applied sciences

MDPI

Article

Personalized Privacy Protection-Preserving Collaborative Filtering Algorithm for Recommendation Systems

Bin Cheng [1], Ping Chen [1], Xin Zhang [1], Keyu Fang [1], Xiaoli Qin [1,*] and Wei Liu [2,*]

[1] Medical Support Technology Research Department, Systems Engineering Institute, Academy of Military Sciences, PLA, Tianjin 300161, China
[2] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100085, China
* Correspondence: qinxl9695@126.com (X.Q.); liuwei_bit@126.com (W.L.); Tel.: +86-15320073712 (X.Q.); +86-15320073717 (W.L.)

Abstract: With the rapid development of ubiquitous data collection and data analysis, data privacy in a recommended system is facing more and more challenges. Differential privacy technology can provide strict privacy protection while reducing the risk of privacy leakage, but it also introduces unwanted noise, which makes the performance of the recommender system worsen. Among different users, the degree of their sensitivity to privacy is usually different. Thus, through considering the impact of users' personalized requirements, the collaborative filtering algorithm can be designed to reduce the amount of unwanted noise. Taking the above assertions into account, we propose a collaborative filtering algorithm based on personalized privacy protection. First, it locally classifies ratings by privacy sensitivity on the user side, then utilizes the random flip mechanism to protect the privacy-sensitive ratings. Then, after the server catches the perturbed rating data, we reconstruct the joint item-item distribution through the Bayesian estimation method. Experimental results show that our proposed algorithm can significantly improve the recommendation performance of recommendation systems while protecting users' privacy.

Keywords: recommendation system; differential privacy; collaborative filtering algorithm; privacy leakage

Citation: Cheng, B.; Chen, P.; Zhang, X.; Fang, K.; Qin, X.; Liu, W. Personalized Privacy Protection-Preserving Collaborative Filtering Algorithm for Recommendation Systems. *Appl. Sci.* **2023**, *13*, 4600. https://doi.org/10.3390/app13074600

Academic Editor: Shoujin Wang

Received: 24 February 2023
Revised: 2 April 2023
Accepted: 3 April 2023
Published: 5 April 2023

1. Introduction

Recommendation systems are widely used in real-life recommendation tasks as an effective method to solve the issue of information overload.

Collaborative filtering algorithm is the most classic and commonly used in the field of recommendation system. The basic idea of collaborative filtering algorithm is to recommend items to users based on their previous preferences and the choices of other users with similar interests, that is, based on users' historical behavioral data to discover users' preferences and to predict the items that users may like. It does not rely on any items' additional information (e.g., their own characteristics) or any users' additional information (e.g., age, gender, etc.). Currently, the most widely used collaborative filtering algorithm is the neighborhood-based method, which includes two main types of algorithms. One type is the user-based collaborative filtering algorithm (UserCF), which recommends products to users according to others has the similar interests as them, the other is the item-based collaborative filtering algorithm (ItemCF), which recommends items which are similar to their previously preferred items. Additionally, recommender systems usually need to collect users' personal information and interaction records to train models and perform high-quality product recommendations [1,2]. Traditional recommender systems assume that platform parties and users are completely trustworthy with each other. However, the reality is often not as ideal as we think, there often exists the risk of privacy leakage in real scenarios, and this risk exists in multiple ways, including between users and platforms, between users and users, and between platforms and platforms. This leads to the risk

of leakage of users' private information. Collaborative filtering algorithms are widely applied in various fields; however, these algorithms extensively use users' historical information in the process of preference calculation, which may lead to the risk of privacy leakage [3–5]. Therefore, privacy protection in recommendation algorithms is of great concern to users. To address the issue of privacy leakage during recommendation, many studies have tried to bring the differential privacy-preserving strategy into collaborative filtering algorithms [6]. Some typical algorithms include one designed by Zhu et al. [7], which is an item-based differential privacy recommendation algorithm and a user-based differential privacy recommendation algorithm, which can effectively solve the issue of privacy leakage of collaborative filtering algorithms, but due to its neglect of the sensitivity of user rating data, a large amount of noise data is introduced, resulting in unsatisfactory algorithm efficiency, and at the same time, the user-based strategy and the project-based strategy are independent to each other, so advantages of algorithms are not fully released. Wang et al. [8] combined the Bhattacharyya similarity with the K-medoids clustering to improve the accuracy of the similarity measure of the differential privacy-based collaborative filtering algorithms. Meanwhile, the differential privacy mechanism also introduces a lot of noise data that impairs the performance of the recommendation system. How to trade-off between algorithmic performance and the privacy security has become one of the hot spots in the field of differential privacy-preserving recommendation algorithms.

To better protect users' privacy security, many scholars have tried to introduce the method of data reconstruction, which is an evaluation algorithm based on the probabilistic perturbation mechanism. Traditional evaluation algorithms can only reconstruct the distribution of a single vector; thus, Rade et al. [9] proposed a joint cardinality estimation method to reconstruct the joint distribution of multiple vectors. Then, Chen et al. [10] introduced the joint cardinality estimation method into the neighbor-based collaborative filtering algorithm, which effectively improved the performance of the recommendation system while protecting the security of users' privacy. Guo et al. [11,12] proposed a joint frequency estimation algorithm based on the random response mechanism, and applied it to the neighbor-based collaborative filtering algorithm to protect the security of users' privacy. However, the estimation results of these joint estimation algorithms mentioned above are sometimes negative, which can seriously affect the accuracy of the distribution of rating prediction. Furthermore, Ren et al. [13] proposed a distributed reconstruction method for the high-dimensional crowdsourced data based on the random response mechanism, which could greatly improve the release accuracy of the high-dimensional crowdsourced data. However, there exists great differences in structure between the crowdsourced data and the rating data, resulting in the reconstruction method based on high-dimensional crowdsourced data cannot be directly applied to the recommendation system. In addition, different users have different sensitivity to privacy, and ignoring differences between users not only does not meet the user's personalized privacy requirements, but also introduces additional noise into the data, thereby reducing the performance of the recommendation system.

To address the issue of user privacy leakage, this paper proposes a collaborative filtering algorithm based on personalized privacy protection. Our main contributions include: (1) providing a personalized privacy sensitivity classification and coding method, which can effectively reduce the noise added by the privacy protection; and (2) propose a Bayesian joint distribution estimation method through creatively applying the Bayesian algorithm to the similarity calculation of recommendation systems.

2. Collaborative Filtering Algorithm Based on Personalized Privacy Protection

To better protect the security of users' privacy, while reducing the impact of noise on the performances of the recommendation algorithms, this paper proposes a personalized collaborative filtering-based recommendation algorithm with the differential privacy protection, and the algorithmic architecture is shown in Figure 1.

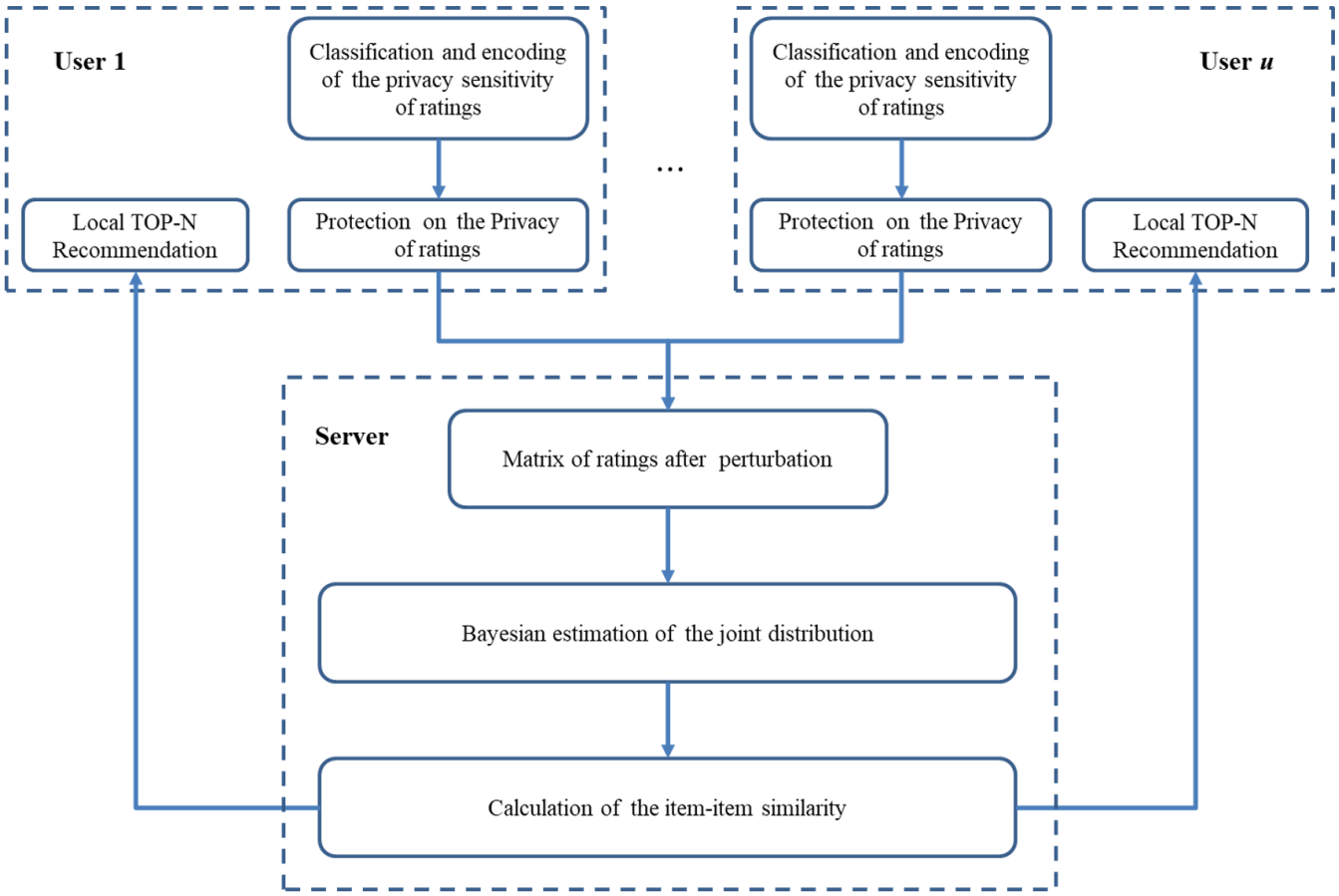


Figure 1. Overview of Algorithmic architecture.

From Figure 1, each user locally classifies the privacy sensitivity of ratings and encodes it, then perturbs the ratings and sends the vector of perturbed ratings to the server. According to all the perturbed ratings, the server first utilizes the Bayesian estimation to reconstruct the joint distribution of items, then calculates the item-item similarity and sends the results to users. Finally, it locally completes the high-quality, personalized recommendation.

2.1. Evaluation and Encoding of the Privacy Sensitivity of User Ratings

For recommendation systems, malicious attackers often rely on users' rating data to infer and obtain user privacy. Numerous studies [14–16] have proved that there exists a certain correlation between the size of the ratings and user privacy, and too high and too low ratings tend to be more favorable for malicious attackers to infer users' private information, such as user preferences. On the other side, from the user's perspective, both users' rating habits and the degree of the sensitivity to privacy protection are different. Based on the above two factors, the classification rules of the privacy sensitivity towards users' ratings are proposed as follows:

$$r_{ui} = \begin{cases} sensitive\ rating, & \text{when } r_{ui} \in [r_{min}, \overline{r_u} - \gamma_u] \cup [\overline{r_u} + \gamma_u, r_{max}]; \\ \text{weakly} - sensitive\ rating, & when\ r_{ui} \in (\overline{r_u} - \gamma_u, \overline{r_u} + \gamma_u); \end{cases} \quad (1)$$

where r_{min} and r_{max} represent the maximum and minimum values specified by the rating rules, respectively; $\overline{r_u}$ denotes the average score of user u; γ_u is an adjustable variable to adjust the range of intervals classified.

Depending on the sensitivity type of ratings, the encoding function E is

$$E(r_{ui}) = \begin{cases} b_{ui}^{s+}, & when \quad r_{ui} \in (\overline{r_u} + \gamma_u, r_{max}); \\ b_{ui}^{ns}, & when \quad r_{ui} \in (\overline{r_u} - \gamma_u, \overline{r_u} + \gamma_u); \\ b_{ui}^{s-}, & when \quad r_{ui} \in (r_{min}, \overline{r_u} - \gamma_u); \end{cases} \tag{2}$$

where b_{ui}^{s+}, b_{ui}^{ns}, and b_{ui}^{s-} are 1, 0, and -1, respectively, that is, the encoding function encodes the high-sensitivity score as 1, the low-sensitivity score as -1, and the weakly sensitive score as 0.

The rating of u constitutes the rating vector of $R_{u.} = \{r_{u1}, r_{u2}, r_{u3}, \ldots, r_{un}\}$. According to the function E, $R_{u.}$ is encoded to be the sensitive rating vector $B_{u.}^{s}$ and weakly sensitive rating vector $B_{u.}^{ns}$

$$B_{u.}^{s} = \left\{b_{ua_1}^{s},\ b_{ua_2}^{s}, b_{ua_3}^{s}, \ldots, b_{ua_x}^{s}\right\}$$

$$B_{u.}^{ns} = \left\{b_{uc_1}^{ns},\ b_{uc_2}^{ns}, b_{uc_3}^{ns}, \ldots, b_{uc_{n-x}}^{ns}\right\}$$

where $a = \{a_1, a_2, a_3, \ldots, a_x\}$ represents the collection of items corresponding to the sensitive rating, and $c = \{c_1, c_2, c_3, \ldots, c_{n-x}\}$ represents the collection of items corresponding to the weakly sensitive rating. Additionally, $b_{ua_i}^{s} \in \left\{b_{ua_i}^{s+}, b_{ua_i}^{s-}\right\}$.

2.2. Differential Privacy Protection for Rating Data

To protect the security of the encoded rating data, users need to locally perturb them. Based on the random flip mechanism, the perturbation function for the sensitive ratings is shown below:

$$b_{ua_i}^{\widehat{s}} = \varnothing\left(b_{ua_i}^{s}\right) = \begin{cases} b_{ua_i}^{s}, & if\ y > p; \\ b_{ua_i}^{\widetilde{s}}, & if\ y \le p; \end{cases} \tag{3}$$

where $p = \frac{1}{1+e^{\varepsilon}}$, $q = 1 - p$, and ε represents the privacy budget, respectively, and y represents one random number evenly distributed within [0, 1]. The variable $b_{ua_i}^{\widetilde{s}}$ is the result of the inverse operation on $b_{ua_i}^{s}$. $b_{ua_i}^{\widehat{s}}$ is the output of $b_{ua_i}^{s}$ being perturbed. In the perturbation function of $\varnothing$, $b_{ua_i}^{s}$ flips with the probability of p and does not flip with the probability of q, i.e.,

$$pr\left(b_{ua_i}^{s+} \to b_{ua_i}^{s+}\right) = pr\left(b_{ua_i}^{s-} \to b_{ua_i}^{s-}\right) = q$$

$$pr\left(b_{ua_i}^{s-} \to b_{ua_i}^{s+}\right) = pr\left(b_{ua_i}^{s+} \to b_{ua_i}^{s-}\right) = p$$

Based on the encoding function E and the perturbation function $\varnothing$, Figure 2 shows the privacy protection operation for the local rating data. First, encode the rating vector $R_{u.}$ of user u to be $B_{u.}^{s}$ and $B_{u.}^{ns}$ through the function E. Second, use the perturbation function $\varnothing$ to perturb each element within $B_{u.}^{s}$ to get the perturbed sensitive rating vector $\widehat{B_{u.}^{s}}$. Finally, stitch $\widehat{B_{u.}^{s}}$ and $B_{u.}^{ns}$ into $\widehat{B_{u.}}$, and send $\widehat{B_{u.}}$ to the server.

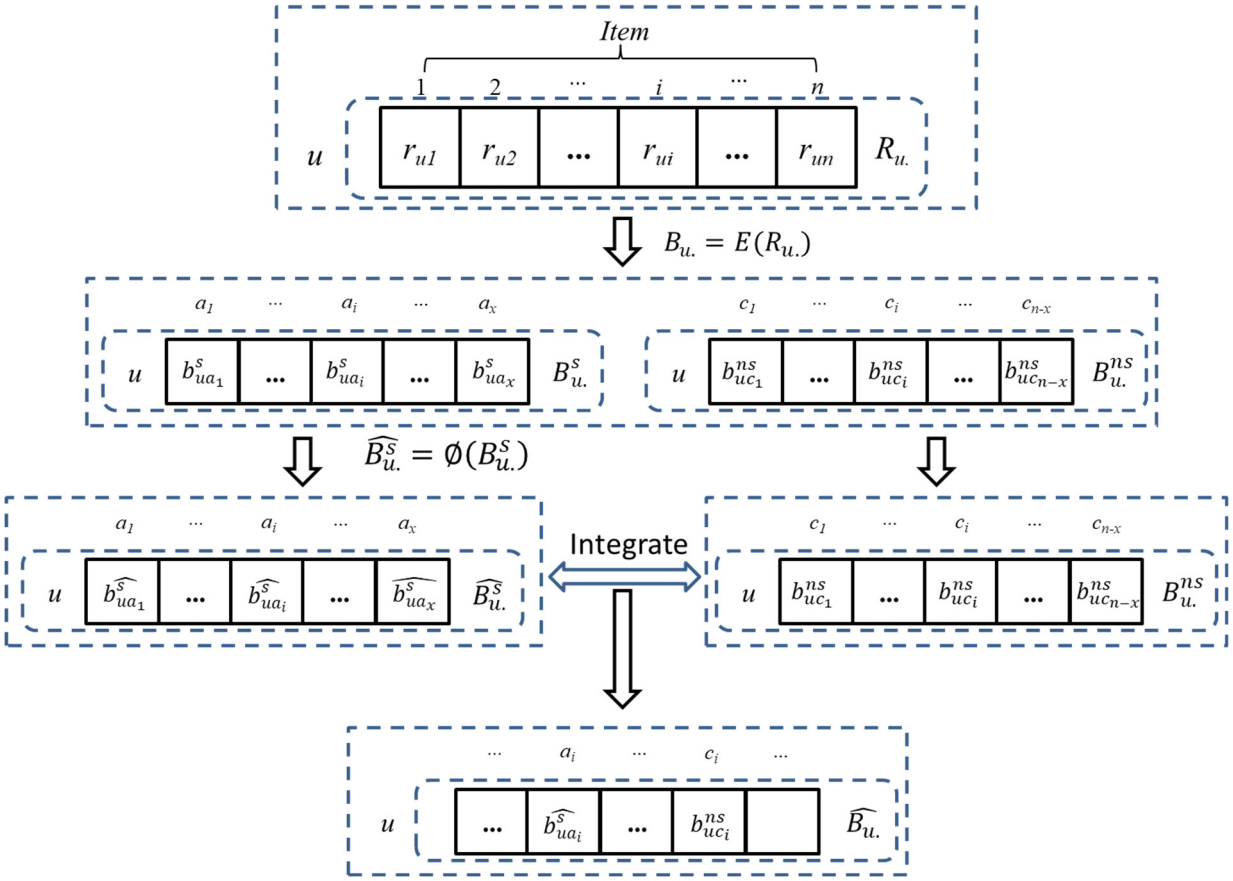


Figure 2. Privacy protection on rating data.

2.3. Item-Item Similarity

After the server receives the perturbed rating data $\widehat{B_{u.}}(u = 1, 2, \ldots, m)$ from each user, the perturbation matrix $\widehat{B}$ of ratings can be obtained as shown in Figure 3. $\widehat{B_{.\alpha}}$ and $\widehat{B_{.\beta}}$ represent the perturbation rating vectors for item α and item β, respectively. According to the type of common ratings stemming from item α and item β, divide the common ratings into the collection $O^s_{\alpha\beta}$ including all the sensitive rating pairs and the collection $O^{ns}_{\alpha\beta}$, including all the weakly sensitive rating pairs, where

$$O^s_{\alpha\beta} = \left\{ \left(\widehat{b^s_{.\alpha}}, \widehat{b^s_{.\beta}} \right) \right\}$$

$$O^{ns}_{\alpha\beta} = \left\{ \left(\widehat{b^s_{.\alpha}}, \widehat{b^{ns}_{.\beta}} \right), \left(\widehat{b^{ns}_{.\alpha}}, \widehat{b^s_{.\beta}} \right), \left(\widehat{b^{ns}_{.\alpha}}, \widehat{b^{ns}_{.\beta}} \right) \right\}$$

Furthermore, $IU^s(\alpha, \beta)$ and $IU^{ns}(\alpha, \beta)$ represent the collections of users corresponding to $O^s_{\alpha\beta}$ and $O^{ns}_{\alpha\beta}$, respectively, then calculate the similarity of each rating pair, then obtain the item-item similarity between α and β with the weighted sum, i.e.,

$$sim(\alpha, \beta) = \lambda \times sim_1 + (1 - \lambda) \times sim_2 \tag{4}$$

where sim_1 represents the similarity of the sensitive rating pair; sim_2 represents the similarity of the weakly sensitive rating pair; the parameter λ denotes the weight coefficient.

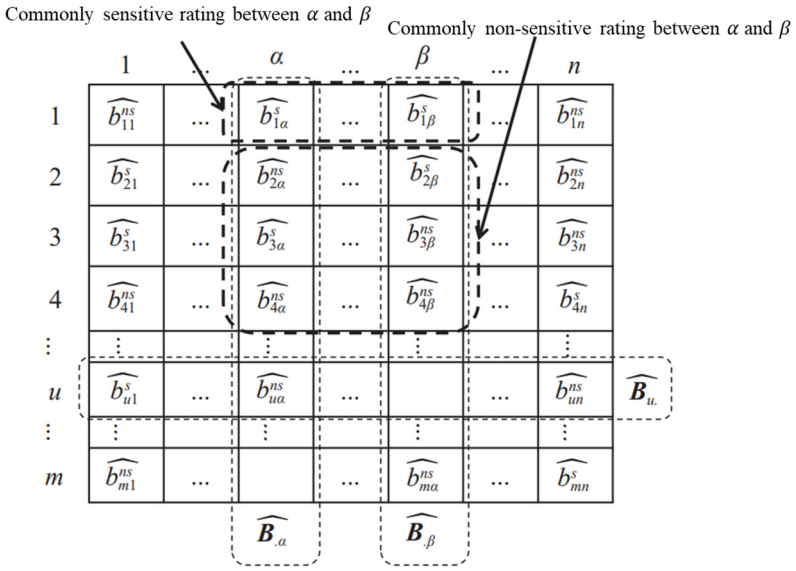


Figure 3. Perturbation matrix $\widehat{B}$.

2.3.1. Similarity of the Sensitive Rating Pair

To improve the performance of the recommendation system, we use the Bayesian algorithm to reconstruct the joint distribution of the sensitive rating pair within $\widehat{B}$. For $\widehat{B^s_{.\alpha}}$ and $\widehat{B^s_{.\beta}}$, W and $\widehat{W}$ represent the collection of pre-perturbation sensitive rating pairs and the collection of post-perturbation sensitive rating pairs.

$$W = \left\{ \left(b^{s-}_{.\alpha}, b^{s-}_{.\beta}\right), \left(b^{s-}_{.\alpha}, b^{s+}_{.\beta}\right), \left(b^{s+}_{.\alpha}, b^{s-}_{.\beta}\right), \left(b^{s+}_{.\alpha}, b^{s+}_{.\beta}\right) \right\}$$

$$\widehat{W} = \left\{ \left(\widehat{b^{s-}_{.\alpha}}, \widehat{b^{s-}_{.\beta}}\right), \left(\widehat{b^{s-}_{.\alpha}}, \widehat{b^{s+}_{.\beta}}\right), \left(\widehat{b^{s+}_{.\alpha}}, \widehat{b^{s-}_{.\beta}}\right), \left(\widehat{b^{s+}_{.\alpha}}, \widehat{b^{s+}_{.\beta}}\right) \right\}$$

The joint distribution estimation algorithm based on Bayesian algorithm is shown as follows.

First, initialize the rating pairs in W to be uniformly distributed, then calculate the prior probability through Bayesian method, i.e.,

$$P\left(\widehat{w} \in \widehat{W} \middle| w \in W\right) = \prod_{i=1}^{len(w)} p^{w[i] \oplus \widehat{w[i]}} \times q^{1-w[i] \oplus \widehat{w[i]}} \tag{5}$$

where p and q represent the random flip probability, respectively; len(w) denotes the number of elements in the collection of w; $w[i]$, and $\widehat{w[i]}$ represent the i-th element in the collections of w and $\widehat{w}$, respectively.

Furthermore, iteratively update the value of the posterior probability estimation for W, i.e.,

$$P_{iter}\left(w \middle| \left(\widehat{b^s_{u\alpha}}, \widehat{b^s_{u\beta}}\right) \in \widehat{W}\right) = \frac{P_{iter}(w) \times P\left(\widehat{b^s_{u\alpha}}, \widehat{b^s_{u\beta}} \middle| w\right)}{\sum_{t \in W} P_{iter}(t) \times P\left(\widehat{b^s_{u\alpha}}, \widehat{b^s_{u\beta}} \middle| t\right)} \tag{6}$$

$$P_{iter+1}(w \in W) = \frac{\sum_{u \in IU^s(\alpha,\beta)} P_{iter}\left(w \middle| \left(\widehat{b^s_{u\alpha}}, \widehat{b^s_{u\beta}}\right) \in \widehat{W}\right)}{N_{IU^s(\alpha,\beta)}} \tag{7}$$

where *iter* represents the times of iterations. In addition, the algorithm stops iterating under the following condition, i.e.,

$$\max_W P_{iter}(w \in W) - \max_W P_{iter-1}(w \in W) \leq \delta \tag{8}$$

where δ is used to control the number of iterations. Finally, Algorithm 1 outputs the Bayesian estimation distribution $P(w \in W)$. According to $P(w \in W)$, the similarity of the sensitive rating pair can be obtained as follows, i.e.,

$$sim_1 = P\left(w = \left(\widehat{b^{s-}_{.\alpha}}, \widehat{b^{s-}_{.\alpha}}\right)\right) + P = \left(\left(\widehat{b^{s+}_{.\alpha}}, \widehat{b^{s+}_{.\alpha}}\right)\right) \tag{9}$$

Algorithm 1: Joint distribution estimation algorithm based on Bayesian method

Input: p,q; $\widehat{B_{.\alpha}}$, $\widehat{B_{.\beta}}$;δ; W,$\widehat{W}$; $IU^s(\alpha, \beta)$
Output: $P(w \in W)$
1. Initialize $P_0(w \in W) = 0.25$;
2. for w in W
3. for $\widehat{w}$ in $\widehat{W}$
4. Calculate the prior probability $P(\widehat{w}|w)$ according to Formula 5;
5. end for
6. end for
7. Initialize $iter = 0$;//The number of iteration executions
8. while $(\max_{W} P_{iter}(w \in W) - \max_{W} P_{iter-1}(w \in W) > \delta)$
9. for u in $IU^s(\alpha, \beta)$
10. for w in W
11. Calculate the posteriori probability $P_{iter}\left(w \middle| \left(\widehat{b^s_{u\alpha}}, \widehat{b^s_{u\beta}}\right) \in \widehat{W}\right)$ according to Formula 6;
12. end for
13. end for
14. Calculate $P_{iter+1}(w \in W)$ according to Formula 7;
15. Update $iter = iter + 1$;
16. return $P(w \in W) = P_{iter}(w \in W)$;//Return after the while loop ends

2.3.2. Similarity of the Weakly Sensitive Rating Pair

The weakly sensitive rating pair remains the same in similarity as the original rating pair, and the detailed theoretical analysis can be seen in Section 3.1. The similarity of the weakly sensitive rating pair can be calculated by:

$$sim_2 = \frac{1}{N_{IU^{ns}(\alpha,\beta)}} \times \sum_{u \in IU^{ns}(\alpha,\beta)} \frac{2 - \left|\widehat{b_{u\alpha}} - \widehat{b_{u\beta}}\right|}{2} \tag{10}$$

where $\left(\widehat{b_{u\alpha}}, \widehat{b_{u\beta}}\right) \in O^{ns}_{\alpha\beta}$.

2.4. Local Top-N Recommendation

Users can obtain the collection $N(\alpha)$ of item α's neighbors according to the similarity results in the server. The local prediction formula for the user u's rating of the item α is shown below.

$$r(u, a) = \frac{\sum_{v \in N(\alpha)} sim(a, v) \times r_{uv}}{\sum_{v \in N(\alpha)} sim(a, v)} \tag{11}$$

where r_{uv} denotes the original rating of item v from user, u, and v is an element of the collection of $N(\alpha)$. Finally, the top N items with the highest predicted ratings are recommended to users.

3. Algorithmic Analysis

3.1. Analysis on Efficiency

In this paper, we mainly use the following two methods to ensure the validity of the rating data, reducing the impact stemming from the privacy noise on the recommendation process.

First, controlling the impact of privacy protection noise on recommendation results through classifying rating data according to its sensitivity. Specifically, for the weakly sensitive rating pair, its similarity is the same as the similarity when the rating data is not perturbed. This is because when encoding the rating data, the weakly sensitive rating is encoded as 0 and the sensitive score is ± 1. The difference between the sensitive rating and the weakly sensitive rating, whether before or after the random flip operation, is always 1. Therefore, the sensitivity classification and the encoding strategy adopted in this paper can effectively reduce the influence of noise on the similarity calculation, thus substantially improving the accuracy of the recommendation system.

Second, for the sensitive scoring pair, we utilize the Bayesian algorithm to reconstruct its distribution. Given the distribution of the random flip probability and the distribution of the rating pairs having been randomly flipped, the distribution of the original rating pair can be estimated. The accuracy of the estimation result obtained by Algorithm 1 is affected by the initial value of P_0 and the iteration times of δ. In this paper, initialize the original distribution be the uniform distribution, and through adjusting the times of iterations to ensure the accuracy of the estimation results. Reconstructing the distribution of the sensitive rating pair can reduce the impact of noisy data on the similarity calculation results of the sensitive rating pair, thereby further improving the performance of the recommendation system.

3.2. Analysis on Security

The proposed algorithm uses the random flip mechanism to perturb the encoded privacy-sensitive ratings and accomplish differential privacy protection for the sensitive rating. The proof that the proposed algorithm can satisfy the differential privacy security is as follows.

Assume $B^s_{u.}$ and $B^{s'}_{u.}$ be two different collections of neighbor data for user, u, and there exists one different sensitive rating element between the two collections, i.e.,

$$B^s_{u.} = \left[b^s_{ua_1},\ b^s_{ua_2}, b^s_{ua_3}, \ldots, b^s_{ua_i}, \ldots, b^s_{ua_x}\right]$$

$$B^{s'}_{u.} = \left[b^s_{ua_1},\ b^s_{ua_2}, b^s_{ua_3}, \ldots, b^{s'}_{ua_i}, \ldots, b^s_{ua_x}\right]$$

$\varnothing$ represents the random flip operation. Assume $X_u = [x_{ua_1}, x_{ua_2}, x_{ua_3}, \ldots, x_{ua_x}]$ denote the random output of $\varnothing$. Then,

$$\frac{pr(\varnothing(B^s_{u.} = X_u))}{pr(\varnothing(B^s_{u.}{}' = X_u))} = \frac{\prod_{t=1}^{x} pr\left(b^s_{ua_t} \to x_{ua_t}\right)}{\prod_{t=1}^{x} pr\left(b^s_{ua_t}{}' \to x_{ua_t}\right)} = \frac{pr\left(b^s_{ua_t} \to x_{ua_t}\right)}{pr\left(b^s_{ua_t}{}' \to x_{ua_t}\right)} \le \frac{q}{p} = e^{\varepsilon}$$

Therefore, it can be proved that the proposed algorithm can satisfy the requirements of the ε-differential privacy.

4. Experimental Analysis

In this paper, the analysis of performance is carried out on the publicly available datasets MovieLens 1M and Yahoo Music, and the details of the two datasets are shown in Table 1. To better observe the experimental effect, MovieLens 1M and Yahoo Music are both divided into the training set and the test set in the ratio of 8:2.

Table 1. Basic information on datasets.

Dataset	Number of Users	Number of Items	Number of Ratings	Range of Ratings	Sparsity (%)
MovieLens 1M	6000	4000	1,000,000	{1,2,3,4,5}	95.83
Yahoo Music	8089	1000	270,121	{1,2,3,4,5}	96.66

Here, we use the mean absolute error (*MAE*) and the root mean square error (*RMSE*) as the evaluation indicators, and detailed definitions of *MAE* and *RMSE* are:

$$MAE = \frac{\sum_{u \in U, v \in V} |r_{uv} - r'_{uv}|}{n} \tag{12}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{u \in U, v \in V} (r_{uv} - r'_{uv})^2}{N}} \tag{13}$$

where r_{uv} and r'_{uv} represent the real rating and the predictive rating of user, u, on item, v, respectively. U and V represent the collection of users and the collection of items, respectively.

4.1. Algorithms for Comparison

To further illustrate the performance of the algorithm in this paper, the following mainstream algorithms are selected.

(1) IBCF-DS [1], which is a collaborative filtering algorithm based on item similarity. Similar to PPPCF, it also adopts the rating encoding method [12] to ensure the consistency of the comparison. However, IBCF-DS does not adopt any data privacy protection measures, so it can be used as a baseline for the performance comparison.

(2) PNCF [7], which is an item-based collaborative filtering algorithm. It brings the exponential mechanism and the Laplace mechanism to protect the privacy of data [13–16], and introduces the recommendation perception sensitivity and truncated similarity to improve the utility of the algorithm.

(3) DPLCF [10], which is a distributed recommendation algorithm. Similar to PPPCF, it uses the random flip mechanism to locally protect the privacy security of the implicit data on the user side, then uses the cardinality estimation mechanism [17,18] on the server side to reconstruct the joint cardinality between items and through reducing the Jekaard similarity error between projects to optimize the accuracy of recommendations.

(4) LDP item-base CF [11], which uses the UE encoding strategy to flip data with a certain probability to locally protect the privacy of data on the user side. On the server side, it uses the frequency estimation and the joint frequency estimation [19–22] to reconstruct data to reduce the error of privacy noise on similarity and ensure the accuracy of recommendations.

(5) Truncated PPPCE, which is constructed through removing the Bayesian estimation module from PPPCF, in order to illustrate the influence of the Bayesian estimation module on PPPCE.

4.2. Parameter Settings

In this paper, assign 0.05 to the iteration threshold δ of the Bayesian estimation, and assign 0.2 to the similarity adjustment parameter λ. However, assign 0.5 to the similarity adjustment parameter λ in Equation 2 and Equation 3, and refer to the literature [11], assign 0.5 to the parameter of conditional probability similarity in the LDP item-base CF algorithm. To observe the algorithm performance under different privacy budgets, the range of the privacy budget is assigned to [0.1, 1] with the step size of 0.1. To observe the effect of the number of neighbors on the algorithm, the number N of neighbors of items is assigned to [20, 100] with the step size of 20.

4.3. Parameter Settings

4.3.1. Effect of *N* on Experimental Results

Assign 1 to the privacy budget ε, and Table 2 shows the performance of each algorithm with different numbers of neighbors, i.e., N.

Table 2. Comparison of the algorithm performance with different values of *N*.

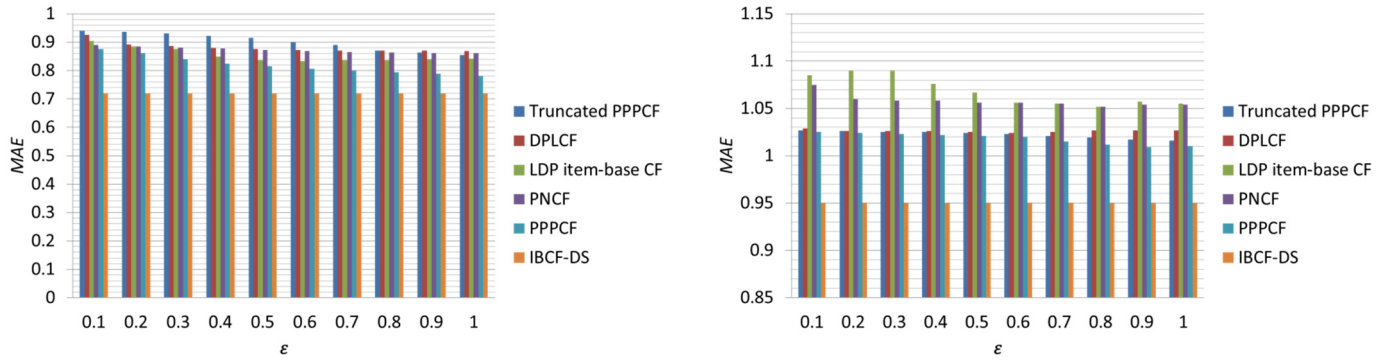
Dataset	Indicator	Algorithm	*N* = 20	*N* = 40	*N* = 60	*N* = 80	*N* = 100
MovieLens 1M	MAE	IBCF-DS	0.7219	0.7176	0.7169	0.7170	0.7171
		DPLCF	0.8912	0.8722	0.8718	0.8726	0.8694
		LDP item-base CF	0.8666	0.8484	0.8407	0.8361	0.8330
		PNCF	0.9600	0.9290	0.9110	0.8960	0.8870
		Truncated PPPCE	0.8582	0.8527	0.8537	0.8502	0.8498
		PPPCE	0.7911	0.7830	0.7783	0.7781	0.7798
	RMSE	IBCF-DS	0.9288	0.9220	0.9208	0.9207	0.9208
		DPLCF	1.1284	1.1050	1.1049	1.1050	0.1021
		LDP item-base CF	1.1217	1.0983	1.0888	1.0825	01.0784
		PNCF	1.2470	1.2040	1.1790	1.160	1.1480
		Truncated PPPCE	1.0944	1.0868	1.0883	1.0840	1.0837
		PPPCE	1.0071	0.9974	0.9220	0.9913	0.9932
Yahoo Music	MAE	IBCF-DS	0.9482	0.9484	0.9484	0.9485	0.9486
		DPLCF	1.0218	1.0243	1.0228	1.0235	1.0213
		LDP item-base CF	1.0461	1.0438	1.0427	1.0433	1.0414
		PNCF	1.0700	1.0620	1.0530	1.0389	1.0339
		Truncated PPPCE	1.0052	1.0038	1.0014	1.0037	1.0026
		PPPCE	0.9981	0.9980	0.9996	0.9989	0.9987
	RMSE	IBCF-DS	1.2464	1.2449	1.2447	1.2446	1.2445
		DPLCF	1.2842	1.2827	1.2813	1.2813	1.2790
		LDP item-base CF	1.3298	1.3257	1.3265	1.3260	1.3247
		PNCF	1.5120	1.4950	1.4790	1.4500	1.4290
		Truncated PPPCE	1.2830	1.2777	1.2756	1.2765	1.2751
		PPPCE	1.2727	1.2689	1.2713	1.2699	1.2685

Table 2 shows that with the increasing value of *N*, both *MAE* and *RMSE* of each algorithm continue to decrease, indicating that the increase in the number of neighbors will improve the accuracy of the recommendation system. PNCF performed poorly because it did not have the data reconstruction module. Among these differential privacy protection-based algorithms, PPPCE performed best in terms of *MAE* and *RMSE*, which were closest to the baseline algorithm IBCF-DS. Meanwhile, comparing with Truncated PPPCE, the average *MAE* of PPPCE is reduced by 8.2% (0.07) and the average *RMSE* of PPPCE is reduced by 8.3% (0.09) on the dataset of MovieLens 1M; the average *MAE* of PPPCE is reduced by 8.2% (0.07) and the average *RMSE* of PPPCE is reduced by 8.3% (0.09) on the dataset of Yahoo Music, indicating that the Bayesian estimation module has a good reconstruction effect on the perturbated data and can effectively improve the performance of the recommendation algorithm. Additionally, seen from Table 2, PPPCE outperforms DPLCF and LDP item-base CF on the two datasets, indicating that the Bayesian estimation method and the data sensitivity division method adopted in PPPCE perform better than the cardinality estimation method and frequency estimation method used in those two algorithms in terms of the data reconstruction.

4.3.2. Effect of ε on Experimental Results

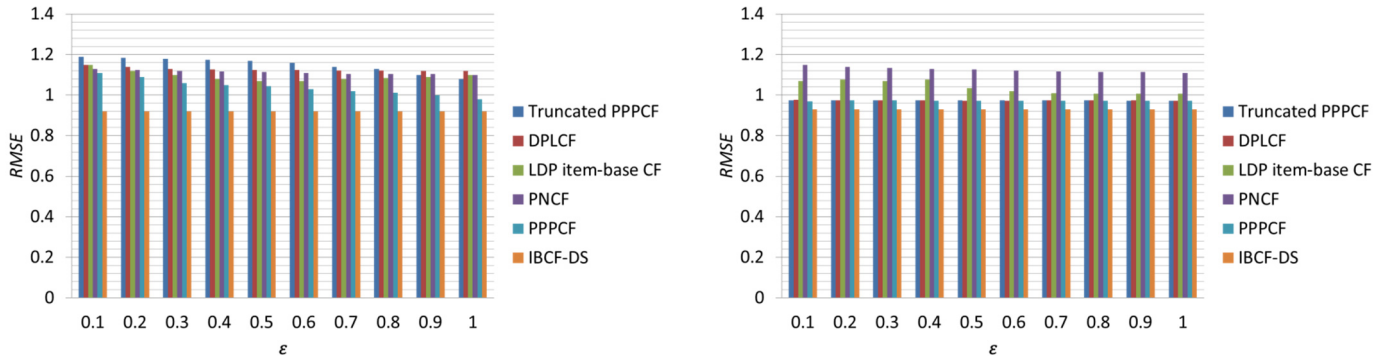
To ensure the effectiveness of privacy protection, we set the value of the privacy budget to be in the range of [0.1,1]. Figures 4 and 5, respectively show the changes of *MAE*

and *RMSE* on the dataset as the value of ε changes. Due to the lack of the differential privacy protection mechanism, the curve of IBCF-DS remains unchanged.



(**a**) (**b**)

Figure 4. *MAE* of each algorithm with different privacy budgets ε. (**a**) Movielens 1M; (**b**) Yahoo Music.



(**a**) (**b**)

Figure 5. *RMSE* of each algorithm with different privacy budgets ε. (**a**) Movielens 1M; (**b**) Yahoo Music.

With the increase in ε of the two datasets, both *MAE* and *RMSE* show a downward trend. This is because as ε increases, the noise of data decreases and the availability of data increases. Seen from the above figures that PPPCF outperforms other algorithms except IBCF-DS on both datasets. First, each algorithm performs in terms of *MAE* on the dataset of MovieLens 1M, PPPCF is 9.2% (0.083) less than the Truncated PPPCE, 9.7% (0.089) less than DPLCF, 3.3% (0.028) less than LDP item-base CF, and 5.1% (0.044) less than PNCF. Then, each algorithm performs in terms of *RMSE* on the dataset of MovieLens 1M, PPPCF is 9.3% (0.106) less than the Truncated PPPCE, 10.6% (0.123) less than DPLCF, 4.2% (0.046) less than LDP item-base CF, and 5.7% (0.063) less than PNCF. Second, due to the data on the dataset of Yahoo Music is sparser, causing the effect of the Bayesian estimation mechanism, the cardinality estimation mechanism, and the frequency estimation mechanism is not obvious, but compared with PNCF without the data reconstruction module, these three data reconstruction algorithms can still bring obvious improvement in utility. Among the three algorithms with data reconstruction, the prediction error of PPPCF still performs best, indicating that PPPCF can indeed improve the accuracy of the recommendation algorithm.

4.3.3. Algorithmic Efficiency

PPPCF mainly consists of three operations: local data processing, similarity calculation, and rating prediction. The local data processing operation can be further divided into two parts, one is the classification and encoding of ratings, the other is the differential privacy protection for ratings. As the local operation needs to traverse all the user-rating vectors in the dataset, its time complexity is $O(n)$.

The item-item similarity calculation includes the similarity calculation of the sensitive rating pairs and the similarity calculation for the weakly sensitive rating pairs. Assuming that the number of sensitive rating pairs is k, according to the execution process of BBBCF, when calculating the similarity of an item-item pair, it should traverse the weakly sensitive rating pairs once and the sensitive rating pairs iter times, so the time complexity is $O(((k-1)\cdot iter)+m)$, and the time complexity of the similarity calculation on the server side is $O\left(((k-1)\cdot iter+m)\cdot n^2\right)$. From Equation (11), the time complexity of the user's local prediction for all items is $O(n)$. Choose the DPLCF and LDP item-based CF algorithms, which also apply the localized differential privacy frameworks, for the comparative analysis. According to the calculation process of these two algorithms, the time complexity of their local data processing is also $O(n)$, and the time complexity of the similarity calculation is $O\left(m\cdot n^2\right)$. Additionally, since the rating prediction model used in the comparison algorithms is the same as PPPCF, so their time complexities of the rating prediction are $O(n)$. Compared with all the comparison algorithms, the time complexity of the local data processing section and the rating prediction section of PPPCF is the same as others, but in the similarity calculation section, the time complexity of PPPCF is higher due to the need to iteratively reconstruct the distribution of rating data. However, since $k \ll m$ and $iter \ll m$, it is still in the same range as all the comparison algorithms.

To further verify the results of the complexity analysis, taking the Movielens 1M and Yahoo Music datasets for example, assign ε and N be 1, 100, then evaluate the time overhead of each algorithm on the two datasets, experimental results are shown in Table 3. Each algorithm spends about the same amount of time on local data processing operations, with the PPCF algorithm having the least time overhead. This is because the PPPCF algorithm classifies the data sensitivity before perturbing the data, which greatly reduces the amount of data that needs to be perturbed.

Table 3. Comparison of the time overhead.

Dataset	Runtime/s	DPLCF	LDP Item-Base CF	Truncated PPPCE	PPPCF
Yahoo Music	Local Data Processing	2.36	2.55	1.78	1.83
	Similarity Calculation	32.79	41.25	22.71	382.54
	Rating Prediction	53.47	52.80	47.59	57.39
MovieLens 1M	Local Data Processing	8.22	8.62	7.79	7.71
	Similarity Calculation	652.23	833.37	507.95	5497.68
	Rating Prediction	464.94	488.14	467.07	475.33

In the process of the similarity calculation, PPPCF is affected by k (i.e., number of the sensitive rating pairs) and iter (i.e., times of iterations), so multiple iterations are required to ensure a lower prediction error, causing its time overhead for the similarity calculation is greater than the other comparison algorithms. Meanwhile, the benefit of high time overhead is that its prediction error is lower than others.

Additionally, PPPCF utilizes the Hamming distance to measure the similarity, and in the actual operation process, the Hamming distance between vectors can be obtained through XOR operation, which can improve the operation efficiency of PPPCF algorithm. This is confirmed by the fact that the Truncated PPPCE has a smaller time overhead in calculating similarity in Table 3. For the recommendation algorithm, the efficiency of the rating prediction calculation is particularly important, and the time taken to predict ratings for a single user on the Yahoo Music and Movielens 1M datasets is 6~7 ms and 78~81 ms, respectively, which can meet the actual needs.

In summary, although the PPPCF increases the time overhead caused by the similarity calculation on the server side, it does not affect the efficiency of the rating prediction, and the recommendation error of PPPCF is superior to other algorithms for comparison on datasets of different sparsity. This shows that under the interaction of the privacy

sensitivity classification module and the Bayesian data reconstruction module, the similarity calculation is less affected by the perturbed data to achieve higher data availability and recommendation accuracy.

5. Conclusions

Based on the random flip mechanism, this paper proposes a collaborative filtering algorithm with differential privacy-preserving protection. In order to trade off the relationship between privacy protection and performance of recommendation, it classifies ratings into privacy-sensitive and weakly sensitive privacy ratings according to user's personalized needs for privacy. Then, it brings a coding rule for ratings to effectively reduce unwanted noise resulting from privacy protection. Furthermore, it reconstructs the distribution of the perturbated data through the Bayesian joint estimation model. Experimental results on two public datasets verify that the proposed algorithm has better performance in trading off between the performance of recommendation system and the security of privacy, and it is of certain practical value. Due to the privacy protection reduces the recommendation efficiency of the system to a certain extent, in our future work, we will focus on the data pre-processing strategies, such as cleaning and classification of rating data to further reduce time consumption. Additionally, we will also consider doing some exploration in hybrid recommendation algorithms.

Author Contributions: Conceptualization and methodology, B.C. and W.L.; software, X.Z.; validation, B.C., X.Q. and P.C.; formal analysis, P.C.; investigation, X.Q.; writing—original draft preparation, B.C. and K.F.; writing—review and editing, B.C. and W.L.; funding acquisition, B.C. and X.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MILITARY PROJECT from PLA EQUIPMENT DEVELOPMENT DEPARTMENT, grant numbers are NO. JK2021A030450 and NO. JK20211A010443, and also funded by the Key-Area Research and Development Program of Guangzhou City from GUANGZHOU SCIENCE AND TECHNOLOGY DEPARTMENT, grant number is NO. 202206030009.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: In order to prevent the data from being abused, we open our data based on reasonable requests. Please contact authors with a formal application form to access the data from liuwei_bit@126.com.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wei, S.; Ye, N.; Zhang, S.; Huang, X.; Zhu, J. Item-Based Collaborative Filtering Recommendation Algorithm Combining Item Category with Interestingness Measure. In Proceedings of the 2012 International Conference on Computer Science and Service System, Nanjing, China, 11–13 August 2012; pp. 2038–2041.
2. Su, X.; Khoshgoftaar, T.M. A Survey of Collaborative Filtering Techniques. *Adv. Artif. Intell.* **2009**, *2009*, 421425. [CrossRef]
3. Kenthapadi, K.; Mironov, I.; Thakurta, A.G. Privacy-preserving Data Mining in Industry. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, 11–15 February 2019; pp. 840–841.
4. Zhang, F.; Xue, E.; Guo, R.; Qu, G.; Zhao, G.; Zomaya, A.Y. DS-ADMM++: A Novel Distributed Quantized ADMM to Speed up Differentially Private Matrix Factorization. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 1289–1302. [CrossRef]
5. Chen, C.; Wu, H.; Su, J.; Lyu, L.; Zheng, X.; Wang, L. Differential Private Knowledge Transfer for Privacy-Preserving Cross-Domain Recommendation. In Proceedings of the ACM Web Conference 2022, Virtual Event, Lyon, France, 25–29 April 2022; pp. 1455–1465.
6. Zhao, Y.; Chen, J. A Survey on Differential Privacy for Unstructured Data Content. *ACM Comput. Surv.* **2022**, *54*, 207. [CrossRef]
7. Zhu, X.; Sun, Y. Differential Privacy for Collaborative Filtering Recommender Algorithm. In Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics, New Orleans, LA, USA, 11 March 2016; pp. 9–16.
8. Ran, X.; Yin, E.; Wang, Y. Differential Privacy-Preserving Recommendation Algorithm Based on Bhattacharyya Coefficient Clustering. *J. Beijing Univ. Posts Telecommun.* **2021**, *44*, 81–88. [CrossRef]
9. Fenske, E.; Mani, A.; Johnson, A.; Sherr, M. Accountable Private Set Cardinality for Distributed Measurement. *ACM Trans. Priv. Secur.* **2022**, *25*, 25. [CrossRef]

10. Gao, C.; Huang, C.; Lin, D.; Jin, D.; Li, Y. DPLCF: Differentially Private Local Collaborative Filtering. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 961–970.
11. Guo, T.; Luo, J.; Dong, K.; Yang, M. Locally differentially private item-based collaborative filtering. *Inf. Sci.* **2019**, *502*, 229–246. [CrossRef]
12. Wang, T.; Zhang, X.; Feng, J.; Yang, X. A Comprehensive Survey on Local Differential Privacy toward Data Statistics and Analysis. *Sensors* **2020**, *20*, 7030. [CrossRef] [PubMed]
13. Wu, C.; Wu, F.; Lyu, L.; Huang, Y.; Xie, X. FedCTR: Federated Native Ad CTR Prediction with Cross-platform User Behavior Data. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 62. [CrossRef]
14. Knijnenburg, B.P.; Berkovsky, S. Privacy for Recommender Systems: Tutorial Abstract. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 394–395.
15. Chen, C.; Liu, Z.; Zhao, P.; Zhou, J.; Li, X. Privacy preserving point-of-interest recommendation using decentralized matrix factorization. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
16. Meng, X.; Wang, S.; Shu, K.; Li, J.; Chen, B.; Liu, H.; Zhang, Y. Personalized privacy-preserving social recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
17. Jalili, M.; Ahmadian, S.; Izadi, M.; Moradi, P.; Salehi, M. Evaluating collaborative filtering recommender algorithms: A survey. *IEEE Access* **2018**, *6*, 74003–74024. [CrossRef]
18. Alhijawi, B.; Kilani, Y. A collaborative filtering recommender system using genetic algorithm. *Inf. Process. Manag.* **2020**, *57*, 102310. [CrossRef]
19. Herlocker, J.L.; Konstan, J.A.; Riedl, J. Explaining collaborative filtering recommendations. In Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, Philadelphia, PA, USA, 2–6 December 2000; pp. 241–250.
20. Koren, Y.; Rendle, S.; Bell, R. Advances in collaborative filtering. In *Recommender Systems Handbook*; Springer: New York, NY, USA, 2021; pp. 91–142.
21. Miyahara, K.; Pazzani, M.J. Collaborative Filtering with the Simple Bayesian Classifier. In Proceedings of the PRICAI 2000 Topics in Artificial Intelligence, Melbourne, Australia, 28 August–1 September 2000; pp. 679–689.
22. Wang, W.; Duan, L.-Y.; Jiang, H.; Jing, P.; Song, X.; Nie, L. Market2Dish: Health-aware food recommendation. *ACM Trans. Multimed. Comput. Commun. Appl.* **2021**, *17*, 33. [CrossRef]