Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 2959503, 14 pages
https://doi.org/10.1155/2023/2959503

WILEY | Hindawi

Research Article

A Normative Approach to Privacy-Preserving Recommender Systems: Integrating Matrix Factorization and Genetic Algorithms

Ming He and Sheng Hu

Chongqing Industry Polytechnic College, Chongqing 401120, China

Correspondence should be addressed to Sheng Hu; husheng@cqipc.edu.cn

Received 29 March 2023; Revised 1 August 2023; Accepted 21 August 2023; Published 4 September 2023

Academic Editor: Vasudevan Rajamohan

As recommendation systems heavily depend on user data, these systems are susceptible to potential privacy breaches. To mitigate this issue, differential privacy (DP) protection techniques have been developed to offer robust privacy safeguards. Nevertheless, a majority of the extant DP-based recommendation algorithms tend to introduce excessive noise, consequently impairing the quality of recommendations. In response, this study presents a novel DP-preserving recommendation algorithm that integrates matrix factorization (MF) and a genetic algorithm (GA). Initially, the MF problem is transformed into two interrelated optimization problems, namely, the user-hidden factor and the item-hidden factor. Subsequently, GA is employed to address these optimization issues. An enhancement index mechanism is incorporated into the individual selection of GA, while the variation process of GA is devised based on identifying crucial hidden factors. Utilizing the enhancement index mechanism aids in minimizing the algorithm's perturbation level, thereby achieving an optimal balance between privacy protection and algorithm utility. Experimental analyses, encompassing recommendation accuracy, efficiency, and parameter variation effects, are conducted on Last.fm and Flixster datasets. The findings corroborate that the proposed system outperforms existing alternatives under stringent privacy constraints, thereby attesting to its efficacy.

1. Introduction

In recent years, with the rapid development of the Internet economy, there is a trend of rapid growth in the amount of information on the network, making it sometimes difficult for users to quickly filter out the information they are interested in from the vast amount of information. Although some search engines (such as Baidu and Google) have developed and adopted some special search algorithms to achieve targeted search according to keywords entered by users, the results searched in this way often cannot meet the real needs of users to quickly obtain the information they need. In order to make it more convenient and easier for users to get the information they need from the Web, personalized recommendation services are becoming an integral part of web application services [1]. Therefore, recommendation systems are widely used and rapidly spreading. Recommendation systems are designed to deliver information of interest directly to users according to their preferences, which can significantly reduce the workload of sifting through large amounts of information and bring convenience to users' life and work [2].

The core part of a recommendation system is the recommendation algorithm, and a high-performance recommendation algorithm is naturally the key to building a high-quality recommendation system. As a mainstream recommendation algorithm, collaborative filtering (CF) uses the user's historical evaluation information to predict the user's preference for unknown items and make recommendations accordingly [3]. CF techniques need to use a large amount of user data, and there is a risk of user privacy leakage. In the neighbor-based CF technique, an attacker can infer the target user's rating of an item by tracking the changes in the recommendation list of neighboring users [4]. In MF-based CF techniques, since the decomposition of the resulting matrix of hidden factors carries data information, it may be

used by attackers to infer users' rating data through reconstruction attacks and other means. The compromised ratings may be further used to infer the user's gender, age, and other information, violating the user's privacy [5]. If users refuse to provide some information for security reasons, the performance of the recommendation system may be degraded and even personalized services may not be provided. Therefore, it is very necessary to consider the privacy protection of users' information in the recommendation system.

In recent years, DP techniques have become a hot research topic by adding controlled noise to protect individual user privacy information without changing the overall pattern characteristics of the data. Santos-Lozada et al. [6] proposed a definition of DP. They provided a good theoretical basis for implementing effective privacy protection in recommender systems. Bao et al. [7] introduced DP protection into the CF technique and achieved DP protection by perturbing the item covariance matrix. Meng et al. [8] applied DP to the CF recommendation algorithm based on neighbors and achieved privacy protection by adding noise to the neighbor selection and similarity metric processes. Xu et al. [9] proposed two privacy-preserving schemes by adding Laplace noise to the original rating and user similarity metric processes, respectively. Mewada [10] proposed a DP-preserving neighbor-based CF algorithm for the privacy leakage problem faced by the k-nearest neighbor algorithm. For label-based recommendation systems, Wang et al. [11] proposed a DP-preserving algorithm for modifying and publishing user profiles. The algorithm is able to perform label recommendation and protect users' privacy within a certain accuracy loss.

For the MF-based recommendation algorithm, Hien and Gillis [12] perturbed the objective function of the MF algorithm under the consideration of the untrustworthiness of the recommendation system. They used the matrix of item-hidden factors with privacy protection implemented for the recommendation task. Shin et al. [13] proposed a personalized DP recommendation algorithm based on probability MF assuming that users have different degrees of privacy protection needs. Yu et al. [14] proposed a privacy MF scheme based on joint optimization by perturbing the objective function. Gao et al. [15] applied DP protection to the MF recommendation algorithm and designed three ways to add noise, i.e., in the input information, in the training process, and in the output information, respectively. Based on this idea, Yang et al. [16] designed 3 DP-preserving models on the SVD++ model.

Most of the current work implements DP protection by adding noise terms to various results of the MF process (e.g., gradient, hidden factor matrix, and objective function), and such schemes have the following problems: (1) higher noise: higher privacy protection requirements or sensitivities can increase the variance of the noise distribution, leading to the inclusion of excessive noise. (2) Nongeneralizability: the noise addition method may cause the final solution to be infeasible for constrained problems. (3) The importance of the hidden factor is not considered, which affects the algorithm's solving efficiency.

To address the abovementioned problems, this paper proposes a DP-preserving recommendation algorithm that fuses MF and the genetic algorithm. It introduces the genetic algorithm into the MF task so that the DP protection can be achieved by perturbing the selection process of candidate solutions without relying on the abovementioned method of adding noise. Meanwhile, the search for solutions in the genetic algorithm will be performed in the feasible domain, which can be easily extended to MF problems with constraints. Several sets of experimental results show that the algorithm in this paper can obtain higher accuracy on the basis of ensuring privacy protection and has better practical application value.

2. State of the Art

2.1. MF and CF. Usually, recommendation systems can be classified into three categories as follows: content-based recommendations, CF-based recommendations, and hybrid approach-based recommendations [17]. The specific classification is shown in Figure 1.

The recommendation based on CF is to analyze the user's historical behavior data to find people who have similar behavior with that of the user or items that are similar to the items they are interested in. Then, by continuously filtering out those items that do not interest them, the needs of that user are increasingly satisfied. CF generally uses the nearest neighbor technique. Its recommendation mechanism is to calculate the distance between users or items based on the user-item rating matrix and then find the nearest neighbors of the target user or item and make recommendations based on the nearest neighbors. According to whether machine learning ideas are applied, CF recommendation can be divided into neighborhood-based CF recommendation and model-based CF recommendation [18]. Figure 2 explains what is meant by CF through a schematic diagram.

As shown in Figure 2, the CF algorithm finds videos that users may like in an intuitive way by using the user's viewing history to find similar users who have watched the same videos as the target user, Sam. Then, it finds other videos that these similar users like to watch and recommends them to the target user Sam.

MF is a typical algorithm of the implicit semantic recommendation model, which is one of the CF methods belonging to the model-based approach [19]. Compared with the traditional memory-based CF recommendation methods for users and items, the model-based MF recommendation algorithm has a better recommendation effect. Moreover, MF can fully consider various factors that have an impact on the data and possesses very good scalability. MF algorithms are widely used because they are easy to implement and have excellent scalability. In the Netflix Prize competition, the recommendation based on MF has achieved very good recommendation results.

2.2. Research on the Application of DP in CF Recommendation Algorithms. Differential privacy (DP) is a privacy-preserving technique [20] that aims to protect individual

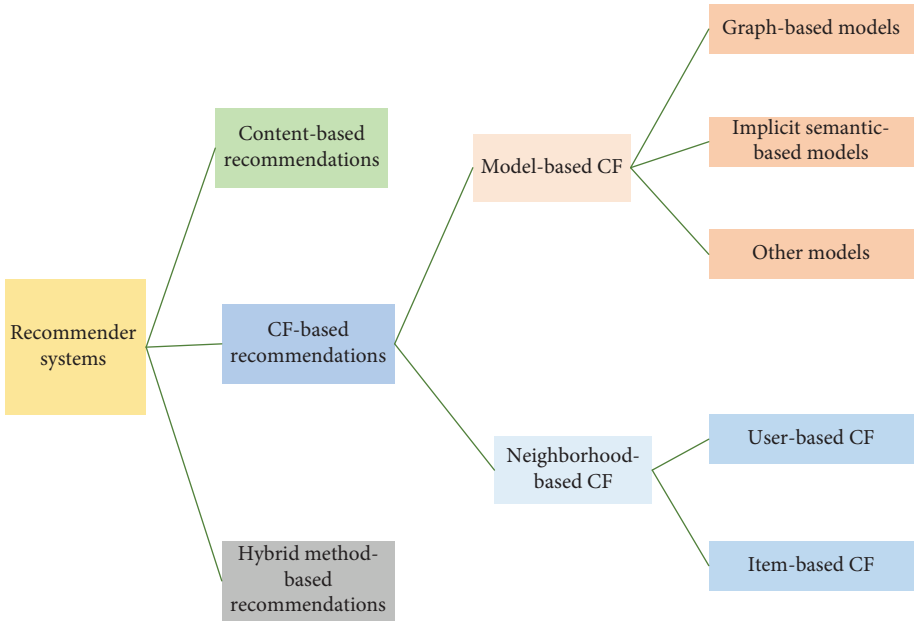

Figure 1: Classification of recommendation system algorithms.

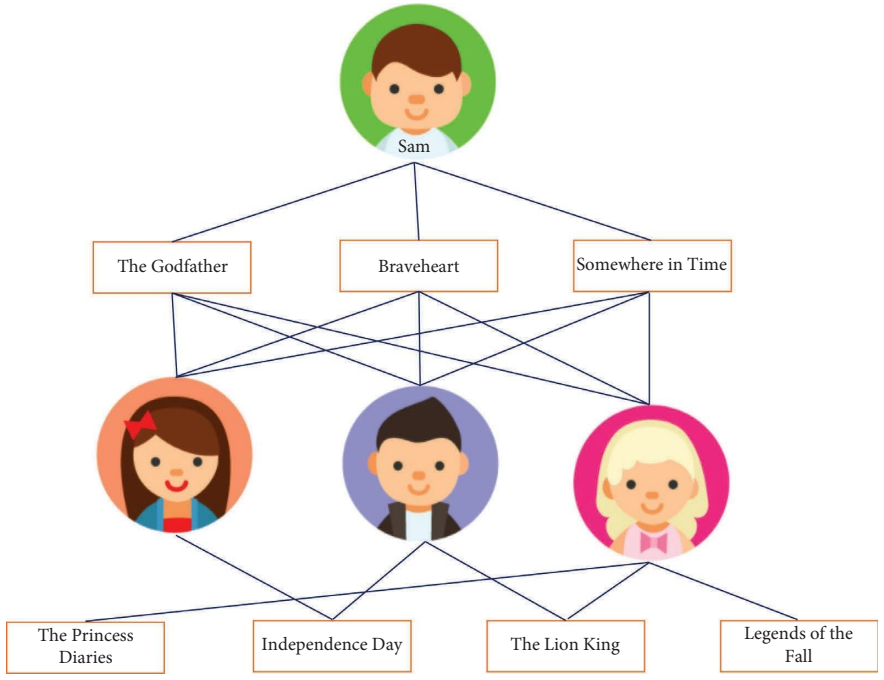

Figure 2: Schematic diagram of CF.

private information while allowing statistical analysis of data. The technique was first proposed by Li et al. [21] in 2006 and has been widely used in the fields of data publishing, machine learning, and recommender systems. The core principle of differential privacy is to introduce controlled random noise into the statistical results so that an attacker cannot infer any specific information about an individual from the output data. DP provides a powerful privacy-preserving mechanism that allows dataset publishers to share the aggregated results of the data without revealing sensitive information about individuals.

The exponential mechanism is an important tool for implementing randomization in differential privacy. Its basic principle is to introduce noise by calculating the sensitivity of each output to provide differential privacy protection. In the exponential mechanism, for each possible output, its sensitivity with respect to the input data is computed and randomly sampled according to the

probability distribution of the exponential distribution to obtain the final differential privacy output.

Differential privacy and exponential mechanisms have a wide range of applications in recommender systems. By introducing the randomization mechanism of differential privacy, recommender systems can provide users with accurate and personalized recommendation results while protecting their privacy. For example, in collaborative filtering-based recommendation algorithms, by introducing differential privacy and the exponential mechanism, noise can be added to the user rating matrix, thus realizing user privacy protection. Meanwhile, the recommender system can use the index mechanism to randomize the recommendation results according to the user's personalized preferences and increase the diversity of the recommendation results. Therefore, the application of DP technology to the protection of recommender systems has become a hot spot in the current research.

Wang et al. [22] proposed a recommendation scheme based on DP protection through the Laplacian mechanism of the DP recommendation scheme. However, in their scheme, they introduced a random perturbation method to process the user's historical data and generate useful data, which may have uncontrollable noise. Shen et al. [23] mentioned a federated differential privacy method for collaborative filtering. The method introduces the idea of federated learning in collaborative filtering. However, these issues need to be better handled as the difference in data distribution between different data sources, and the imbalance in data volume may lead to degradation in the generalization performance of the model. Hu et al. [24] proposed a new federated edge learning framework based on hybrid differential privacy and adaptive compression for industrial data processing. Experimental results show that the method is very effective in industrial edge computing scenarios and also opens up new directions for the effect of differential privacy in federated learning. However, the adaptive federated differential privacy technique proposed in this method may lead to an increase in noise or over elimination of noise while protecting individual privacy, affecting the accuracy of the recommendation algorithm. A better tradeoff between privacy protection and data quality is needed.

Matrix factorization (MF) is a commonly used technique in recommender systems, which maps users and items to a hidden factor space of the same dimension. By learning the hidden factors of users and items, the potential feature relationships between them can be extracted to achieve personalized recommendations. There are also many studies on the application of matrix decomposition methods in recommender systems; for example, Fan et al. [25] proposed a collaborative filtering model based on graph neural networks for heterogeneous graphs. They learned the complex relationship between users and items through graph neural networks and fully exploited the interaction information between users and items in the graph structure, thus improving the performance of the recommender system. However, the high computational complexity of graph neural networks when facing large-scale data may lead to a decrease in the efficiency of training and inference. Jing et al. [26] proposed a multiview fusion recommendation algorithm with an attentive deep neural network. The model designs a two-stage joint learning solution that combines user attributes, item attributes, and user-item interaction information into a unified framework. Experimental results on real datasets show that the algorithm achieves high recommendation accuracy even with extremely sparse data. The method fuses information from different views, thus improving the personalization effect of the recommender system. However, when faced with heterogeneous data with large differences in quality, the method may overfit low-quality views, thus affecting the recommendation effect.

The genetic algorithm (GA) is an optimization algorithm that can be used to solve complex optimization problems. In recent years, researchers have begun to apply genetic algorithms to recommender systems to improve the accuracy and diversity of recommendation algorithms. Alhijawi and Kilani [27] proposed a novel genetic-based recommender system. The system relies on semantic information and historical rating data. The experimental results demonstrate a more accurate prediction performance than other collaborative filtering recommender systems. However, the system may suffer from slow convergence, and better design of algorithm parameters and strategies is needed to improve the efficiency and convergence of the algorithm. Wei et al. [28] proposed a hybrid probabilistic multiobjective evolutionary algorithm for solving the cold-start problem. The method optimizes the recommendation results by the genetic algorithm to improve the recommendation accuracy in the case of cold start. However, it may be more sensitive to missing data when dealing with the cold-start problem and needs to better deal with data incompleteness to further improve the recommendation results.

3. Methodology

3.1. MF Algorithm. MF is the mapping of both users and items into the same d-dimensional hidden factor space [17]. The hidden factor corresponding to user p is denoted as $U_p \in \mathbb{R}^d$, and the matrix consisting of the hidden factors of all users is denoted as U. The hidden factor of item x is denoted as $V_x \in \mathbb{R}^d$, and the matrix consisting of the hidden factors of all items is denoted as V. MF is shown in Figure 3.

Then, the MF algorithm is to solve for best U and V satisfying the following equation:

$$\min f(U,V) = \min \sum_{(p,x)\in\mathscr{K}} \left(r_{px} - U_p^N V_x\right)^2, \tag{1}$$

where r_{px} is the rating of item x by user p in the user rating matrix and $\mathscr{K}$ is the set of user-item pairs corresponding to the observed rating data. Assuming that the number of users contained in r is w and the number of items is t, then there is $r \in \mathbb{R}^{w\times t}$, where $d \ll w, t$.

3.2. Differential Privacy. For any neighboring datasets D and D′ that differ by at most one data, and for all possible outputs O in the range of the randomized algorithm G, the differential privacy (DP) guarantee holds. (The O and G in this

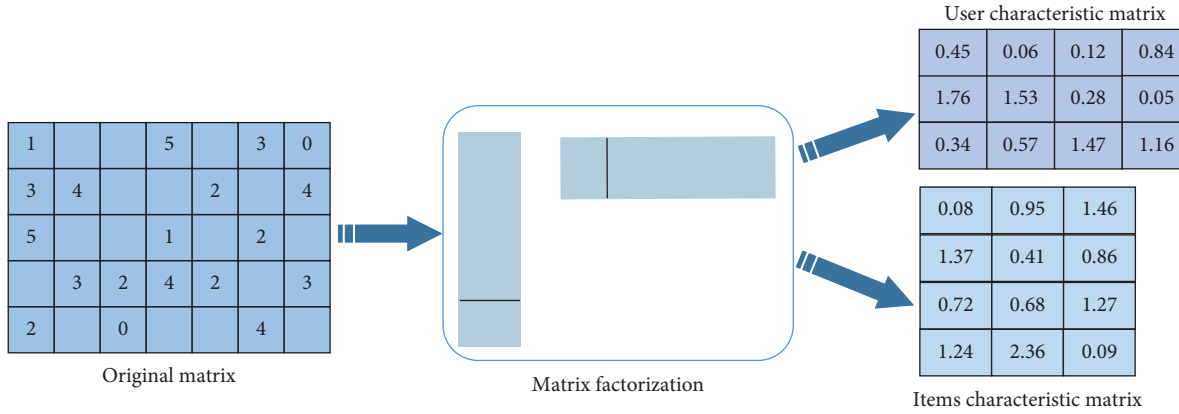

Figure 3: Schematic diagram of MF.

sentence are formulas). G satisfies ε-DP when and only if the following equation is satisfied:

$$Ur[G(D) \in O] \leqslant e^{\varepsilon} Ur\left[G\left(D^{'}\right) \in O\right], \tag{2}$$

where ε is the privacy budget, and the smaller the value of ε, the higher the level of demand for privacy protection.

3.2.1. Exponential Mechanism. The exponential mechanism [29] is a technical means to achieve DP protection, which is defined as follows.

Exponential mechanism: let the input of the randomized algorithm W be the dataset D and the output be $\omega \in \Omega$. The function $V(D, \omega) \longrightarrow \mathbb{R}$ is the availability function of ω. If the algorithm W selects and outputs ω from Ω with a probability proportional to $\exp(\varepsilon V(D, \omega)/\Delta)$, then algorithm W provides ε-DP protection and is said to be an exponential mechanism, where Δ is the damping factor of the availability function $V(D, \omega)$, also called the sensitivity of $V(D, \omega)$. It indicates the maximum impact of the difference of individual data on $V(D, \omega)$. We assume that D' and D are neighboring datasets, and Δ satisfies the inequality:

$$\Delta \geqslant 2 \max_{\omega \in \Omega, D, D^{'}} V(D, \omega) - V\left(D^{'}, \omega\right). \tag{3}$$

3.2.2. Enhanced Exponential Mechanism. In contrast to the exponential mechanism, the application of the augmented exponential mechanism is limited to availability functions with a specific form:

$$f(D, \omega) = b(\omega) + \sum_{n \in D} v(n, \omega), \tag{4}$$

where D is a dataset containing t tuples, $\mathcal{T}$ is the range of values of any tuple n, $v(n, \omega)$ is the tuple fit function, which indicates how well the model fits a single tuple n in D, and $b(\omega)$ is a function independent of the dataset D. Based on this availability function, the augmented exponential mechanism is defined as follows.

Enhanced exponential mechanism (EEM): let the input of the randomized algorithm W be the dataset D and the output be $\omega \in \Omega$. The algorithm W selects and outputs from Ω with a probability proportional to $\exp(\varepsilon f(D, \omega)/\Delta)$, where $f(D, \omega)$ satisfies equation (4), and Δ satisfies the following inequality:

$$\Delta \geqslant \min \left\{ \begin{array}{l} 2 \max\limits_{n, n^{'} \in \mathcal{T}, \omega \in \Omega} v(n, \omega) - v\left(n^{'}, \omega\right), \\ 2 \max\limits_{n \in \mathcal{T}, \omega, \omega^{'} \in \Omega} v(n, \omega) - v\left(n, \omega^{'}\right). \end{array} \right\}. \tag{5}$$

Then, the algorithm W provides ε-difference privacy protection and is called the enhanced exponential mechanism. From equation (5), it can be seen that the difference between the enhanced exponential mechanism and the standard exponential mechanism is that the damping factor Δ of the former takes into account the maximum difference between $v(n, \omega)$ and $v(n, \omega^{'})$. This is more suitable for the case where the degree of variation between solutions in the candidate solution set is relatively small due to the fact that max may obtain a smaller value at this time. The privacy-preserving scheme proposed in this paper will take advantage of this feature to improve the algorithm utility.

3.3. Privacy Genetic MF Algorithm. This algorithm improves the privacy genetic algorithm and proposes the adjusted private genetic algorithm (APrivGene). The APrivGene algorithm is used to solve the optimization problem shown below, and the privacy protection of the MF process is implemented by introducing an enhanced index mechanism in the selection phase. The APrivGene algorithm is presented in the order of execution in 3 aspects as follows: initialization, selection, and variation.

Initialization phase: each control parameter, including ε, is set. Then, ld-dimensional vectors are randomly generated as the initial candidate solution set Ω, and the objective function value of each solution is calculated as the fitness value of the genetic algorithm.

Selection phase: with $f(D, \omega)$ as the availability function and using $\varepsilon/2NA$ as the privacy budget for the selection operation, the enhanced exponential mechanism (EEM) is applied to select ω from Ω with probability proportional to $\exp(\varepsilon f(D, \omega)/2NA\Delta)$. To effectively mitigate the

perturbations introduced in the selection phase, only a single individual is selected for the subsequent operation, after which Ω is left empty and ready to admit new solutions.

Variation stage: to avoid excessive sensitivity caused by crossover operations, only variation operations are used. To improve the efficiency of the search, the variational perturbation is generated using the Corsi variational operator with high global search efficiency; that is, a random perturbation is generated from the standard Corsi distribution C (0,1). Then, with the aim of finding the most important hidden factor, let the variation operation vary for each hidden factor and search only on one dimension z at a time. Since the preference of a user or item for a hidden factor can be classified as positive or negative, perturbations on individual hidden factors are correspondingly designed in both positive and negative directions. The abovementioned variation is performed for each dimension, and two new solutions are generated for each variation and added to Ω, finally forming a new set of candidate solutions.

After generating the new set, the variation step size η is reduced using the decay factor β to gradually reduce the search range and improve the search efficiency. Then, the selection session is returned, and the next cycle is entered. When the maximum number of iterations A is reached, the final solution ω^* is selected using the EEM approach. The pseudocode of the abovementioned improved privacy genetic algorithm is shown in Algorithm 1.

The proposed algorithm revolves around the decomposition of the scoring matrix of the recommendation system and transforms the process of solving the hidden factor matrices U and V into two alternating optimization processes. A genetic algorithm is used to solve the optimization process, and an enhanced exponential mechanism is introduced in the solution process, which in turn makes the MF process satisfy DP protection. The general flow of the algorithm in this paper is as follows:

(1) To improve the rating prediction accuracy, the user rating matrix r is preprocessed. That is, the boundary parameter is set to B, and the ratings are transformed to the range of $[-H, H]$ to obtain the new user rating matrix R. Then, the matrix R is decomposed by the hidden factor:

$$U, V = \arg\min_{U,V} \sum_{(p,x)\in\mathcal{K}} \left(R_{px} - U_p^N V_x\right)^2. \quad (6)$$

Where R_{px} is the true rating of item x by user p in R. The objective of the hidden factor decomposition is to find the U and V matrices that minimize the sum of squared errors between the predicted and true ratings.

(2) The objective problem of equation (6) is converted into two types of feature solving tasks as follows: (1) solving the vector of hidden factors of users and (2) solving the vector of hidden factors of items. That is, in solving U_p, the matrix V is considered as a constant, and the objective function is constructed as follows:

$$f_V^p(D_p, U_p) = -\sum_{x\in X_p} \left(R_{px} - U_p^N V_x\right)^2 = \sum_{n\in D_p} v(n, U_p). \quad (7)$$

Where $v(n, \mathbf{U}_p) = -(R_{px} - \mathbf{U}_p^N \mathbf{V}_x)^2$ is a tuple fitting function for a single tuple $n = (V_x, R_{px})$, which characterizes the prediction effect on a single rating, $V_x = (V_{x1}, V_{x2}, \cdots, V_{xd})$ denotes the vector of hidden factors for item x, $D_p = \{(\mathbf{V}_x, R_{px}) | x \in X_p\}$ is the set of binary tuples about the user p, and X_p is the set of items evaluated by the user p. To guarantee the accuracy of rating prediction, upper and lower bounds are set on the hidden factor V_x: $|V_{xz}| \leqslant 1, z \in \{1, 2, \cdots, d\}$.

Similarly, in solving V_x, keeping the V matrix constant, the objective function is constructed as follows:

$$f_U^x(D_x, V_x) = -\sum_{p\in P_x} \left(R_{px} - U_p^N V_x\right)^2 = \sum_{n\in D_x} v(n, V_x). \quad (8)$$

Where $v(n, V_x) = -(R_{px} - U_p^N V_x)^2$, $n = (U_p, R_{px})$, and $U_p = (U_{p1}, U_{p2}, \cdots, U_{pd})$ are the vector of hidden factors for the user p.$D_x = \{(\mathbf{U}_p, R_{px}) \mid p \in P_x\}$ is the set of binary groups about the item x.P_x is the set of users who have evaluated the item x. We set upper and lower bounds on the hidden factor U_p: $|U_{pz}| \leqslant 1, z \in \{1, 2, \cdots\cdots, d\}$.

(3) First, keeping the matrix V constant, we use previously designed APrivGene to solve the optimization problem shown in equation (7) for each user, obtain the corresponding user hidden factor, and update the matrix U. Then, keeping the matrix U constant, we use previously designed APrivGene to solve the optimization problem shown in equation (8) for each item, obtain the corresponding item hidden factor, and update the matrix Q. The abovementioned process is repeated alternately to continuously optimize the matrices U and V until the maximum number of iterations N is reached.

The pseudocode of the above privacy genetic MF algorithm is shown in Algorithm 2.

The computational complexity of the proposed method was discussed and analyzed, summarized as follows.

In terms of time complexity, the method involves several crucial steps. The preprocessing step, where the user rating matrix R is transformed into a bounded range, takes $O(n)$ time, with n representing the number of nonzero elements in the rating matrix. The matrix factorization (the MF algorithm) contributes $O(nd)$ time, where d signifies the dimension of the hidden factor matrices U and V. The outer loop iteration, conducted N times, accounts for $O(N)$ time. In addition, calculating objective functions and obtaining binary groups have time complexities of $O(1)$ and $O(m)$, respectively, where m is the number of items evaluated by a user. The privacy genetic algorithm (APrivGene) used for obtaining hidden factors takes $O(Ald)$ time for each user and $O(n * Ald)$ time for each item, where A corresponds to the number of genetic algorithm iterations and l denotes the

Input: D, the set of binary groups D_p or D_x. f, the objective function f_q^p or f_p^x;
Output: a vector of hidden factors $\omega^* = (\omega_1, \omega_2, \cdots, \omega_d)$;
Control parameters in the initialization algorithm: set the number of hidden factors d, the privacy budget ε, the variation step η, the decay factor $\beta < 1$, the maximum number of iterations A, and the size l of the candidate solution set Ω;
The initial candidate solution set Ω is generated randomly;
For $a = 1$ to A-1 do
Compute $f(D, \omega)$ for each $\omega \in \Omega$;
$\omega = \mathrm{EEM}_f^e\ (D)$ select individuals using the augmented index mechanism;
Set Ω to null.
For $z = 1$ to d do
$i = C(0,1)$//draw random noise according to the standard Corsi distribution;
$q_1 = (\omega_1, \omega_2, \cdots, \omega_z + \eta i, \cdots, \omega_d)$// positive directional variation;
$q_2 = (\omega_1, \omega_2, \cdots, \omega_z - \eta x, \cdots, \omega_d)$// negative directional variation;
$\Omega = \Omega \cup \{q_1, q_2\}$
End for $\eta = \eta\beta$.
End for
Compute $f(D, \omega)$ for each $\omega \in \Omega$;
$\omega^* = \mathrm{EEM}_f^e\ (D)$
Return ω^*

Algorithm 1: Improved privacy genetic algorithm (APrivGene).

Input: user rating matrix r, user set *Users*, item set *Items*, number of iterations N.
Output: U, V;
The r matrix is preprocessed to obtain the homotopy matrix R. The optimization problem is established as in equation (6);
For $n = 1$ to N;
For p in Users;
Construct the objective function f_V^p;
$D_p = \{(V_x, R_{px}) | x \in X_p\}$// get the set of user binary groups;
$U_p = \mathrm{APrivGene}\ (D_p, f_V^p)$// solve for the users' hidden factor;
End for
For x in Items
Construct the subobjective function f_U^x;
$D_x = \{(U_p, R_{px}) | p \in P_x\}$// obtain the set of item binary groups;
$V_x = \mathrm{APrivGene}\ (D_x, f_U^x)$// solve for the item hidden factor;
End for
Return U, V

Algorithm 2: Privacy genetic MF algorithm.

population size. The total time complexity of the method is summarized as follows:

$$O(n + nd + N * (m * Ald) + N * (n * Ald)). \quad (9)$$

In terms of space complexity, the method requires storage for various components. The user rating matrix R', with bounded ratings, has a space complexity of $O(n)$. Meanwhile, the hidden factor matrices U and V occupy the $O(nd)$ space. Consequently, the total space complexity is

$$O(n + nd). \quad (10)$$

In conclusion, the proposed method has been thoroughly analyzed for its computational complexity. The time complexity assessment provides valuable insights into the method's efficiency in real-world scenarios, while the space complexity estimation helps in understanding the memory requirements for storing matrices and data during algorithm execution. This comprehensive understanding of computational complexities is crucial for evaluating the method's practical feasibility and potential applications.

In summary, this paper proposes a differential privacy-preserving recommendation algorithm that incorporates matrix decomposition and the genetic algorithm. The algorithm employs an augmented exponential mechanism to mitigate the degree of perturbation of the algorithm, thus better realizing the differential privacy protection. The method successfully solves the problems of privacy leakage and data security in recommender systems while maintaining the efficiency of the recommendation algorithm and the personalized features of the recommendation results. The following block diagram analyzes the function of each module in the model of this paper while explaining its role and the problem it solves (see Figure 4).

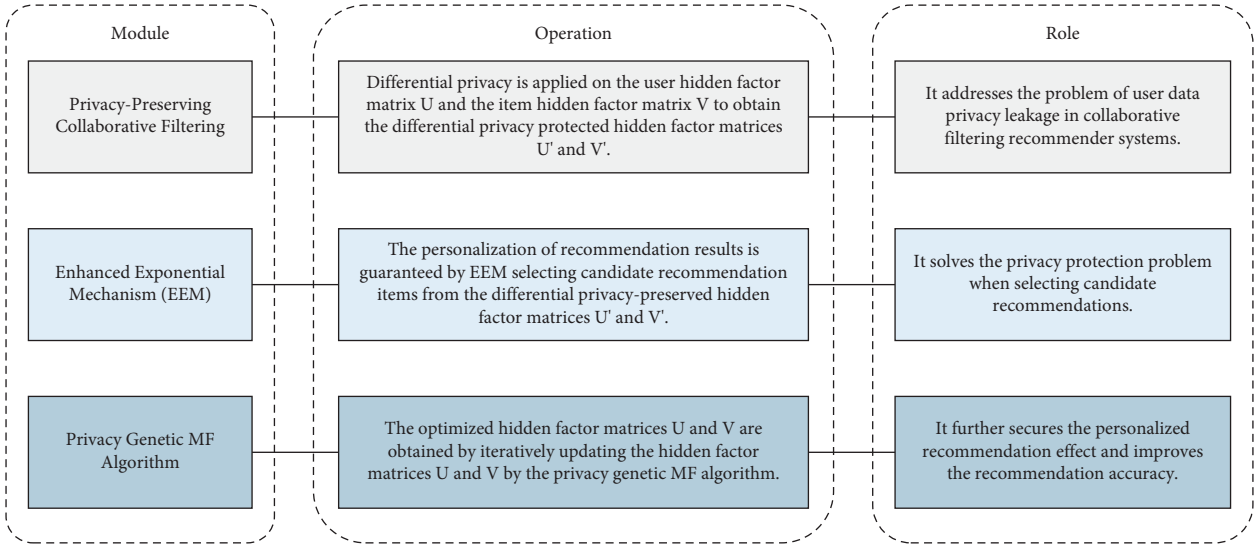

Figure 4: Analysis of the problems solved by the proposed method.

The hyperparameter setting of the model proposed in this paper is shown in Table 1. The table includes parameter settings related to matrix decomposition, genetic algorithm, and differential privacy. The application of these hyperparameters in the model, as well as their role and significance, is emphasized.

4. Result Analysis and Discussion

4.1. Experimental Setup. The hardware configuration selected for the experiments in this section is as follows: Intel i5-6400K 3.2 GHz, memory 8GHx, and system Win8 flagship version. To verify the effectiveness of the algorithm in this paper, four algorithms from literatures [29–32] are selected here as the comparison algorithms. The experimental objects of the algorithms in this paper are the data in Table 2.

The relevant parameters in Table 2 are parameter P which is the number of user sets in the experimental object, parameter S1 which is the number of relationships between users in the experimental object, parameter S2 which is the number of relationships between users and items in the experimental object, and the parameter item which is the number of items in the experimental object. In order to verify the usability of the DP recommendation results of the five algorithms, the experimental evaluation index selected is the NDCG index. It is defined in the following form:

$$\mathrm{NDCG}\ (z,p) = \sum_{p\in P} \frac{DCG(\widehat{L}(z),p)}{DCG(L(z),p)} \times \frac{1}{|P|}, \tag{11}$$

where the parameter term *NDCG* (*z*,*p*) is an evaluation metric for the usability recommendation of the DP item *z* by the user *p* in the experimental object. Its definition form is as follows:

$$\mathrm{DCG}\left(\widehat{L}(z),p\right) = \sum_{X_x\in L(z)} \frac{\mathrm{Rank}\left(p,X_x\right)}{\max\left(1,\log_2 \mathrm{index}\left(X_x\right)+1\right)}, \tag{12}$$

where the parameter term $index(X_x.)$ is the location index of the DP item X_x in the $\widehat{L}(z)$ dataset. Meanwhile, to evaluate the similarity of DP recommendations between the two datasets Flixster and Last.fm, the nearest neighbor relationship index is selected in this section.

$$\begin{aligned} \mathrm{Jaccard}\left(p,q\right) &= \frac{|\Gamma(p)\cap\Gamma(q)|}{|\Gamma(p)\cup\Gamma(q)|}, \\ \frac{\mathrm{Adamic}}{\mathrm{Adar}\ \ (p,q)} &= \sum_{k\in\Gamma(p)\cap\Gamma(q)} \frac{1}{\log\ |\Gamma(k)|}, \end{aligned} \tag{13}$$

where the parameter term $\Gamma(p)$ is the set of relational nearest neighbors of user p in the same dataset.

4.2. Usability Metric Evaluation. The two experimental datasets shown in Table 2 are used, and the evaluation metrics *NDCG* (*z*, *p*) for usability recommendations of different sizes are obtained by setting different item recommendation numbers *z* and privacy budgets *ε* during the experiments. The parameters $\varepsilon = \{0.1, 0.4, 0.7, 1.0\}$ and $z = \{10, 40, 70, 100\}$. The parametric experiments are as follows:

(1) Privacy budget *ε* influence experiment: the recommended number of DP in this experiment is set to $k = 40$ in the dataset, the privacy budget *ε* is chosen to be changed during the experiment, and the effectiveness of the algorithm is verified by using Jaccard metrics. Then, the NDCG evaluation metrics of the comparison algorithm on the selected experimental set are shown in Figures 5 and 6.

Table 1: Overview of hyperparameter settings.

Hyperparameters	Role and significance	Application in modeling
Differential privacy parameter (ε)	Controls the level of privacy protection, and smaller ε values indicate stronger privacy protection	Determining the noise size of the differential privacy mechanism
Number of iterations (N)	Controls the number of iterations of the optimization process	Controlling the number of genetic algorithm iterations, i.e., the number of iterations to update the hidden factor matrices U and V
Number of candidate recommendation items (l)	Achieve differential privacy protection through the number of recommendation terms selected by the exponential mechanism	Determine the privacy budget of the selection operation to achieve differential privacy protection of the recommendation results
Mutation step size (η)	Controls the step size of the mutation operation	Control the step size of the genetic algorithm to perturb the hidden factors during the search process
Mutation attenuation factor (β)	Controlling the degree of attenuation of the mutation operation	Controlling the diminishment of the mutation step size during the iteration of the genetic algorithm
Number of genetic algorithm iterations (A)	Controlling the number of iterations of the genetic algorithm	Controlling the number of iterations of the genetic algorithm
Privacy budget for selection operation(ε/2NA)	Realize differential privacy protection for the selection operation	Realize differential privacy protection of the selection operation to protect the privacy of the selection operation of the genetic algorithm

Table 2: Selected datasets.

Parameters	Datasets	
	Last.fm	Flixster
P	1778	137372
S1	11603	1269074
S2	91081	7527931
Item	16523	48754

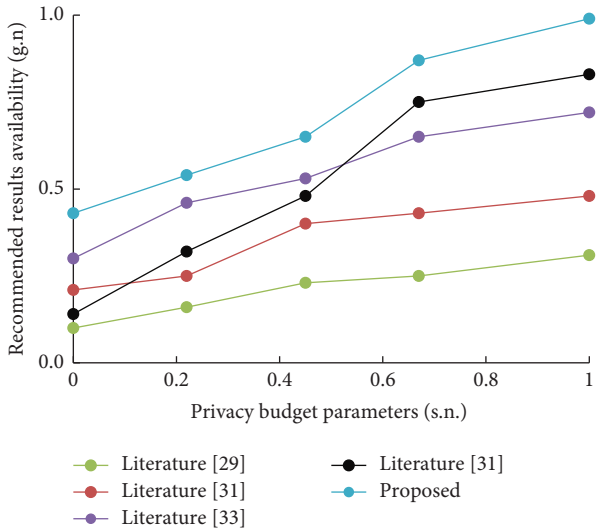

Figure 5: Changes of NDCG indicators on the experimental set (Last.fm).

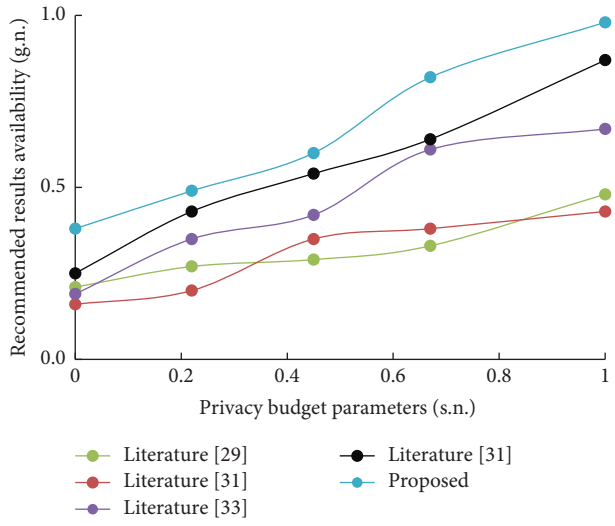

Figure 6: NDCG index changes on the experimental set (Flixster).

According to the experimental results in Figures 5 and 6, it can be seen that the trend of the indicator NDCG value changes when the experimental parameter privacy budget ε is varied in the interval (0.1, 1.0). The main reason is that the larger the value of the privacy budget ε, the smaller the value of the Laplace noise required in the implementation of the algorithm. Since the algorithms are based on the traditional optimization methods of literature [30] and literature [29], when the privacy budget ε is smaller, the noise introduced is larger, resulting in a large gap between the solved hidden factor vector and the optimal solution, and the accuracy of the recommendation is reduced. However, the algorithm in this paper uses the server gradient perturbation of user privacy, so the NDCG value that it obtains is larger and the recommended accuracy is higher.

(2) The parameter m changes affect the experiment. In this part of the experimental session, the privacy budget is set to a fixed value, i.e., $\varepsilon = 0.4$. For different values of the number of DP recommendations m, the Jaccard metric is selected to verify the effectiveness of the algorithm. The usability recommendation results of different algorithms on the selected experimental set are shown in Figures 7 and 8.

According to the experimental results in Figures 7 and 8, it can be seen that the NDCG indices of the centralized comparison algorithms all show a monotonically increasing trend when the number of DP recommendation items is gradually increased from 10 to 100. The results show that the NDCG indices of the algorithms in this paper are kept above 90% on the selected experimental datasets, and the algorithms in literature [32] can be kept above 80%, the algorithm in literature [31] is kept above 70%, and the algorithms in literature [29, 30] are only maintained at less than 50%. The main reason is that the algorithm in this paper has low noise among the project users for the experimental dataset.

4.3. Algorithm Comparison Test. The abovementioned four algorithms are selected as the comparison calculation method, and the comparison index object is shown in Table 2. Algorithm calculation time and privacy data recovery accuracy are selected as the comparison index to verify the data recommendation quality of the algorithm, and the experimental results are shown in Figure 9.

In order to evaluate the performance of the algorithms more consistently, the average computational metrics of 10 experimental runs are selected here as the comparison results. As can be seen from Figure 9, in terms of the calculation time index, the DP recommendation time of this algorithm on the Last.fm test set is about 2.5 s, while the algorithms in literature [30] and literature [32] are 12.5 s, 10 s, 8 s, and 5.5 s, respectively. The calculation time of this algorithm is improved by 80%, 75%, 68.8%, and 54.5%, respectively, compared with the other four algorithms. In the recommended accuracy index, the recommended accuracy of this algorithm is 99%, which is 4.5%, 6.2%, 11.4%, and 13.6% higher than that of the four algorithms in literature [29–31] and [32]. This indicates that the DP recommendation efficiency and recommendation quality of this paper's algorithm on the Last.fm test set are better than those of the four selected comparison algorithms. Meanwhile, the data on the Flixster test set prove that this paper's algorithm also outperforms the other four comparison algorithms in the literature, showing similar performance.

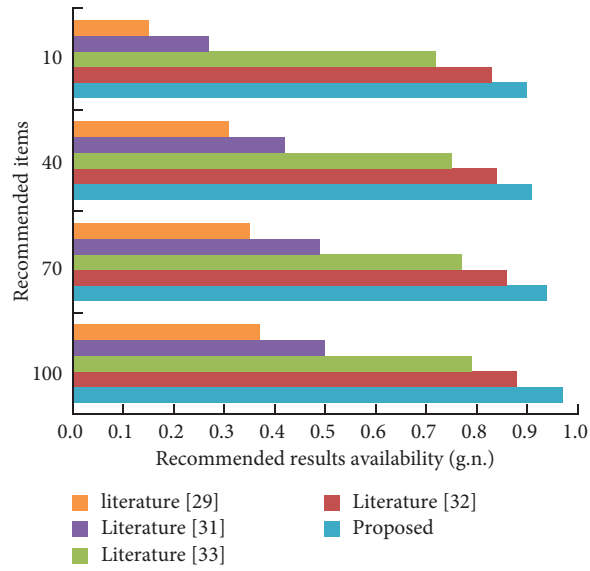

Figure 7: Recommended results on Last.fm.

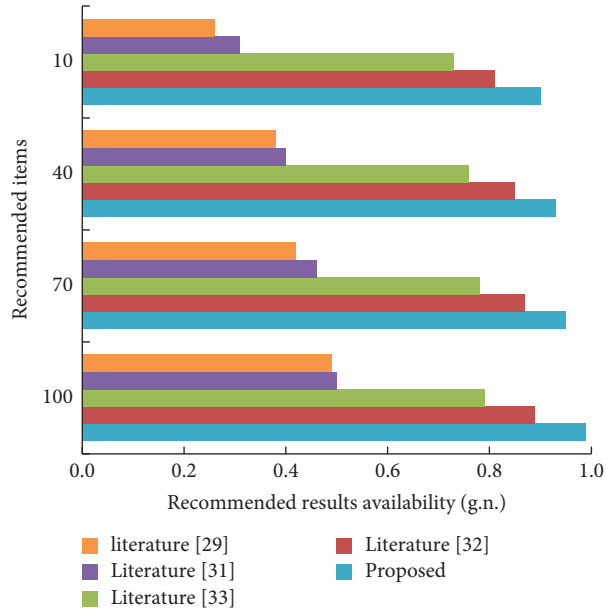

Figure 8: Recommendation results on Flixster.

4.4. Effect of Different Privacy Budgets on Recommendation Results. The purpose of this set of experiments is to examine the impact of this paper's algorithm on the recommendation results while achieving DP protection for users (see Figure 10). The root mean square error (RMSE) is used to measure the performance of the algorithm. The privacy budget ε of this algorithm is 5. As can be seen in Figure 10, when $\varepsilon < 1$, the accuracy of this algorithm is smaller than that of the underlying literature's [30] algorithm, which makes the algorithm unusable. This is because when ε is small, the algorithm adds a large amount of noise and thus affects the results of the model. When $1 < \varepsilon < 3$, the accuracy of the proposed algorithm is between that of literature [30] and literature [29]. When $3 < \varepsilon < 5$, the accuracy of the algorithm in this paper is between literature [29] and literature [31]. When $5 < \varepsilon < 6$, the accuracy of the proposed algorithm is between literature [31] and literature [32]. When $\varepsilon > 6$, the accuracy of this algorithm is higher than that of literature [32]. ε is still one of the difficulties in DP research, and it is generally believed that the smaller ε is, the higher the degree of privacy protection is.

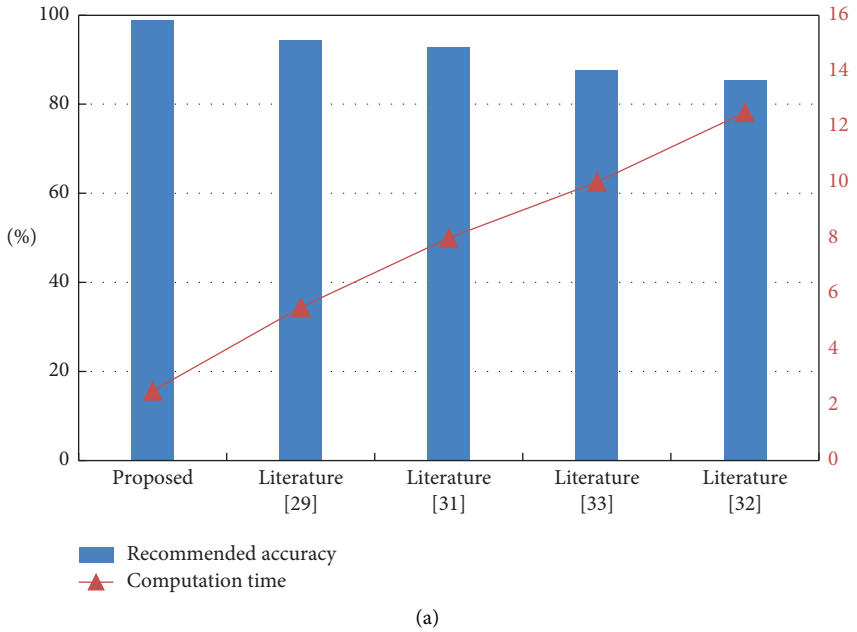

Figure 9: Comparison of test results of different algorithms. (a) Last.fm test set. (b) Flixster test set.

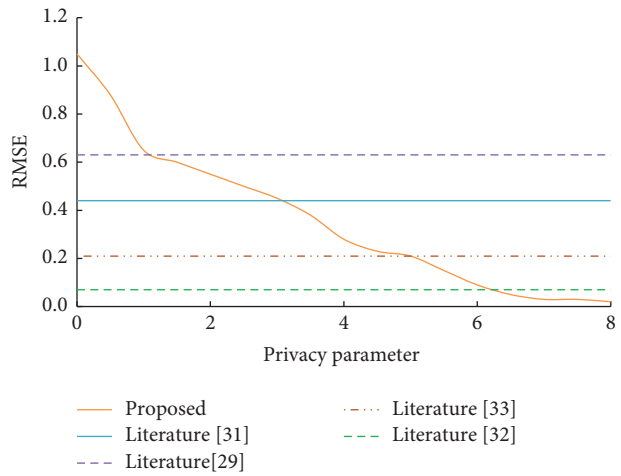

Figure 10: Impact of privacy protection budget ε on accuracy.

5. Conclusion

In this paper, a privacy-preserving recommendation algorithm based on MF and the genetic algorithm is proposed for the privacy problem in recommendation systems. The algorithm transforms MF into two alternating user-hidden factor and item-hidden factor optimization problems, which effectively overcomes the problems of high dimensionality of the solution space and nonconvexity in optimization in the solution process. Moreover, an enhanced exponential mechanism is used in the selection operation of the genetic algorithm to make the whole MF process satisfy the DP protection. However, based on the idea of searching for significant hidden factors, the mutation operation of the genetic algorithm is redesigned to mutate the hidden factors from both positive and negative directions. This not only improves the efficiency of the algorithm but also effectively enhances the performance of the understanding. Experimental results on two standard datasets show that the algorithm in this paper can better balance privacy and recommended accuracy. Especially under the condition of high privacy protection requirements, it can still achieve a good recommendation effect and has good application potential.

The proposed model exhibits two limitations that require further research. (1) Privacy budget selection: the model relies on the privacy budget parameter (ε) to control the level of privacy protection for users. However, determining the appropriate privacy budget for individual users poses a significant challenge. Setting the same privacy budget for all users may not adequately cater to varying privacy concerns among users. Future research needs to address the issue of dynamically selecting privacy budget parameters based on users' privacy preferences and risk tolerance. (2) Individual user characteristics: the model assumes a uniform privacy protection level for all users, overlooking the fact that different users may have distinct privacy requirements. Future research should explore methods to capture individual user attributes, such as privacy preferences, risk tolerance, and past privacy-related behaviors. By integrating such information into the model, personalized privacy protection strategies can be devised, ensuring that users feel comfortable and confident in sharing their data while receiving personalized recommendations.

Data Availability

The dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

Sheng Hu contributed to the writing of the manuscript and data analysis. Ming He made important contributions in the revision, provided great help in the revision of the final draft, and agreed to be included in the author list of the article. All authors unanimously agreed to the above arrangement. All the authors have read and agreed the final version to be published.

References

[1] Z. Cui, X. Xu, F. Xue et al., "Personalized recommendation system based on collaborative filtering for IoT scenarios," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 685–695, 2020.

[2] Q. Guo, F. Zhuang, C. Qin et al., "A survey on knowledge graph-based recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3549–3568, 2022.

[3] F. Fkih, "Similarity measures for collaborative filtering-based recommender systems: review and experimental comparison," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 7645–7669, 2022.

[4] Z. Han, Y. Fan, G. Chen, and T.. Zhou, "TeCNTS: A robust collaborative filtering recommendation scheme based on time-effective close neighbor trusted selection strategy," in *Proceedings of the 2022 Tenth International Conference on Advanced Cloud and Big Data (CBD)*, pp. 194–199, IEEE, Guilin, China, November 2022.

[5] T. Li, Y. Wang, Y. Ren, Y. Ren, Q. Qian, and X. Gong, "Nonnegative MF-based privacy-preserving collaborative filtering on cloud computing," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 6, 2022.

[6] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and D. Megías, "Individual differential privacy: a utility-preserving formulation of differential privacy guarantees," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1418–1429, 2017.

[7] T. Bao, L. Xu, L. Zhu, L. Wang, R. Li, and T. Li, "Privacy-preserving collaborative filtering algorithm based on local differential privacy," *China Communications*, vol. 18, no. 11, pp. 42–60, 2021.

[8] S. Meng, S. Fan, Q. Li et al., "Privacy-aware factorization-based hybrid recommendation method for healthcare services," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5637–5647, 2022.

[9] C. Xu, A. S. Ding, and S. S. Liao, "A privacy-preserving recommendation method based on multi-objective optimisation for mobile users," *International Journal of Bio-Inspired Computation*, vol. 16, no. 1, pp. 23–32, 2020.

[10] S. Mewada, "Data mining-based privacy preservation technique for medical dataset over horizontal partitioned," *International Journal of E-Health and Medical Communications*, vol. 12, no. 5, pp. 50–66, 2021.

[11] H. Wang, J. Zhang, C. Lu, and C. Wu, "Privacy preserving in non-intrusive load monitoring: a differential privacy perspective," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2529–2543, 2021.

[12] L. T. K. Hien and N. Gillis, "Algorithms for nonnegative matrix factorization with the kullback–leibler divergence," *Journal of Scientific Computing*, vol. 87, no. 3, p. 93, 2021.

[13] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1770–1782, 2018.

[14] X. Yu, D. Zhan, L. Liu, H. Lv, L. Xu, and J. Du, "A privacy-preserving cross-domain healthcare wearables recommendation algorithm based on domain-dependent and domain-

independent feature fusion," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 1928–1936, 2022.

[15] C. Gao, C. Huang, Y. Yu, H. Wang, Y. Li, and D. Jin, "Privacy-preserving cross-domain location recommendation," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, pp. 1–21, 2019.

[16] Q. Yang, A. Huang, L. Fan et al., "Federated learning with privacy-preserving and model IP-right-protection," *Machine Intelligence Research*, vol. 20, no. 1, pp. 19–37, 2023.

[17] D. Bahl, V. Kain, A. Sharma, and M. Sharma, "A novel hybrid approach towards movie recommender systems," *Journal of Statistics & Management Systems*, vol. 23, no. 6, pp. 1049–1058, 2020.

[18] A. Noulapeu Ngaffo and Z. Choukair, "A deep neural network-based collaborative filtering using a matrix factorization with a twofold regularization," *Neural Computing & Applications*, vol. 34, no. 9, pp. 6991–7003, 2022.

[19] C. Tang and J. Zhang, "An intelligent deep learning-enabled recommendation algorithm for teaching music students," *Soft Computing*, vol. 26, no. 20, pp. 10591–10598, 2022.

[20] J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "The limits of differential privacy (and its misuse in data release and machine learning)," *Communications of the ACM*, vol. 64, no. 7, pp. 33–35, 2021.

[21] J. Li, H. Ye, T. Li et al., "Efficient and secure outsourcing of differentially private data publishing with multiple evaluators," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 67–76, 2022.

[22] T. Wang, X. Zhang, J. Feng, and X. Yang, "A comprehensive survey on local differential privacy toward data statistics and analysis," *Sensors*, vol. 20, no. 24, p. 7030, 2020.

[23] S. Shen, T. Zhu, D. Wu, W. Wang, and W. Zhou, "From distributed machine learning to federated learning: in the view of data privacy and security," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 16, 2022.

[24] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[25] W. Fan, Y. Ma, Q. Li et al., "A graph neural network framework for social recommendations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2033–2047, 2022.

[26] W. Jing, A. K. Sangaiah, L. Wei, L. Shaopeng, L. Lei, and L. Ruishi, "Multi-view fusion for recommendation with attentive deep neural network," *Evolutionary Intelligence*, vol. 15, no. 4, pp. 2619–2629, 2022.

[27] B. Alhijawi and Y. Kilani, "A collaborative filtering recommender system using genetic algorithm," *Information Processing & Management*, vol. 57, no. 6, Article ID 102310, 2020.

[28] G. Wei, Q. Wu, and M. Zhou, "A hybrid probabilistic multiobjective evolutionary algorithm for commercial recommendation systems," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 3, pp. 589–598, 2021.

[29] G. Beigi and H. Liu, "A survey on privacy in social media: identification, mitigation, and applications," *ACM/IMS Transactions on Data Science*, vol. 1, no. 1, pp. 1–38, 2020.

[30] L. Zhang, Y. Liu, R. Wang, X. Fu, and Q. Lin, "Efficient privacy-preserving classification construction model with differential privacy technology," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 170–178, 2017.

[31] C. Wang, Y. Zheng, J. Jiang, and K. Ren, "Toward privacy-preserving personalized recommendation services," *Engineering*, vol. 4, no. 1, pp. 21–28, 2018.

[32] R. Bosri, M. S. Rahman, M. Z. A. Bhuiyan, and A. Al Omar, "Integrating blockchain with artificial intelligence for privacy-preserving recommender systems," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1009–1018, 2021.

[33] C. Xu, J. Wang, L. Zhu, C. Zhang, and K. Sharif, "PPMR: a privacy-preserving online medical service recommendation scheme in eHealthcare system," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5665–5673, 2019.