57–62, ACM, Vigo, Spain, June 2016.

[20] K. Chaudhuri, C. Monteleoni, and A. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, pp. 1069–1109, 2011.

[21] C. Dwork, F. F. McShery, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Conference on Theory of Cryptography (TCC '06)*, pp. 265–284, New York, NY, USA, March 2006.

[22] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the 39th Annual ACM Symposium on Theroy of Computing (STOC '07)*, pp. 75–84, San Diego, Calif, USA, June 2007.

[23] M. Hardt and K. Talwar, "On the geometry of differential privacy," in *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC '10)*, pp. 705–714, Cambridge, Mass, USA, June 2010.

[24] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 447–456, Paris, France, June 2009.