PAPER

Sparsity Reduction Technique Using Grouping Method for Matrix Factorization in Differentially Private Recommendation Systems

Taewhan KIM[†], **Kangsoo JUNG**[†], ***Nonmembers***, ***and*** **Seog PARK**[†a)], ***Member***

SUMMARY Web service users are overwhelmed by the amount of information presented to them and have difficulties in finding the information that they need. Therefore, a recommendation system that predicts users' taste is an essential factor for the success of businesses. However, recommendation systems require users' personal information and can thus lead to serious privacy violations. To solve this problem, many research has been conducted about protecting personal information in recommendation systems and implementing differential privacy, a privacy protection technique that inserts noise into the original data. However, previous studies did not examine the following factors in applying differential privacy to recommendation systems. First, they did not consider the sparsity of user rating information. The total number of items is much more than the number of user-rated items. Therefore, a rating matrix created for users and items will be very sparse. This characteristic renders the identification of user patterns in rating matrixes difficult. Therefore, the sparsity issue should be considered in the application of differential privacy to recommendation systems. Second, previous studies focused on protecting user rating information but did not aim to protect the lists of user-rated items. Recommendation systems should protect these item lists because they also disclose user preferences. In this study, we propose a differentially private recommendation scheme that bases on a grouping method to solve the sparsity issue and to protect user-rated item lists and user rating information. The proposed technique shows better performance and privacy protection on actual movie rating data in comparison with an existing technique.

key words: *privacy, recommendation system, differential privacy, sparse matrix, grouping*

1. Introduction

With the rise of the Internet, the amount of information found in web services has been increasing day by day. Consequently, web service users are overwhelmed by the amount of information they receive and have trouble finding the information they need. Thus, a recommendation system that suggests appropriate options to users is important for the success of businesses. However, recommendation systems need users' personal information, which may have sensitive details. Careless use of this information by recommendation systems may lead to serious privacy violations. To solve this problem, existing research has explored ways to protect personal information in recommendation systems. In addition, differential privacy, which inserts noise into the original data, has been used as a de facto standard for privacy protection.

However, existing studies that applied differential privacy to the recommendation system did not consider the following two factors. First, they did not inspect the sparsity of user rating information. Matrix sparsity is a natural phenomenon caused by an increase in service scale; it renders the identification of user patterns in recommendation systems difficult. Therefore, several studies have attempted to solve this problem [1], [2]. However, the sparsity problem in differential privacy should be considered with additional noise insertion caused by matrix sparsity. The existing technique does not consider this additional noise and thus cannot be applied to a differentially private recommendation. In this study, we examine this sparsity issue as we apply differential privacy to recommendation systems. Second, existing studies treated only user rating information as a personal information because it reveals users' taste for certain items. However, the lists of items rated by users are also sensitive information. For example, [3] demonstrated that a user can be identified through a list of movies they have rated. Therefore, user rating and user-rated item list information should both be protected.

In this research, we propose a differentially private recommendation scheme that bases on the grouping method to solve the sparsity issue in recommendation systems. We generate a submatrix by user grouping and perform matrix factorization using this submatrix. Although the information is lost in the grouping process, this loss can be offset by the reduction of noise. We validate the proposed technique to improve utility performance through experiments. In addition, we propose the differentially private recommendation method to protect user ratings and rated item list information in the matrix factorization system.

The composition of this paper is as follows. In Sect. 2, we introduce existing studies and analyze the features and limitations of each work. In Sect. 3, we explain the proposed grouping method-based differentially private recommendation scheme. In Sect. 4, we analyze the experimental results of our proposed method and validate that the proposed method gives meaningful results in terms of recommendation performance. Finally, we discuss the conclusion of this study and recommendations for future research in Sect. 5.

2. Related Works

In this section, we introduce basic concepts for an understanding of the proposed method and summarize the related studies. We briefly describe the basic concepts of recommendation systems and differential privacy. Then, we

Manuscript received September 5, 2019.
Manuscript revised February 19, 2020.
Manuscript publicized April 1, 2020.

[†]The authors are with Department of Computer Science and Engineering, Sogang University, Mapo-gu, Seoul 04107, Korea.
a) E-mail: spark@sogang.ac.kr

DOI: 10.1587/transinf.2019EDP7238

analyze the related studies and describe current research gaps.

2.1 Recommendation System

A recommendation system is a system that suggests a product or information by predicting user preference. Recommendation systems are used in various fields, such as movies, music, news, and books. Two typical algorithms can implement recommendation systems: content-based recommendation and collaborative filtering. Content-based recommendation returns similar items by analyzing user information. For example, a movie classified under the action genre is suggested to a user who has watched an action movie. Collaborative filtering uses a similar evaluation of a user's history. For example, collaborative filtering recommends a movie viewed by other users who have similar movie preferences. Collaborative filtering is classified into two categories: neighborhood methods and latent factor methods. The neighborhood method uses the correlation between items and users, whereas the latent factor method derives the latent vector that represents the item and user relationship. In this work, we use matrix factorization [4] as a representative latent factor method, which shows outstanding performance in recommender systems.

2.1.1 Matrix Factorization

Recommendation systems use items and user information. Let r_{ij} be the rating of a user u_i for item v_j. For the set of the user and the items, the rating matrix R is defined as follows.

Definition 1. Rating matrix
We assume the user set $U = \{u_1, \ldots, u_n\}$ and the item set $V = \{v_1, \ldots, v_m\}$. Let r_{ij} be the rating of a user u_i for item v_j. The rating matrix R is as follows:

$$R = [r_{ij}]_{n \times m}$$

Matrix factorization decomposes rating matrix R to a user latent vector and an item latent vector, which are used to create the prediction matrix $\hat{R} = [\hat{r}_{ij}]_{n \times m}$. The user's latent vector and the item's latent vector are defined as follows.

Definition 2. User latent vector and item latent vector
We assume a prediction matrix $\hat{R}$, a dimension of latent vector d, a set of users $U = \{u_1, \ldots, u_n\}$, a set of items $V = \{v_1, \ldots, v_m\}$, a user latent vector $p_i \in \mathbb{R}^d$ for user u_i, $(1 \leq i \leq n)$, and an item latent vector $q_j \in \mathbb{R}^d$ for item v_j, $(1 \leq j \leq m)$. The prediction matrix $\hat{R}$ is calculated as follows:

$$\forall u_i \in U,\ \forall v_j \in V,\ \hat{r}_{ij} = p_i^T \cdot q_j.$$

A user latent matrix P and an item latent matrix Q are as follows:

$$P = [p_i]_{1 \leq i \leq n}, \qquad Q = [q_j]_{1 \leq j \leq m}.$$

The goal of matrix factorization is to decompose the rating matrix R into user latent vectors and item latent vectors such that they are maximally similar to R. That is, the user latent vector and the item latent vector are to minimize the following loss function L.

$$L = \frac{1}{\mathcal{M}} \cdot \sum_{(i,j) \in \mathcal{M}} \left(r_{ij} - p_i^T q_j\right)$$

$\mathcal{M}$ is a set of ratings that are given by users to items. The optimization problem that minimizes the loss function can be expressed as follows:

$$(\mathrm{U}, \mathrm{V}) = \min_{\mathrm{U},\mathrm{V}} \sum\nolimits_{r_{ij} \in \mathcal{M}} \left[\left(r_{ij} - p_i^T q_j\right)^2 + \lambda\left(\|p_i\|^2 + \|q_j\|^2\right)\right].$$

The above optimization problem can be solved by the stochastic gradient descent method.

Figure 1 shows an example of matrix factorization. We suppose the presence of five users $u_1, \ldots, u_5$ and five items $v_1, \ldots, v_5$, and we do not know r_{32} and r_{55}, which are the ratings of users u_3, u_5 to items v_2, v_5, respectively (Fig. 1 (a)). We use SGD [4] to decompose the rating matrix into the user latent matrix and item latent matrix to minimize the loss of function L. Then, we predict empty ratings (Fig. 1 (c)).

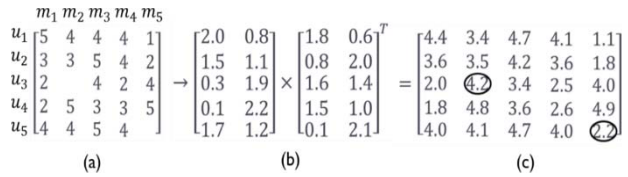

Fig. 1 Example of matrix factorization

2.2 Differential Privacy

Differential privacy [5] is a mathematical model that prevents information exposure, thus ensuring privacy protection at a specified level ε, which is customized by users. Given two neighboring databases D_1 and D_2, which differ by only one record, a randomized function K provides ε-differential privacy if all datasets with D_1 and D_2 differ by only one element and all $\mathrm{S} \subseteq \mathrm{Range(K)}$, that is,

$$\frac{\mathrm{Prob}(K(D_1) = S)}{\mathrm{Prob}(\mathrm{K}(D_2) = S)} \leq \mathrm{e}^{\varepsilon},\ S \in \mathrm{Range}(K),\ \varepsilon > 0.$$

This description of differential privacy means that specific individuals in the statistical database cannot be deduced correctly by keeping the possibility of a change in query results by inserting/deleting one datum to be less than e^{ε}. The larger the ε, the greater the probability that the two results are different. Conversely, the smaller the ε, the greater the likelihood that the two results are similar. In addition, differential privacy has the following properties.

Sequential composition: If $\mathcal{M}_i$ is an algorithm that satisfies ϵ_i-differential privacy, then for any database D, an algorithm $\mathcal{M}_{[k]}(D)$ (which carries out all $(\mathcal{M}_1(D), \ldots, \mathcal{M}_k(D))$) satisfies $\left(\sum_{i=1}^{k} \epsilon_i\right)$-differential privacy.

Parallel composition: If $\mathcal{M}_i$ is an algorithm that satisfies ϵ_i-differential privacy and D_i is an arbitrary subset of database D, then $\mathcal{M}_{[k]}(D_{[k]})$ (which carries out all $(\mathcal{M}_1(D_1), \ldots, \mathcal{M}_k(D_k))$) satisfies $\left(\max_{i=1,\ldots,k} \epsilon\right)$-differential privacy.

Sequential and parallel compositions have been proven in [6]. These two properties can be used to represent the degree of privacy protection of complex algorithms.

A typical mechanism for applying differential privacy is the Laplace mechanism; the Laplace distribution is the simplest mechanism for applying differential privacy, and the Laplace mechanism satisfies ϵ-differential privacy [6]. A ε-differentially private Laplace noise mechanism is defined as L(D) = f(D) + X, where X is a random variable drawn from the Laplace distribution with mean = 0 and standard deviation = $\sqrt{2}\,\Delta f/\varepsilon$. Δf is the sensitivity of the function, which means that the maximum value of the change in the query results from the insertion/deletion of a specific individual. The Laplace distribution is as follows:

$$\mathrm{f}(\mathrm{x}\,|\,\mu, \mathrm{b}) = \frac{1}{2\mathrm{b}} \exp\left(-\frac{|\mathrm{x}-\mu|}{\mathrm{b}}\right), \ \mathrm{b} > 0.$$

2.3 Privacy Protection in Recommendation Systems

In 2006, Netflix, a leading movie streaming service, held a competition called "Netflix Prize" to improve its recommendation performance. For the competition, Netflix provided 100 million rating details, 480,000 users, and more than 17,000 films to contest participants. The company implemented anonymization methods, such as the removal of identifiers, to protect its users' privacy. However, the combination of the provided details and IMDb data allowed the identification of Netflix users [3]. As a result, various studies have emerged to seek ways to protect user privacy while providing appropriate recommendation performance. Two main types of studies deal with privacy protection in existing recommender systems. The first type involves recommendation systems in distributed environments [7], [8]. Personal information is distributed in different places. Thus, an attacker will not easily find a user's personal information by combining multiple repositories with information.

Second, there is a way to encrypt [9] or mix noise to user data [10]–[13]. Encrypting personal data can help protect privacy, but the time complexity for encryption/decryption is too high. Therefore, most studies focus on inserting noise to satisfy differential privacy.

McSherry's study [10] was the first study to apply differential privacy to a recommender system. The neighborhood method, a collaborative filtering technique that makes recommendations by analyzing the correlation between items, was used. The noise was injected into the item's average ratings based on a Laplace mechanism to satisfy differential privacy. However, the weak point of the neighborhood method is that it performs poorly compared with the latent factor method. Berlioz's study [11] applied differential privacy to matrix factorization and tried to determine the best step for adding noise in matrix factorization for recommendation performance. The first step was to add noise into an original rating matrix before the matrix factorization. The second step was to add noise during matrix factorization. The final step was to add noise into latent matrixes resulting from matrix factorization. Among the three methods, the addition of noise into the rating matrix before matrix factorization performed the best. Existing studies have limitations because they did not consider the following two factors.

i) The sparsity of rating matrix: As the size of web services grows, the number of items increases more rapidly than the number of user ratings. Consequently, the sparsity of rating matrixes also increases, which causes difficulty in finding patterns in matrix factorization and degrades recommendation performance. Thus, many studies have attempted to solve the matrix sparsity problem in recommendation systems. However, when we apply differential privacy into matrix factorization, too much noise is inserted into the sparse matrix because noise should be inserted for items that are not rated by users. Thus, when the matrix sparsity problem is considered in a differentially private recommendation system, the noise insertion problem must be considered as well. As far as we know, the proposed technique is the first to solve this sparsity problem in differentially private matrix factorization.

ii) Privacy of list of user-rated items: Existing studies focus only on user rating information to protect the privacy and adds noise to keep the information safe. However, user-rated item lists are also sensitive information. For example, a user's pattern of regularly buying a feminine product can be used to infer that the user is a woman, regardless of whether they think positively or negatively about that item. Thus, we should protect lists of user-rated items.

In this study, the proposed differentially private recommendation algorithm considers the two abovementioned factors. The proposed technique exhibits better recommendation performance and ensures more thorough privacy protection than the existing technique.

3. Differentially Private Recommendation Scheme Using Grouping

In this chapter, we explain how the proposed grouping technique in matrix factorization enhances recommendation performance. We attempt to solve the sparsity issue by grouping techniques and use Laplace mechanisms to protect the user rating information and the user-rated item list.

3.1 Overview

Figure 2 shows the overall structure of the proposed technique. We assume a user u with n users $u_1, \ldots, u_n$ and m items $i_1, \ldots, i_m$. First, we generate each user i's subgroup based on the users' similarities and create a user i's submatrix $R_{i,u}$ (noisy grouping). During this process, we add noise into the similarity calculation to protect the user-rated item list. Then, we insert the noise into the user rating

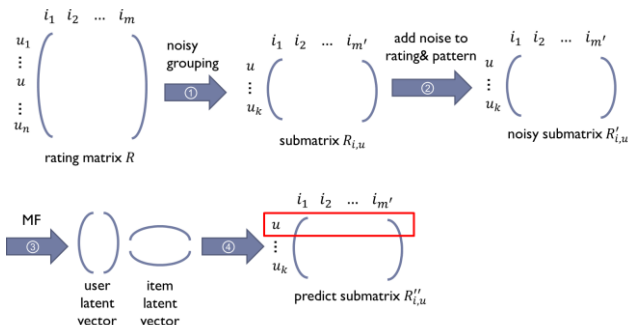

Fig. 2 Overall structure of the proposed recommendation system

information to satisfy differential privacy (adding noise to rating and pattern). The noisy submatrix $R'_{i,u}$ is decomposed into the user's latent vectors and the item's latent vectors. Finally, we generate submatrix $R''_{i,u}$ to predict user i's rating information for a recommendation.

3.2 Grouping Method

3.2.1 Grouping Score

The ideal submatrix for a recommendation should be a good representation of a user's pattern and must have high density. Therefore, the grouping score function should consider the following two factors.

The first criterion is to evaluate the density of the user's rating. The more ratings a user makes, the more dense the submatrix. Thus, building a group with a user who makes numerous ratings is advantageous. We define a rating frequency score function below.

Definition 3. Rating frequency score function
The rating frequency score function with user i freq(i) is defined as follows:

$$\text{freq}(i) = \frac{\text{the number of items which user } i \text{ rated}}{\text{the number of whole items}}.$$

The rating frequency score function represents how many times the user has given ratings.

The second criterion is to estimate how similarly other users have rated to user i. If user i is grouped with other users who have a similar rating pattern, the recommendation performance is enhanced. We use the Jaccard similarity [14] to evaluate the user's rating similarity.

Definition 4. Similarity score function
The similarity score functions for the two users i and j is defined as follows:

$$\text{sim}(s_i, s_j) = \text{jaccard}(s_i, s_j) = \frac{|s_i \cap s_j|}{|s_i \cup s_j|}.$$

The similarity score function represents the user's rating similarity as a value between 0 and 1. These proposed criteria can be considered in several ways. Through experiments,

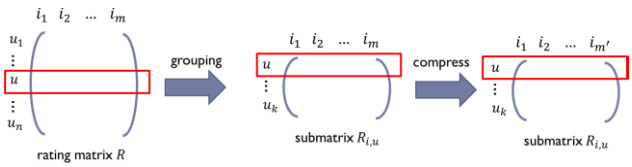

Fig. 3 Example of the grouping method

we find that weighing two factors together to create a submatrix is the most effective method. Therefore, we define a new grouping score function that allows two criteria to be considered at once.

Definition 5. Grouping score function
With the size of group k, user i, and for all user j, $(1 \leq j \leq n)$, the grouping score function is as follows:

$$\text{scoree}_i(j) = \begin{cases} \frac{k}{n} \cdot freq(j) + \left(1 - \frac{k}{n}\right) \cdot sim(s_i, s_j), & i \neq j \\ 1, & i = j \end{cases}.$$

The grouping score function is expressed as the weighted sum of the rating frequency score function and the similarity score function. The importance of the two functions depends on group size. For example, if the group size is small, then the density can easily be raised, whereas the rating patterns are not maintained well. Therefore, the similarity of the rating pattern becomes increasingly important. Conversely, if the group size is large, then the frequency of the user's ratings becomes highly important because rating patterns can be maintained well, whereas raising the density can be difficult.

3.2.2 Grouping Process

We explain the grouping method without noise insertion in this section. We modify this grouping method into a differentially private grouping method in Sect. 3.3

Given rating matrix R, size of group k, and user u, we can obtain a submatrix $R_{i,u}$ through grouping. The process of grouping is as follows. First, each user obtains a grouping score for other users based on themselves. Then, k users are grouped in descending order of the grouping score, and submatrix $R_{i,u}$ is generated. We add noise into the grouping score calculation process and explain the results in Sect. 3.3. Through this grouping process, items that have not been rated by k users are removed. Thus, the number of users decreases from n to k, and the number of items decreases from m to m' (Fig. 3).

Figure 4 shows the algorithm for the grouping techniques. The weight constants of the grouping score are determined in line 1. Then, we obtain grouping scores for all users from line 2 to line 8. From line 9 to line 12, users with k number of high grouping scores are grouped into R_u in order of scores. In line 13, items that have not been rated by k users in the R_u are removed.

Algorithm 1 Grouping method

Input :
$R \in \mathbb{R}^{n \times m}$ - rating matrix,
u - target user,
k - group size
Output :
$R_u \in \mathbb{R}^{k \times m'}$ - submatrix for user u

$\alpha = k/n$
for $i \leftarrow 1$ to n **do**
  if $i \neq u$ **then**
    $similar_i = \mathrm{sim}(s_u, s_i)$
    $frequent_i = \mathrm{freq}(i)$
    $score_i = \alpha \cdot frequent_i + (1 - \alpha) \cdot similar_i$
  else
    $score_i = 1.0$
$R_u = []$
$index \leftarrow$ indices in descent order of $score$
for each $idx \in index$ **do**
  add $R[idx]$ to R_u
$R_u \in \mathbb{R}^{k \times m'} \leftarrow$ compress $R_u \in \mathbb{R}^{k \times m}$
return R_u

Fig. 4 Grouping method

3.3 Differentially Private Grouping Method

In this chapter, we explain how to apply the Laplace mechanism into the grouping method for satisfying differential privacy.

3.3.1 Privacy Preservation of User Rating Information

The user rating information reveals a user's preference and thus contains sensitive personal information. Thus, we add noise based on the Laplace mechanism into the user rating information before matrix factorization. When we assume the minimum value of the rating matrix is $r_{\min}$ and the maximum value is $r_{\max}$, matrix factorization sensitivity is as follows:

$$\forall \mathrm{i}, \mathrm{j} \in [1, \ldots, \mathrm{n}],\ \Delta r_{ij} = r_{\max} - r_{\min}.$$

We can extract the noise from the Laplace distribution with the variance $\frac{\Delta r_{ij}}{\epsilon}$ and insert it to all ratings r_{ij} to ensure ϵ-differential privacy. Figure 5 illustrates this algorithm.

3.3.2 Privacy Preservation of User-Rated Item List

The list of the user's rated items should also be protected. Therefore, we apply the randomized algorithm [12] to the user's rating pattern (Fig. 6).

Theorem 1
Randomized algorithm [15] ensures ϵ-differential privacy.
Proof
With $p = \frac{2}{1+e^{\epsilon}}$, if s_{ij} is 1, then the probability that $\hat{s}_{ij}$ will be 1 is $1 - p + \frac{p}{2} = 1 - \frac{p}{2}$. Conversely, when s_{ij} is 0, then the probability that $\hat{s}_{ij}$ will be 1 is $\frac{p}{2}$. Randomized algorithm

Algorithm 2 Differential Private Input Perturbation

Input :
$R = \{r_{ij}\}$ - rating matrix,
Δr - rating range,
ϵ - privacy parameter
Output :
noisy rating matrix R'

Let $R' = \{r_{ij} + Laplace(\frac{\Delta r}{\epsilon}) | r_{ij} \in R\}$
Clamp the ratings in R' to the range $[r_{min}, r_{max}]$
return R'

Fig. 5 Differential private input perturbation

Algorithm 3 Differential private Randomized algorithm

Input :
$s_i = \langle s_{ij} \rangle_{1 \leq j \leq m}$ - rating pattern of user i,
ϵ - privacy parameter
Output :
$\hat{s_i} = \langle \hat{s_{ij}} \rangle_{1 \leq j \leq m}$

$p = 2/(1 + e^{\epsilon})$
for $j \leftarrow 1$ to m **do**

$$\hat{s_{ij}} = \begin{cases} 0, & \text{with probability } p/2, \\ 1, & \text{with probability } p/2, \\ s_{ij}, & \text{with probability } 1 - p \end{cases}$$

return $\hat{s_i}$

Fig. 6 Differential private randomized algorithm

f, which receives s_{ij} and returns $\hat{s}_{ij}$, satisfies the following formula:

$$\frac{\Pr[f(1) = 1]}{\Pr[f(0) = 1]} = \frac{1 - \frac{p}{2}}{\frac{p}{2}} = \frac{2 - p}{p} = \frac{2 - \frac{2}{1+e^{\epsilon}}}{\frac{2}{1+e^{\epsilon}}} = \frac{2 + 2e^{\epsilon} - 2}{2} = e^{\epsilon}.$$

Thus, the random algorithm f meets ε-differential privacy.

3.3.3 Privacy in Grouping Score

When we estimate the grouping score, the user-rated item list can be revealed. Therefore, we add noise based on Laplace mechanisms into the grouping estimation process. For this purpose, we define the grouping score sensitivity as follows.

Theorem 2.
The sensitivity of the grouping score is $\frac{k}{n} \cdot \frac{1}{\mathrm{m}} - \left(1 - \frac{k}{n}\right) \cdot \frac{1}{n_{\min} - 1}$ for the number of users n, the minimum number of the user-rated item $n_{\min}$, and the size of the group k.
Proof
A grouping score is a numerical value used to estimate the benefit when two users i and j belong to the same group. First, if users i and user j are the same user, then the sensitivity of the grouping score is evidently zero.

The grouping score sensitivity when users i and user j are not the same user is as follows.

$$\text{scoree}_i(j) = \frac{k}{n} \cdot \text{freq}(j) + \left(1 - \frac{k}{n}\right) \cdot \text{sim}(s_i, s_j). \quad (1)$$

Let us start with the first term in Eq. (1). Freq (j) = n_j/m represents the ratio of the number of user-rated items against the number of all items. If one of the items rated by user j is changed to n_{j-1} or n_{j+1}, then freq (j) will be (n_{j-1}/m) or (n_{j+1}/m). Therefore, in case of a difference in one rating detail, the difference between the first term of Eq. (1) is, at most, $k/n \cdot 1/m$.

Then, we explain the second term in Eq. (2). Let us assume that $m_{i\cup j}$ is a set of items that user i and user j commonly rated. When one rating detail t differs, $m_{i\cup j}$ is changed in four cases. (1) Add one item to $m_{i\cup j}$, (2) subtract one item to $m_{i\cup j}$, (3) add one item to $(m_{i\cup j})^C$, and (4) subtract one item to $(m_{i\cup j})^C$. We explain the similarity difference for each case.

(1) The similarity difference when adding one item to $m_{i\cup j}$ is as follows:

$$\begin{aligned}\text{sim}(s_i, s_j) = \frac{n_{i\cap j}}{n_{i\cup j}} &= \frac{n_{i\cap j}}{n_i + n_j - n_{i\cap j}} \to \text{sim}(s_i, s_j) \\ &= \frac{n_{i\cap j} + 1}{(n_i + n_j + 1) - (n_{i\cap j} + 1)}\end{aligned}$$

Therefore, the difference in the similarity is as follows:

$$\Delta_{\text{sim}} = \frac{n_{i\cap j} + 1}{n_i + n_j - n_{i\cap j}} - \frac{n_{i\cap j}}{n_i + n_j - n_{i\cap j}} = \frac{1}{n_i + n_j - n_{i\cap j}}.$$

The maximum value of the similarity difference is $\frac{1}{\max(n_i, n_j)}$ because of the maximum value of $n_{i\cap j}$ is $\min(n_i, n_j)$

(2) The similarity difference when subtracting one item to $m_{i\cup j}$ is as follows:

$$\begin{aligned}\Delta_{\text{sim}} &= \frac{n_{i\cap j}}{n_i + n_j - n_{i\cap j}} - \frac{n_{i\cap j} - 1}{n_i + n_j - n_{i\cap j}} = \frac{1}{n_i + n_j - n_{i\cap j}} \\ &= \frac{1}{\max(n_i, n_j)}.\end{aligned}$$

The others are the same as in Case (1).

(3) The similarity difference when adding one item to $(m_{i\cup j})^C$ is as follows:

$$\begin{aligned}\text{sim}(s_i, s_j) = \frac{n_{i\cap j}}{n_{i\cup j}} &= \frac{n_{i\cap j}}{n_i + n_j - n_{i\cap j}} \to \text{sim}(s_i, s_j) \\ &= \frac{n_{i\cap j} + 1}{(n_i + n_j + 1) - n_{i\cap j}}.\end{aligned}$$

The difference of the similarity is as follows:

$$\Delta_{\text{sim}} = \frac{n_{i\cap j}}{n_i + n_j - n_{i\cap j}} - \frac{n_{i\cap j}}{n_i + n_j - n_{i\cap j} + 1}. \quad (2)$$

For the sake of presentation convenience, let $n_i + n_j$ be X and $n_{i\cap j}$ be Y. Then, Eq. (2) is expressed as follows:

$$\Delta_{\text{sim}} = \frac{Y}{X - Y} - \frac{Y}{X - Y + 1} = \frac{Y}{(X - Y)(X - Y - 1)}. \quad (3)$$

The denominator of Eq. (3) can be summarized as follows:

$$(X - Y)(X - Y - 1) = Y^2 + (1 - 2X)Y + X^2 - X. \quad (4)$$

Equation (4) has minimal value when Y = X − 1/2. The denominator of Eq. (3) is minimized when Y is min (n_i, n_j), given that the maximum value of Y, min, (n_i, n_j) is less than $n_i + n_i - 1/2$. In addition, the maximum value of the numerator of Formula 3 is min (n_i, n_j). Therefore, Formula 3 can be summarized as follows:

$$\begin{aligned}\Delta_{\text{sim}} &= \frac{Y}{(X - Y)(X - Y - 1)} \\ &\le \frac{\min(n_i, n_j)}{\max(n_i, n_j) \cdot (\max(n_i, n_j) + 1)} \\ &\le \frac{\min(n_i, n_j)}{\min(n_i, n_j) \cdot (\max(n_i, n_j) + 1)} = \frac{1}{\max(n_i, n_j) + 1}.\end{aligned}$$

(4) The similarity difference when subtracting one item to $(m_{i\cup j})^C$ is similar with that in Case (3).

$$\begin{aligned}\Delta_{\text{sim}} &\le \frac{\min(n_i, n_j)}{\min(n_i, n_j) \cdot (\max(n_i, n_j) - 1)} \\ &= \frac{1}{\max(n_i, n_j) - 1}\end{aligned}$$

Among the four cases discussed above, the largest difference of the similarity function is $\frac{1}{\max(n_i, n_j) - 1}$. If the minimum value among the ratings of all users is $n_{\min}$, then $\frac{1}{\max(n_i, n_j) - 1} \le \frac{1}{n_{\min} - 1}$. Thus, the difference between the second term of Eq. (1) when one rating detail differs is maximum $\left(1 - \frac{k}{n}\right) \cdot \frac{1}{\max(n_i, n_j) - 1}$.

In summary, the sensitivity of the grouping score is as follows:

$$\Delta_{score} \le \frac{k}{n} \cdot \frac{1}{m} + \left(1 - \frac{k}{n}\right) \cdot \frac{1}{n_{\min} - 1}.$$

4. Experiment and Analysis

In this chapter, we validate the proposed technique in comparison with the existing method.

4.1 Experimental Environment

We use Super Micro Computer, Inc.'s SuperServer 7049P-TR (64-bit), consisting of CPU Intel Xeon Silver 4110 and 64 GB memory, and the operating system is Ubuntu 16.04.2 LTS. The proposed technique is implemented in Python 2.7.12.

We use the datasets MovieLens [16] and Book-Crossing [17] to evaluate the proposed techniques. These datasets are commonly used benchmark datasets in recommendation system research.

To evaluate the performance of the recommender system, we compare the original submatrix and predicted submatrix, which removes one arbitrary rating detail r_{ij} and performs matrix factorization. We explain our estimation measure with examples. We remove the rating information that user u_1 rates to item m_1 with five users $u_1, \ldots, u_5$ and

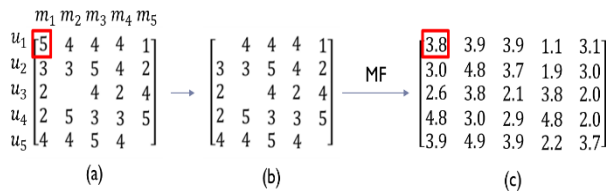

Fig. 7 Example of measurement

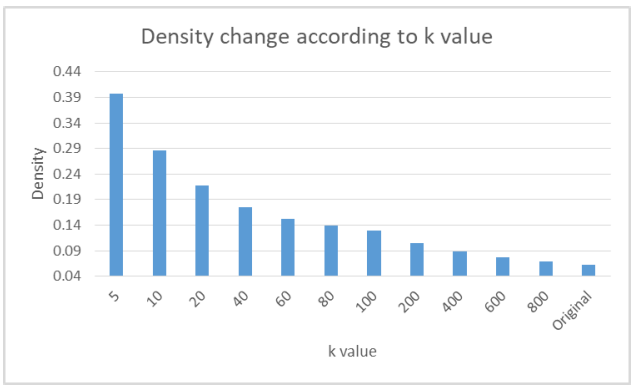

Fig. 8 Density change by group size

five items $m_1, \ldots, m_5$ in Fig. 7 a. Then, we perform a matrix factorization to the rating matrix (Fig. 7 b) that one rating information has been removed to obtain the prediction matrix (Fig. 7 c). We repeat this process for all other user-rated items and estimate the RMSE (Root Mean Square Error) between the prediction matrix and the original matrix for recommendation performance evaluation. RMSE is calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}$$

(n = Total number of rating information, r_i = original rating information i, $\hat{r}_i$ = predictive rating information i)

4.2 Experiment Result

4.2.1 Increased Density by Grouping Method

First, we measure the density of the submatrix according to the group size parameter k to confirm that the density increases through grouping. The density is calculated as follows:

$$density_i = \frac{\#\ of\ rating\ information_i}{\#\ of\ item_i \times \#\ of\ user_i}$$

(# of item_i = Number of items in $\text{submatrix}_\text{i}$, # of user_i = Number of users in $\text{submatrix}_\text{i}$, # of user_i = Number of rating information in $\text{submatrix}_\text{i}$,)

Figure 8 shows the result of measuring the average for density of all submatrix while changing the group size k. As shown in the figure, the density increases as the group size decrease. Thus, density is increased when the grouping algorithm is used as intended in this study.

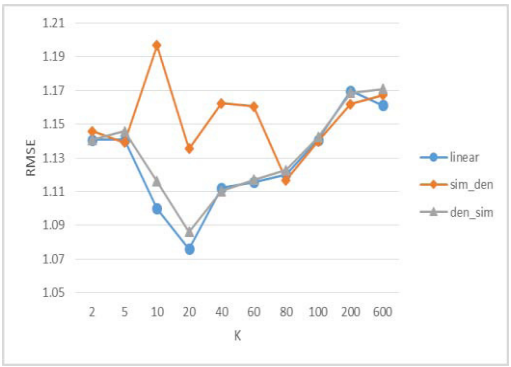

Fig. 9 Recommender performance comparison by grouping

4.2.2 Grouping Score

The proposed grouping score function simultaneously considers the user's rating frequency and the similarity of the rating pattern. The user's rating frequency and the similarity of the rating pattern can be reflected together in several ways. First, only the rating frequency is examined. Then, the users with a rating frequency lower than 10% are removed, and grouping can be performed in order of highest rating pattern similarity. This method is called den_sim. Second, the rating pattern similarity is taken into account first, and the rating similarity of the lower 10% is excluded. Then, the remaining users are grouped in order of highest rating frequency. This method is called sim_den. Finally, the linear method applies weight to both criteria and considers them simultaneously. We experiment with the three methods to determine the most efficient one.

As shown in Fig. 9, we can confirm that the linear method, which uses the weighted user's rating frequency and the similarity of the rating pattern, has the best performance. Therefore, we use the linear method to calculate the grouping scores.

4.2.3 Recommender Performance Increase by Grouping Method

In this section, we confirm the improvement in recommendation performance by grouping. We represent the matrix factorization technique used in the existing recommendation system as MF (Matrix Factorization), and the technique proposed in this paper is referred to as GM (Grouping Method).

We then determine whether the grouping method can enhance the recommendation performance. First, MF is used in existing recommender systems, whereas the technique proposed in this study is called the GM GM.

We compare recommendation performance without applying differential privacy when matrix factorization is applied to the rating matrix and the proposed grouping method. First, let us consider the effect of group size on recommendation performance. The smaller the group size, the higher

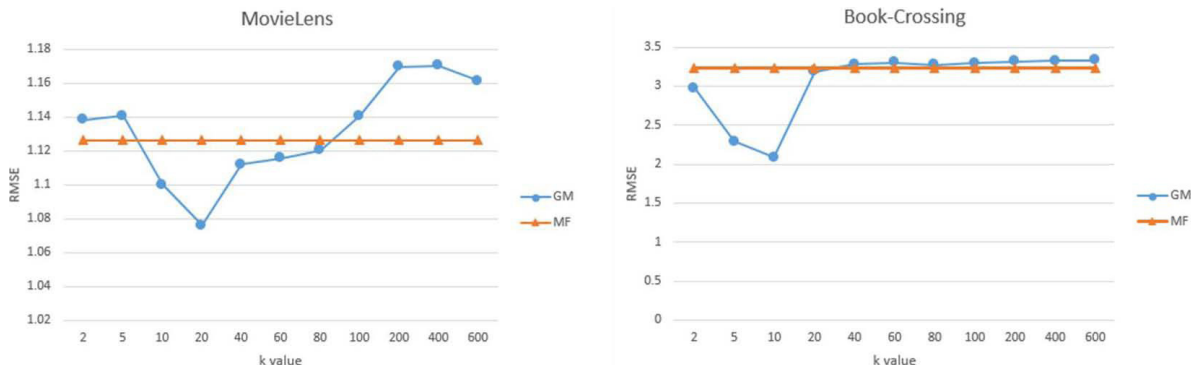

Fig. 10 Recommender performance comparison by group size

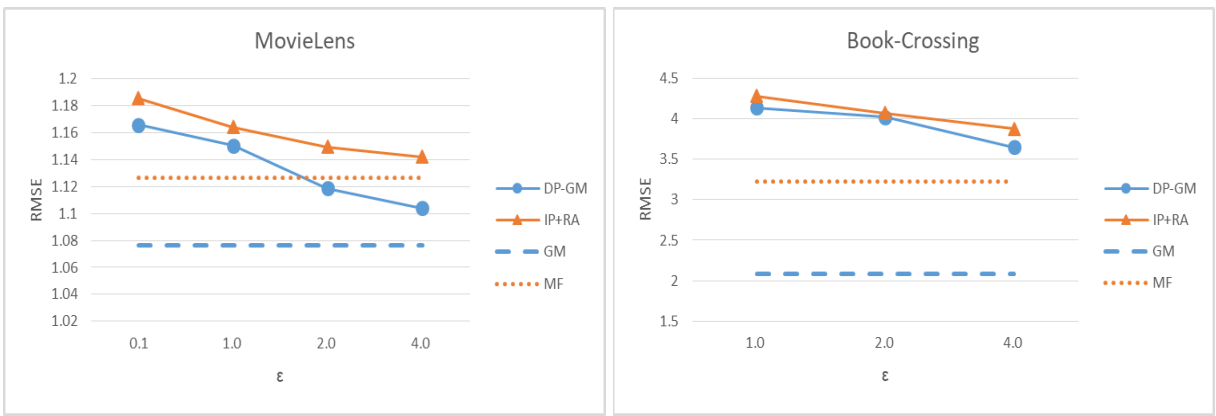

Fig. 11 Performance of recommender systems with consideration for privacy

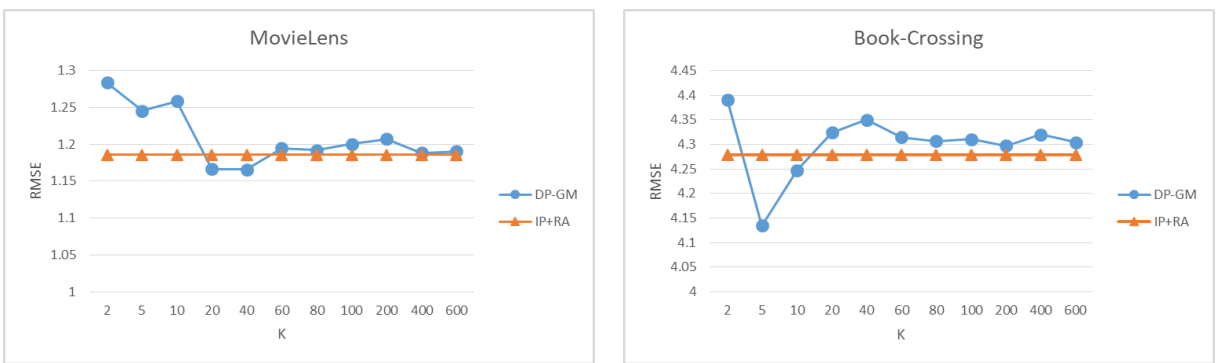

Fig. 12 Recommender performance comparison by group size

the density. However, the information about the rating pattern decreases. Conversely, if the group size increases, the information on the rating pattern increases, but the density decreases. Therefore, the appropriate group size k for grouping is important to the recommendation performance enhancement.

Figure 10 shows that the recommender performance is not favorable when the group size is very small or very large. The performance of the proposed technique in the MovieLense dataset is better than that of the previous recommender techniques until the group size k is 5 to 80. Among them, k shows the best performance when it is 20. In the BookCrossing dataset, the recommender performance is best when k is 10. For the MF, RMSE value is not affected by the k value because grouping is not performed. However, GM shows RMSE value is changed according to the k value as described earlier. This result shows that the appropriate group size is important for our proposed technique. We also evaluates the performance improvement of the proposed technique over the existing technique with the application of differential privacy. We represent the proposed technique as DP-GM (Differential Private Grouping Method) and the existing technique as IP+RA (Input Perturbation + Randomized Algorithm). In Fig. 11, the solid

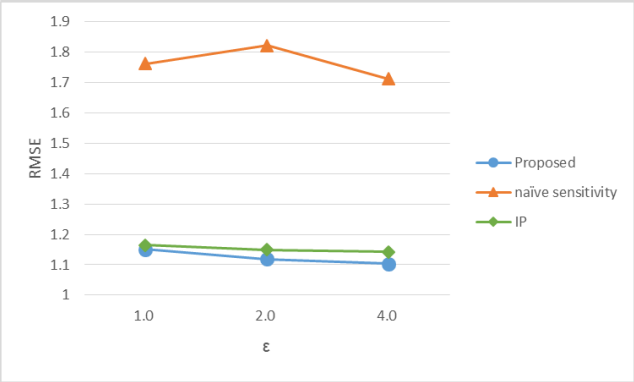

Fig. 13 Recommender performance comparison by sensitivity

lines denote the performance of existing and proposed techniques that consider privacy, and the dotted lines represent the performance of existing and proposed techniques that do not consider privacy. For both datasets, the proposed method GP performs better for all ε than the existing technique IP+RA. This result shows that the matrix sparsity has been reduced by grouping, and the performance enhancement using grouping remains when the noise is inserted for privacy protection. The experimental results also show that as the epsilon value increases, RMSE score approaches the original RMSE score which differential privacy is not applied.

In Fig. 12, we fix the privacy budget ε and change the group size k. For the MovieLens dataset, the optimum value is given when the group size is 20, as in the case without differential privacy. Additionally, the proposed technique's recommendation performance is better than that of the existing technique. For the BookCrossing dataset, the proposed method performs better than the existing technique when the group size is 5. This result verifies that performance improvements by grouping remain when the noise is inserted as same as the previous experiment. In addition, Fig. 10, 11, 12 results show that the appropriate group size k for grouping varies depending on the dataset's property. We cannot find out what features of the dataset affect the k value. Thus, it is our future work to find out how to decide an appropriate k value according to the dataset's property.

Figure 13 is the experimental result of sensitivity. Sensitivity is important for applying differential privacy based on the Laplace mechanism because it determines the amount of noise. In general, sensitivity is set to 1 in Laplace Mechasnim. However, the proposed technique can more accurately calculate the sensitivity less than 1 as shown in Theorem. 2 in Sect. 3.3.3, and the proposed sensitivity significantly reduces the amount of noise. Figure 13 shows the comparison between naïve sensitivity and proposed grouping sensitivity. We can validate that the proposed grouping sensitivity can improve the recommendation performance over that of the existing technique.

5. Conclusion

Web service users are overwhelmed by the vast amount of digital data on the web, and the amount of digital data is constantly growing. Therefore, the need for recommendation systems and recommendation algorithms is steadily increasing. However, the personal information used in recommendation systems is at risk of serious privacy breaches. In this study, we attempt to solve the sparsity problem in the user rating information for recommendation and consider the user-rated item list protection scheme, which is overlooked by the existing privacy protection scheme. For this purpose, we solve the sparsity problem through grouping and propose a differentially private recommendation algorithm, which considers user-rated item list information. We experimentally validate that the proposed method can improve the recommendation performance by the setting of an appropriate grouping size. In the future, we will conduct research on techniques to improve recommendation performance based on privacy budget allocation and grouping.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00498, A Development of De-identification Technique based on Differential privacy). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.NRF-2017R1D1A1B03036252).

References

[1] F. Bach, "Structured sparse methods for matrix factorization," http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.636.945&rep=rep1&type=pdf, 2011.
[2] N. Gillis, "Sparse and unique nonnegative matrix factorization through data preprocessing," Journal of Machine Learning Research, vol.13, pp.3349–3386, 2012.
[3] A. Narayanan and S. Vitaly, "Robust de-anonymization of large sparse datasets," University of Texas at Austin, 2008.
[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computers, vol.42, no.8, pp.30–37, 2009.
[5] C. Dwork, "Differential privacy: A survey of results," International Conference on Theory and Applications of Models of Computation, pp.1–19, 2008.
[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," International Conference on Theory of Cryptography, pp.265–284, 2006.
[7] A. Calandrino, A. Kilzer, A. Narayanan, E.W. Felten, and V. Shmatikov, "You Might Also Like: Privacy Risks of Collaborative Filtering," IEEE Symposium on Security and Privacy, pp.231–246, 2011.
[8] J. Canny, "Collaborative filtering with privacy," IEEE Symposium on Security and Privacy, pp.45–57, 2002.
[9] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," Proc. 2013

ACM SIGSAC conference on Computer & communications security, pp.801–812, 2013.
[10] F. McSherry and M. Ilya, "Differentially private recommender systems: Building privacy into the netflix prize contenders," Proc. 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.627–636, 2009.
[11] A. Berlioz, A. Friedman, M.A. Kaafar, R. Boreli, and S. Berkovsky, "Applying differential privacy to matrix factorization," Proc. 9th ACM Conference on Recommender Systems, pp.107–114, 2015.
[12] R. Balu and T. Furon, "Differentially private matrix factorization using sketching techniques," Proc. 4th ACM Workshop on Information Hiding and Multimedia Security, pp.57–62, 2016.
[13] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," IEEE Trans. Knowl. Data Eng., vol.30, no.9, pp.1770–1782, 2018.
[14] S. Niwattanakul, et al., "Using of Jaccard coefficient for keywords similarity," Proc. International MultiConference of Engineers and Computer Scientists, pp.380–384, 2013.
[15] S.L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," Journal of the American Statistical Association, vol.60, pp.63–69, 1965.
[16] MovieLens dataset, available: https://grouplens.org/datasets/movielens/100k/
[17] Book Crossing dataset, available: http://www2.informatik.uni-freiburg.de/~cziegler/BX/

Taewhan Kim was born in Seoul, Republic of Korea in 1994. He received the B.S. and M.S degrees in computer engineering from Sogang University in 2017, 2019. His research interests include privacy, social network and recommendation system.

Kangsoo Jung was born in Seoul, Republic of Korea in 1984. He received the B.S., M.S degrees and Ph.D. degree in computer engineering from Sogang University in 2007, 2009, 2017. He is a PostDoc in Sogang University. His research interests include privacy, access control and game theory.

Seog Park received the B.S. degree in computer science from Seoul National University, Korea, in 1978, the M.S. and the Ph.D. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST) in 1980 and 1983, respectively. He is a professor of computer science and engineering at Sogang University, Seoul, Korea. Since 1983, he has been working in the Department of Computer Science of the College of Engineering, Sogang University. Dr. Park is a member of the IEEE Computer Society, ACM, and the Korea Information Science Society. Also, he has been a member of Database Systems for Advanced Applications (DASFAA) steering committee from 1996 to 2007. His research interests include privacy and large data processing.