Expert Systems With Applications 168 (2021) 114366

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Differentially private user-based collaborative filtering recommendation based on *k*-means clustering

Zhili Chen [a,*], Yu Wang [a], Shun Zhang [a], Hong Zhong [a], Lin Chen [b]

[a] *Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, 230601 Hefei, China*

[b] *School of Data and Computer Science, Sun Yat-sen University, 510006 Guangzhou, China*

ARTICLE INFO

*Keywords:*
Differential privacy
*k*-means clustering
Recommendation system
Collaborative filtering

ABSTRACT

Collaborative filtering (CF) recommendation is well-known for its outstanding recommendation performance, but previous researches showed that it could cause privacy leakage for users due to *k*-nearest neighboring (KNN) attacks. Recently, the notion of differential privacy (DP) has been applied to privacy preservation in recommendation systems. However, as far as we know, existing differentially private CF recommendation systems degrade the recommendation performance (such as recall and precision) to an unacceptable level. In this paper, to address the performance degradation problem, we propose a differentially private user-based CF recommendation system based on *k*-means clustering (KDPCF). Specifically, to improve the recommendation performance, KDPCF first clusters the dataset into categories by *k*-means clustering and appropriately adjusts the size of the target category to which the target user belongs, so that only users in the well-sized target category are used for recommendation. Then, it selects efficiently a set of neighbors from the target category at one time by employing only one instance of exponential mechanism instead of the composition of multiple ones, and then uses a CF algorithm to recommend based on this set of neighbors. We theoretically prove that our system achieves differential privacy. Empirically, we use two public datasets to evaluate our recommendation system. The experimental results demonstrate that our system has a significant performance improvement compared to existing ones.

## 1. Introduction

Today's Internet is inundated by data, which makes it more and more difficult for users to quickly locate the data they look for. Recommendation systems are thus developed to facilitate the information search task by recommending pertinent data or resources to users. A class of efficient and personalized recommendation systems are based on collaborative filtering (CF) algorithms, which discover the potential consumption trend of users by mining users' historical data. Despite the good performance of CF-based recommendation systems, the involvement of users' historical data in the recommendation process may cause severe information leakage, as demonstrated by the *k*-nearest neighboring (KNN) attack (Frey, Guerraoui, Kermarrec, & Rault, 2018).

To address the privacy protection issue, Dwork, McSherry, Nissim, and Smith (2006) first proposed the notion of differential privacy (DP). It is well acknowledged that algorithms with DP can deal with background knowledge based attacks such as the differential attack. In view of its strong privacy guarantee, many researchers have recently applied DP to recommendation systems for privacy protection. However, simple applications of DP often lead to poor recommendation performances. As a result, how to balance between privacy protection and recommendation performance becomes the heart of the matter.

Previous user-based CF recommendation systems with DP suffer a great performance degradation due to the achievement of DP (Zhu, Li, Ren, Zhou, & Xiong, 2013; Zhu & Sun, 2016). Zhu et al. (2013) found that the mean absolute error (MAE) values of recommendations degraded too much after achieving differential privacy, and used local sensitivities other than global sensitivities to improve the recommendation performance. However, applying local sensitivities weakens the privacy guarantee (Dwork, Roth, et al., 2014). Zhu and Sun (2016) applied the advanced composition theorem to improve the recommendation performance, but their experimental results demonstrated that the recommendation recalls with DP degraded severely compared to the original ones. In practice, this recommendation performance degradation may be unacceptable.

[*] Corresponding author.
*E-mail addresses:* zlchen@ahu.edu.cn (Z. Chen), wangyu956622870@gmail.com (Y. Wang), szhang@ahu.edu.cn (S. Zhang), zhongh@mail.ustc.edu.cn (H. Zhong), chenlin69@mail.sysu.edu.cn (L. Chen).

https://doi.org/10.1016/j.eswa.2020.114366
Received 23 April 2019; Received in revised form 22 November 2020; Accepted 22 November 2020
Available online 24 November 2020

There are probably two main causes for the above performance degradation. First, when applying the exponential mechanism, too many unrelated or weakly related output items are defined for the differentially private recommendations, and these output items may overwhelm the accurate recommendation results due to the randomness of the exponential mechanism. Second, too much "noise" is added to the recommendations due to the composition of many differentially private algorithms. Having confirmed these causes, inspired by Chu, Tsai, Lee, and Pan (2018), we use the $k$-means clustering algorithm to design a differentially private recommendation system. Also, we reduce the number of applications of the exponential mechanism as much as possible to improve the recommendation performance.

In this paper, we use the $k$-means clustering algorithm to design a differentially private user-based CF recommendation system. First of all, our system clusters the users of the entire rating matrix $M$, and adjusts the size of the target category, which contains the target user. Next, it selects a neighbor set from the target category at one time, using the exponential mechanism. Finally, it predicts user ratings and gives top-$m$ items recommendations based on the selected neighbor set.

The contributions of this paper are summarized as follows:

(1) We come up with an approach based on the bisecting $k$-means to adjust the size of the target category, such that the target category is of an appropriate size, and hence both recommendation performance and privacy protection are reasonably balanced.

(2) We design a random neighbor selection to select a neighbor set at one time with only an application of the exponential mechanism instead of multiple ones. This algorithm adds much less noise to the proposed mechanism compared to using multiple instances of the exponential mechanism, and thus improves the recommendation performance.

(3) Based on (1) and (2), we propose a differentially private user-based CF recommendation system based on $k$-means clustering. Theoretically, we prove that our system satisfies $\epsilon$-differential privacy. Also, we have experimentally shown that our system provides a higher recommendation performance than existing systems.

The remainder of this paper is organized as follows. In Section 2, we briefly review related work. Section 3 introduces some basic concepts and theorems, and describes the privacy issue in collaborative filtering recommendations. We provide the detailed design of our system in Section 4. Section 5 discusses and analyzes our experimental results. Finally, we conclude the paper and make plans for future work in Section 6.

## 2. Related work

There have been a large amount of work on privacy preserving collaborative filtering (CF) recommendations, many of which were surveyed in Bilge, Kaleli, Yakut, Gunes, and Polat (2013) and Ozturk and Polat (2015). Here, we only give a very brief review of the related work in the following two groups.

**Traditional Privacy-preserving Recommendations.** In recommendation systems, traditional approaches to privacy protection are mainly based on cryptography, anonymization and random perturbation, respectively. The cryptography based approaches protect privacy for recommendation computations among multiple mutually distrustful parties using various cryptographic techniques (Armknecht & Strufe, 2011; Kaur, Kumar, & Batra, 2018). However, they normally incur heavy computational costs, especially when the amount of historical data is large. Instead, the anonymization based approaches normally preserve privacy by publishing anonymized data (Brickell & Shmatikov, 2008; Casino, Patsakis, & Solanas, 2019; Chang, Thompson, Wang, & Yao, 2010), and thus have much higher computational efficiency. Unfortunately, they cannot resist background knowledge attacks, and may weaken the utility of data severely (especially for high-dimensional data). The random perturbation based approaches protect privacy by adding noise to the original data in an ad hoc way (Lyu, Bezdek, Law, He, & Palaniswami, 2018; Polat & Du, 2003, 2005; Polatidis, Georgiadis, Pimenidis, & Mouratidis, 2017). They appear effective and practical, but cannot ensure rigorous theoretical privacy guarantees.

**Differentially Private Recommendations.** The notion of differential privacy has been applied to recommendation systems previously. McSherry and Mironov (2009) first applied differential privacy to a recommendation system, and proved that it is feasible to achieve differential privacy for the recommendation system without serious loss of accuracy. Hua, Xia, and Zhong (2015) first analyzed the privacy threats in various situations, concerning whether the recommender is trusted and whether it is online, and then designed a differentially private recommendation system based on matrix factorization. Meng et al. (2018) divided users' privacy into sensitive privacy and non-sensitive privacy in recommendations, and allowed users to add noise to their data locally. Similarly, Feng, Guo, and Chen (2016) and Zhu et al. (2013) improved the recommendation performance in view of recommendation-aware sensitivity or different levels of privacy. In this work, we aim to improve the recommendation performance of differentially private user-based CF recommendations mainly based on the $k$-means clustering (KDPCF).

The closest work to ours is Zhu and Sun (2016), in which the authors applied multiple exponential mechanisms to repeatedly select items to recommend in both item-based and user-based scenarios. Moreover, in order to improve the recommendation performance, they compared multiple similarity functions and selected the best one as the utility function for the exponential mechanism. However, the overall recommendation performance of this work still degraded too much. For this reason, in our system we employ a preprocessing based on $k$-means clustering and reduce the number of applications of exponential mechanism to merely one. It is worth noting that Lyu et al. (2018) achieved differential privacy also involving clustering, but they achieved differential privacy for trajectory clustering. Instead, our work uses $k$-means clustering as a preprocessing, and achieves differential privacy in CF recommendation systems.

## 3. Technical preliminaries

### 3.1. Differential privacy

Differential privacy (DP) is a new privacy model based on data distortion (Dwork et al., 2014). By adding controllable noise, DP guarantees the protection of personal information in a provable way, while ensures that the perturbed data have similar statistical properties to the original ones. Specifically, DP guarantees that adding a record to or deleting a record from a dataset does not have a significant influence on query results. Hence, even if an adversary knows all records except a sensitive one in the dataset, the protection of the sensitive record can be still ensured.

**Definition 1** (*$\epsilon$-Differential Privacy (Dwork, 2008; Dwork et al., 2014)*). A randomized algorithm $\mathcal{M}$ satisfies $\epsilon$-differential privacy if for any two neighboring datasets $t$ and $t'$ differing on at most one element, and for any set of outcomes $R \subseteq Range(\mathcal{M})$, $\mathcal{M}$ satisfies:

$$P[\mathcal{M}(t) \in R] \leq exp(\epsilon) \cdot P[\mathcal{M}(t') \in R]. \tag{1}$$

where $\epsilon$ is the privacy budget, which decides the privacy level of the algorithm. A greater $\epsilon$ means less noise added and thus a lower privacy level.

### 3.2. Exponential mechanism

Exponential mechanism (EM) (McSherry & Talwar, 2007) is a common technique for designing algorithms with differential privacy. EM is applicable to both numeric and non-numeric problems.

**Algorithm 1** $k$-means++ Algorithm (Arthur & Vassilvitskii, 2007)

```
Input: $k \leftarrow$ the number of clusters, $U \leftarrow$ user set, $\mathcal{D} \leftarrow$ data point set of users;
Output: $c_1, c_2, \dots, c_k \leftarrow k$ initial cluster centers;
Choose a user point randomly from $\mathcal{D}$ as the first initial cluster center $c_1$;
for all $i = 2 : k$ do
    Calculate the shortest distance $D(u)$ between each user and all current cluster centers;
    Sample every user $u \in U$ with probability $P(u)$, and make the user point as the next cluster center $c_i$
        $P(u) = \frac{D(u)^2}{\sum_{u\in U} D(u)^2}$
end for
```

**Definition 2** (*Exponential Mechanism (McSherry & Talwar, 2007)*). Given a quality function $q(t,r) : \mathbb{N}^{|\mathcal{X}|} \times \mathcal{R} \rightarrow \mathbb{R}$, $\triangle\, q$ is the sensitivity of quality function. The exponential mechanism $\mathcal{M}$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to $exp(\frac{\epsilon q(t,r)}{2\triangle q})$.

**Definition 3** (*Sensitivity of Exponential Mechanism (Dwork et al., 2014)*). The sensitivity of exponential mechanism is defined as follows:

$$\triangle\, q = \max_{r\in R}\ \max_{t,t' : \|t-t'\|_1 \le 1} \mid q(t,r) - q(t',r) \mid \tag{2}$$

where $q(t,r)$ is a quality function, $r \in \mathcal{R}$ is a valid output of the exponential mechanism.

The sensitivity of the function q indicates the greatest effect of a single data change on the output. The size of $\triangle\, q$ directly affects the amount of noise introduced by the algorithm. In default, $t$ and $t'$ represent any pair of neighboring datasets and the sensitivity is called *global sensitivity*. Sometimes, if we let $t$ represent the original dataset (and thus be fixed), and let $t'$ be any neighboring dataset of $t$, the sensitivity is called *local sensitivity*. Obviously, the global sensitivity is independent on the original dataset, while the local sensitivity is dependent. Thus, the former ensures a strong privacy, while the latter ensures a weak one.

### 3.3. *k*-means clustering

The $k$-means clustering has been used in recommendation systems without privacy guarantee to improve the recommendation efficiency (Kant, Mahara, Jain, Jain, & Sangaiah, 2017; MacQueen et al., 1967). With clustering, the selection of a target user's neighbors is limited in the category containing the target user, and thus the computational cost can be greatly reduced, especially when the user–item rating matrix is large. In our case, we use the $k$-means clustering to improve the sampling quality in the exponential mechanism, which in turn improves the recommendation performance.

In order to get a good clustering result using the $k$-means clustering, two parameters should be determined properly in advance: the number of clusters $k$ and the set of initial cluster centers. The value of $k$ is related to specific datasets and usually determined by approaches based on Silhouette coefficient (Sai, Shreya, Subudhi, Lakshmi, & Madhuri, 2017) or Elbow method (Syakur, Khotimah, Rochman, & Satoto, 2018). In our work, we adjust the value of $k$ such that the target category, to which the target user belongs, contains an appropriate number of users, so as to balance between performance and privacy. It is shown that the initial clustering centers should be selected uniformly to get a good clustering result (Arthur & Vassilvitskii, 2007). Thus, we use the $k$-means++ algorithm to determine the initial cluster center, as shown in Algorithm 1.

**Algorithm 2** User-based CF Recommendation

```
Input: $M \leftarrow$ user–item rating matrix, $m \leftarrow$ recommendation list length, $N \leftarrow$ neighbor set size, $u \leftarrow$ target user, and $I_u \leftarrow u$'s rating record.
Output: $R_u \leftarrow u$'s recommendation list.
Calculate similarities between user $u$ and all other users according to $M$.
for all $i \in I - I_u$ do
    Find user $u$'s neighbor set $\mathcal{N}_u$ with top $N$ users in term of similarities.
    Predict all of user $u$'s unrated scores based on its neighbor set $\mathcal{N}_u$.
end for
Add the top $m$ predicted items to user $u$'s recommendation list $R_u$.
```

**Table 1**
Privacy issue of user-based CF recommendations.

|  | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ | $i_6$ |
|---|---|---|---|---|---|---|
| $u_1$ | 2 |  | 4 |  | 5 |  |
| $u_2$ |  | 3 | 5 | 4 | ? |  |
| $u_3$ | 2 | 3 |  | 3 |  | 5 |
| sibyl |  | 3 | 5 | 4 |  |  |

### 3.4. Problem statement

We consider a user-based collaborative filtering (CF) recommendation (Chen, Uliji, Wang, & Yan, 2018; Ortega, Rojo, Valdiviezo, & Raya, 2018), where users submit their historical rating scores on items to a recommender. The recommender computes the rating similarities between a target user and other users, and then predict the target user' future rating scores. Let $M = \{M_{u,i}\}_{|U|\times|I|}$ denote the user–item rating matrix, where $U$ is the user vector, $I$ is the item vector, and $M_{ui}$ represents the historical rating score on item $i \in I$ given by user $u \in U$ if it is a non-zero value, and represents an unrated score otherwise. In practice, matrix $M$ is usually sparse, and the goal is to predict these missing ratings. The user-based CF recommendation can be described in Algorithm 2. The main idea is that users with similar preferences recommend items to each other.

In our user-based CF recommendations, the recommender is assumed to be trusted. A malicious user may receive recommendation results from the recommender, and analyze these results to infer the information of other users. Thus, the privacy issue mainly comes from the selection of a target user's neighbors. In order to make the recommendation results more accurate, the recommendation algorithm tends to select neighbors most similar to the target user. However, it is the very way of selecting neighbors that may cause a privacy breach.

Table 1 illustrates an example on the privacy issue mentioned above. Initially, the adversary knows a user $u_2$'s rating record $I_2 = \{0,3,5,4,0,0\}$, where each "0" represents an unrated item. Some time later, user $u_2$ rated an item $i_5$ and the adversary gets $I_2 = \{0,3,5,4,?,0\}$, where the "?" represents a new, unknown rating score. Now, the adversary can launch an attack to find which item has been rated newly as follows. It first registers a sibyl user in the system while keeping the sibyl user's rating record consistent with user $u_2$'s historical record, i.e. $I_s = \{0,3,5,4,0,0\}$. Next, the sibyl user sends a recommendation request to the system, and according to the algorithm, the sibyl user has a great chance of having user $u_2$ selected as its neighbor. Then, based on the recommendation results, the adversary can easily deduce which item is newly rated by $u_2$, and can even deduce the rating score of the item $i_5$ if further learning the similarity change between the sibyl and $u_2$.

The above privacy issue has been basically addressed before (Zhu & Sun, 2016). However, the previous work suffers a great degradation of recommendation performance, and thus is far from practical applications. In this work, we aim to improve the recommendation performance under the same privacy guarantees.

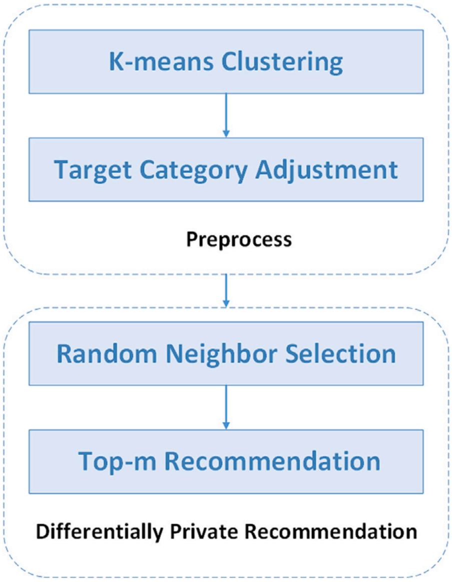

K-means Clustering
Target Category Adjustment
Preprocess
Random Neighbor Selection
Top-m Recommendation
Differentially Private Recommendation

**Fig. 1.** KDPCF algorithm flow diagram.

## 4. Our system

### 4.1. Overall design

In order to balance well between recommendation performance and user privacy, we design a differentially private user-based CF recommendation system based on $k$-means clustering (KDPCF). The main process of KDPCF is shown in Fig. 1, and the design rationale for every step is described as follows.

**Step 1. K-means Clustering.** In the rating matrix, the quantity of users is normally large, and usually only a small part of users are significant to the recommendation for a target user. Through the $k$-means clustering, such a part (called the target category) can be found, and the recommendation can be based upon these users.

**Step 2. Target Category Adjustment.** The target category resulted from $k$-means clustering may be in various sizes. An inappropriate category size (too big or too small) may cause performance or privacy issues. Thus, this step adjusts the target category with two size thresholds, $C_{max}$ and $C_{min}$, so that the category size falls in $[C_{min}, C_{max}]$.

**Step 3. Random Neighbor Selection.** The exponential mechanism is employed to select the neighbor set of a target user with differential privacy. To achieve a good recommendation performance, a neighbor set is selected at one time with an application of the exponential mechanism, instead of selecting the neighbors one by one with repeated applications of the exponential mechanism.

**Step 4. Top-m Recommendation.** Based on the neighbor set selected in the previous step, this step predicts the rating scores on the unrated items for the target user, and in terms of the predicted scores, the top-m items are recommended to the target user.

### 4.2. The detailed design

The algorithmic design of KDPCF is presented in Alg. 3. The details are described and discussed as follows.

**Step 1. K-means Clustering.**

This step applies $k$-means clustering described in Section 3.3 directly to the overall user–item rating matrix $M$. The matrix $M$ is normally very large, e.g., it may involve thousands of users and hundreds of items. Among these user ratings, usually only the ratings of a small part of users are significant to the recommendation for a target user. The idea is that $k$-means clustering can help to find out such a small part of users. In short, this step simply uses the similarity defined in Eq. (3) to perform $k$-means clustering over matrix $M$, and finds out the target category which contains the target user. It is noteworthy that, to overcome the curse of dimensionality, in the $k$-means clustering we use $D(.,.) = 1 - |Sim(.,.)|$ as the distance, where $Sim(.,.)$ represents the Pearson correlation coefficient as shown in Eq. (3).

**Algorithm 3** KDPCF Algorithm

```
Input: $M \leftarrow$ user–item rating matrix, $m \leftarrow$ recommendation list length,
    $N \leftarrow$ neighbor set size, $u \leftarrow$ target user, and $I_u \leftarrow u$'s rating record,
    $\epsilon \leftarrow$ privacy budget;
Output: $R_u \leftarrow u$'s recommendation list;
    Step 1. K-means Clustering.
    Perform $k$-means clustering on matrix $M$, and get the target category $C^*$;
    Step 2. Target Category Adjustment.
    Adjust category $C^*$ in size, and get the corresponding rating matrix $M^*$;
    Step 3. Random Neighbor Selection.
    Calculate the similarities between target user $u$ and other users $v \in C^*$:
    for all $(v \in C^* \wedge v \neq u)$ do
        Compute $Sim(u, v)$ with Eq. (3);
    end for
    Sample a random user set $U^*$ from $C^*$, and calculate the probability
    distribution on the family $\mathbb{N}$ of all possible neighbor sets of size $N$
    from the set $U^*$ as follows:
    for $\mathcal{N} \in \mathbb{N}$ do
        Compute $Pr(\mathcal{N})$ with Eqs. (4) and (5);
    end for
    Select a neighbor set $\mathcal{N}_u \in \mathbb{N}$ with the probability $Pr(\mathcal{N}_u)$;
    Step 4. Top-m Recommendation.
    Compute user $u$'s recommendation list $R_u$ of length $m$ from $\mathcal{N}_u$;
```

**Step 2. Target Category Adjustment.**

The purpose of applying $k$-means clustering is to limit the neighbor selection of a target user to its target category, and improve the recommendation performance. However, due to the randomness of $k$-means clustering, the resulted target category may be of various sizes. The size of the target category has a significant impact on both recommendation performance and user privacy. Specifically, if the target category is much larger than the neighbor set size, then the performance enhancement of the recommendation would be very limited, since there are still a large quantity of users in the category which overwhelm the neighbor selection. On the contrary, if the target category is close to or even less than the neighbor set size, the user privacy protection would be difficult to achieve, since there are too few neighbor sets to select, and the neighbor selection tends to be deterministic.

To address the category size issue above, this step adjusts the target category as follows. First, it sets the maximum and minimum size thresholds $C_{\max}$ and $C_{\min}$ to appropriate values. Then, it compares the size of the target category with the two thresholds: if the category size is greater than $C_{\max}$, it uses the target user as one of the clustering centers, and applies Algorithm 2 to perform bisecting $k$-means clustering again; if the category size is less than $C_{\min}$, it merges the target category with its nearest category. This process repeats until the target category size falls in the range from $C_{\min}$ to $C_{\max}$. Note that the size thresholds $C_{\min}$ and $C_{\max}$ can be set empirically. For example, in our experiments, $C_{\max}$ and $C_{\min}$ are set to be $10N$ and $5N$, respectively, where $N$ is the fixed size of neighbor sets.

**Step 3. Random Neighbor Selection.**

Given the target category of an appropriate size, this step turns to the random neighbor selection with differential privacy. The algorithm first computes the similarities between the target user $u$ and all other users in the target category. Then, based on these similarities, it randomly selects a neighbor set of the target user from the target category by the exponential mechanism to achieve differential privacy.

In recommendations, Pearson correlation coefficients are widely used to calculate similarities between users. Let $I_u$ and $I_v$ be the rating records of users $u$ and $v$, respectively, and a Pearson correlation coefficient can be used to calculate the similarity between $u$ and $v$ as follows:

$$Sim(u,v) = \frac{\sum_{i\in I_{uv}}(r_{ui}-\bar{r}_u)(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{uv}}(r_{ui}-\bar{r}_u)^2\sum_{i\in I_{uv}}(r_{vi}-\bar{r}_v)^2}} \tag{3}$$

where $\bar{r}_u$ and $\bar{r}_v$ represent the average rating scores of users $u$ and $v$, respectively (i.e. $\bar{r}_u = \sum_{i\in I_u} r_{ui}/|I_u|$, and $\bar{r}_v = \sum_{i\in I_v} r_{vi}/|I_v|$), $I_{uv}$ represents the intersection of rating records of users $u$ and $v$. It is noteworthy that $Sim(u,v) \in [-1,1]$, and its absolute value $|Sim(u,v)|$ indicates the strength of the correlation between users $u$ and $v$. Moreover, when the similarity is positive (resp. negative), it indicates that the two users are positively (resp. negatively) related.

In this step, we use Pearson correlation coefficients to measure similarities. Specifically, in Alg. 3, Lines from Lines 3 to 6 calculate the similarities between the target user $u$ and other users in the target category using Eq. (3). These similarities provide the basis for random neighbor selection.

To protect user privacy, the random neighbor selection should satisfy differential privacy. Furthermore, the recommendation performance should be cared about. For these purposes, our main idea is to select the neighbor set by the exponential mechanism, and carefully design the quality function.

Given the target category $C^*$, the target user $u$, and a user set $\mathcal{N} \subseteq C^* - \{u\}$, the quality of selecting the set $\mathcal{N}$ as user $u$'s neighbor set is given by:

$$q(C^*,u,\mathcal{N}) = \sum_{v\in\mathcal{N}} |Sim(u,v)| \tag{4}$$

Here, we treat positive and negative correlations in the same way, that is, we define the quality function as the sum of absolute values of similarities. According to the exponential mechanism, the probability of outputting the set $\mathcal{N}$ as the neighbor set is

$$Pr(\mathcal{N}) = \frac{exp(\epsilon q(C^*,u,\mathcal{N})/(2\Delta q))}{\sum_{\mathcal{N}'\in\mathbb{N}} exp(\epsilon q(C^*,u,\mathcal{N}')/(2\Delta q))} \tag{5}$$

where $\Delta q$ is the global sensitivity of quality function $q$. Indeed,

$$\Delta q = \max_{\mathcal{N}} \max_{\|C_1^*-C_2^*\|_1\le 1} |\, q(C_1^*,u,\mathcal{N}) - q(C_2^*,u,\mathcal{N}) \,| = 1 \tag{6}$$

where $C_1^*$ and $C_2^*$ are any two adjacent target categories that only differ in at most one user.

In the random neighbor selection, we apply the quality function defined in Eq. (4) to the exponential mechanism, and select the neighbor set of the target user at one time. Specifically, in Algorithm 3, Lines from 7 to 10 sample a random user set $U^*$ from the target category $C^*$, enumerate all possible neighbor sets of size $N$ from the user set $U^*$ to form a family $\mathbb{N}$ of neighbor sets, and calculate the probability distribution over the set $\mathbb{N}$. Afterwards, Line 11 simply selects a neighbor set with the calculated probability distribution. There are two points noteworthy as follows.

(1) Instead of selecting one neighbor with an application of the exponential mechanism repeatedly, selecting a neighbor set at one time avoids the composition of algorithms with DP, and hence induces less noise. Thus, we may achieve better recommendation performance.

(2) The two adjacent categories should contain exactly the same number of users and only one single user differs in its rating scores. For this we have to use the bounded differential privacy concept (Li, Lyu, Su, & Yang, 2016) in which any pair of neighboring sets have the same size.

**Step 4. Top-m Recommendation.**

Given the neighbor set $\mathcal{N}^*$ selected, this step constructs the recommendation list $R_u$ for the target user $u$. The algorithm first collects the rated items of all neighbors in $\mathcal{N}^*$ while excluding the rated items of $u$, and then predicts the rating scores of $u$ on the unrated items by

$$r_{ui}^* = \bar{r}_u + \frac{\sum_{v\in\mathcal{N}}^* Sim(u,v) * (r_{vi}-\bar{r}_v)}{\sum_{v\in\mathcal{N}}^* |Sim(u,v)|} \tag{7}$$

where $\bar{r}_u$, $\bar{r}_v$ represent the average rating scores of users $u$ and $v$, respectively, and $i$ represents the unrated items of $u$. Finally, in term of rating scores $r_{ui}^*$, the top-$m$ items are used to construct the recommendation list $R_u$ for $u$.

### 4.3. Privacy analysis

We establish the differential privacy of KDPCF in Theorem 1.

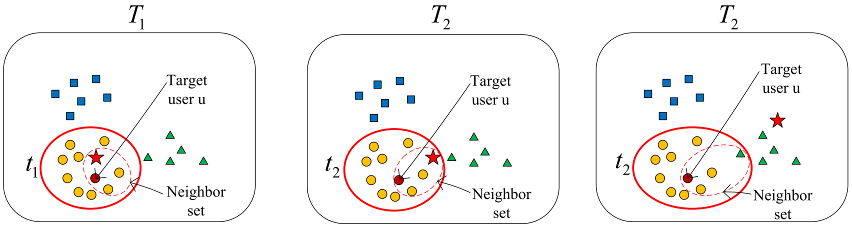
**Theorem 1.** *KDPCF (described in Alg. 3) satisfies $\epsilon$-differential privacy.*

**Proof.** Let $T_1$ and $T_2$ be a pair of adjacent datasets, that is, they contain exactly the same user records and only one single user differs in rating scores, and $t_1$ and $t_2$ represent the target categories of them, respectively, as shown in Fig. 2. We regard the steps of $k$-means clustering and target category adjustment as a preprocess for each target user $u$, that is, we let the differing happen after the clustering, and let target categories still contain exactly the same number of users, and thus they differ in at most a single user in rating scores. In Alg. 3, since the number of users of a target category is normally too big for enumerating all neighbor sets, only a subset $U^*$ of it is sampled, and based on $U^*$ a family $\mathbb{N}$ of neighbor sets are generated. According to the exponential mechanism, we can derive the probability ratio of any output $\mathcal{N} \in \mathbb{N}$ of the Random Neighbor Selection step as follows:

$$\begin{aligned}
&\frac{Pr[M_{RNS}(t_1)=\mathcal{N}]}{Pr[M_{RNS}(t_2)=\mathcal{N}]}\\
&= \frac{Pr(t_1,\mathbb{N})\cdot \dfrac{exp(\frac{\epsilon q(t_1,u,\mathcal{N})}{2\Delta q})}{\sum_{\mathcal{N}'\in\mathbb{N}} exp(\frac{\epsilon q(t_1,u,\mathcal{N}')}{2\Delta q})}}{Pr(t_2,\mathbb{N})\cdot \dfrac{exp(\frac{\epsilon q(t_2,u,\mathcal{N})}{2\Delta q})}{\sum_{\mathcal{N}'\in\mathbb{N}} exp(\frac{\epsilon q(t_2,u,\mathcal{N}')}{2\Delta q})}}\\
&= \left(\frac{exp(\frac{\epsilon q(t_1,u,\mathcal{N})}{2\Delta q})}{exp(\frac{\epsilon q(t_2,u,\mathcal{N})}{2\Delta q})}\right)\cdot\left(\frac{\sum_{\mathcal{N}'\in\mathbb{N}} exp(\frac{\epsilon q(t_2,u,\mathcal{N}')}{2\Delta q})}{\sum_{\mathcal{N}'\in\mathbb{N}} exp(\frac{\epsilon q(t_1,u,\mathcal{N}')}{2\Delta q})}\right)\\
&\le exp(\frac{\epsilon}{2})\cdot\left(\frac{\sum_{\mathcal{N}'\in\mathbb{N}} exp(\frac{\epsilon}{2})exp(\frac{\epsilon q(t_1,u,\mathcal{N}')}{2\Delta q})}{\sum_{\mathcal{N}'\in\mathbb{N}} exp(\frac{\epsilon q(t_1,u,\mathcal{N}')}{2\Delta q})}\right)\\
&\le exp(\frac{\epsilon}{2})\cdot exp(\frac{\epsilon}{2})\cdot\left(\frac{\sum_{\mathcal{N}'\in\mathbb{N}} exp(\frac{\epsilon q(t_1,u,\mathcal{N}')}{2\Delta q})}{\sum_{\mathcal{N}'\in\mathbb{N}} exp(\frac{\epsilon q(t_1,u,\mathcal{N}')}{2\Delta q})}\right)\\
&= exp(\epsilon).
\end{aligned}$$

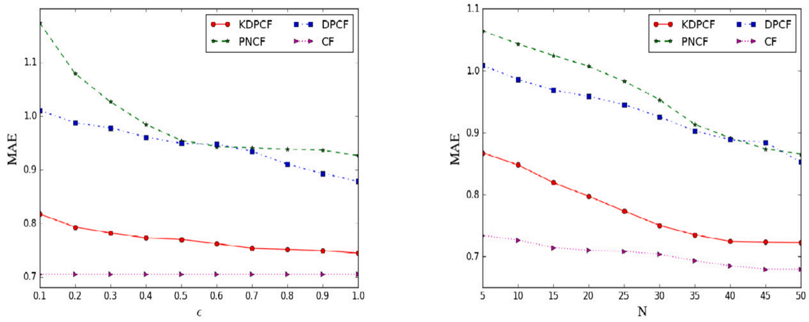
where $Pr(t_1,\mathbb{N})$ and $Pr(t_2,\mathbb{N})$ are the probabilities of sampling $\mathbb{N}$ from categories $t_1$ and $t_2$, respectively. Since both target categories have exactly the same number of users, and the sampling is independent on rating scores, it is noteworthy that we have $Pr(t_1,\mathbb{N}) = Pr(t_2,\mathbb{N})$.

So far, we can conclude that the first three steps of KDPCF satisfy differential privacy. Meanwhile, since the last step Top-m Recommendation is merely based on the output of the first three steps, it is actually a post-process. Therefore, due to the post-processing property of differential privacy, KDPCF satisfies differential privacy. This completes the proof. □



(a) (b) (c)

**Fig. 2.** An illustration of a pair of adjacent datasets $T_1$ and $T_2$, their target categories $t_1$ and $t_2$, the target user $u$ and one of its neighbor sets. (a) shows the original dataset $T_1$, its target category $t_1$, and one of $u$'s neighbor sets; A star point (representing a user's rating scores) is arbitrarily changed such that it still falls into the target category (b) or it falls out of the target category (c).



(a) MAE-ϵ (b) MAE-N

**Fig. 3.** Performance comparison of MAE on the MovieLens dataset.

### 4.4. Computational complexity

Now we discuss the computational complexity of KDPCF. The computational complexity of the first two steps (i.e., the preprocessing module) is simply the same as that of $k$-means clustering, which is $O(t \cdot k \cdot d \cdot |U|)$ where $t$ is the number of iterations, $k$ is the number of clusters, $d$ is the dimensionality of user records (i.e., the number of items), and $|U|$ is the number of users. Normally, $t$, $k$ and $d$ can be regarded as constants, so the complexity becomes $O(|U|)$. Let $|\mathbb{N}|$ denote the number of neighbor sets sampled. The computational complexity of the last two steps (i.e., the differentially private recommendation module) is $O(d \cdot N \cdot |\mathbb{N}|)$. If the neighbor size $N$ is also regarded as a constant, the second complexity becomes $O(|\mathbb{N}|)$. Thus, the total computational complexity of KDPCF is $O(\max(|U|, |\mathbb{N}|))$. It is easy to find that the user-based recommendation scheme in Zhu and Sun (2016) is of computational complexity $O(|U|)$, when $d$ is regarded as a constant. As long as the number of neighbor sets $|\mathbb{N}|$ sampled is constrained in $O(|U|)$, our system has the same computational complexity as that of Zhu and Sun (2016) scheme.

## 5. Performance analysis and evaluation

In our experiments, we use two datasets which are commonly used in recommendation systems: MovieLens and Netflix. The former contains 100,000 ratings by 943 users on 1682 items. The format of records is like (user ID, item ID, rating, timestamp), where the field of timestamp is not used in the experiments. Each user rated at least 20 items and the rating range is 1–5. The latter is of the same record format, but its total amount of data is much larger. Therefore, we choose to select a part of the data for the experiments, including the ratings of the 5675 users for the first 1500 items, and the number of ratings for each user is not less than 100. In order to verify the accuracy of the results, we divide the data into two parts: the training set and the testing set, with a ratio of 4 : 1. Finally, considering the randomness of differentially private algorithms, all experimental results take the average of 100 runs.

We compare our system with that proposed by Zhu et al. (2013) and that proposed by Feng et al. (2016) separately and with different performance indicators. In the comparison with the system proposed by Zhu et al. (2013), we use the mean absolute error (MAE) as the performance indicator, which is defined as follows:

$$MAE = \frac{\sum_{u \in U, i \in I'} |r_{ui} - r^*_{ui}|}{|I'|} \tag{8}$$

where $I'$ means the set of rated items in the testing set, $r_{ui}$ is the real user rating that exists in the testing set, and $r^*_{ui}$ is the predicted rating. So MAE refers to the error between the predicted rating and the true rating.

In the comparison with the system proposed by Feng et al. (2016), recall and precision are used to measure recommendation performances. In the context of recommendations, recall and precision are defined as follows:

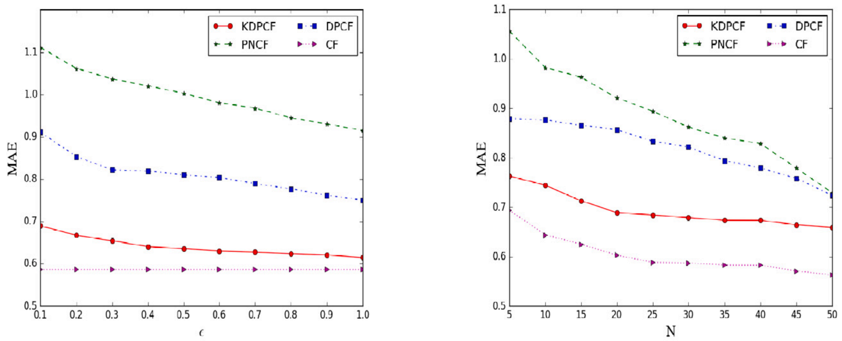
$$Recall = \frac{\sum_{u \in U} |R_u \cap T_u|}{\sum_{u \in U} |T_u|} \tag{9}$$

$$Precision = \frac{\sum_{u \in U} |R_u \cap T_u|}{\sum_{u \in U} |R_u|} \tag{10}$$

where $R_u$ is the recommendation list of user $u$ provided by recommendation schemes based on the training set, and $T_u$ is the rating list of user $u$ in the testing set.

In order to evaluate the recommendation performance, we set $k = 2|U|/(C_{min}+C_{max})$ in our recommendation system, KDPCF, and compare it with the following several schemes in term of MAE, recall and precision.

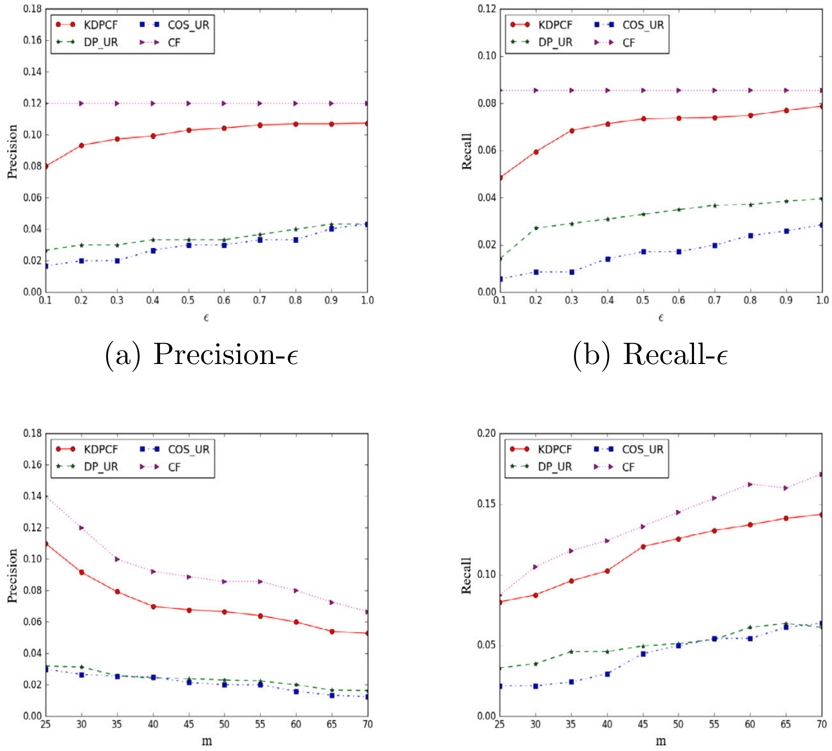
- **CF**: The original user-based collaborative filtering recommendation system without differential privacy guarantee.
- **PNCF**: The Private Neighbor Collaborative Filtering recommendation system proposed by Zhu et al. (2013). Different from Zhu and Sun (2016), the local sensitivity is used in PNCF.
- **DPCF**: In this system, differential privacy is directly applied to the collaborative filtering recommendation algorithm without any



(a) MAE-$\epsilon$

(b) MAE-N

**Fig. 4.** Performance comparison of MAE on the Netflix dataset.



(a) Precision-$\epsilon$

(b) Recall-$\epsilon$

(c) Precision-m

(d) Recall-m

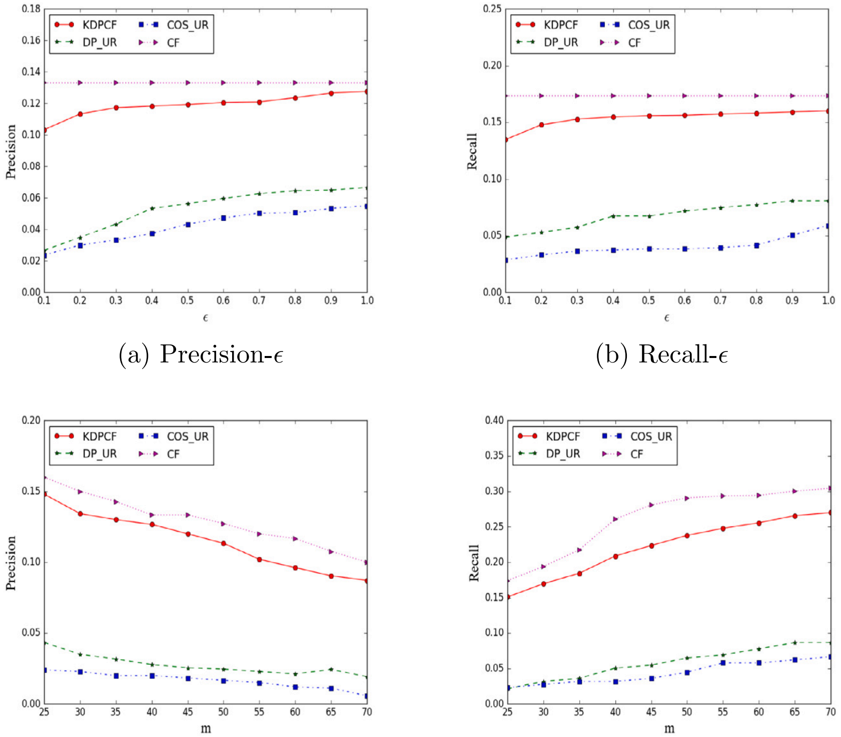
**Fig. 5.** Performance comparison of Precision and Recall on the MovieLens dataset.

optimization. Since this system is mainly used for comparison with PNCF, the local sensitivity is also used.

- **DP_UR**: The differentially private user-based collaborative filtering recommendation scheme using multiple exponential mechanisms. It is essentially the same as the DP_UR system proposed by Zhu and Sun (2016), except that, for comparison fairness, we implement it using the simple composition theorem instead of the advanced composition theorem. It is noted that DP_UR uses global sensitivity.
- **COS_UR**: In order to facilitate the comparison between KDPCF and DP_UR, we design an intermediate scheme COS_UR, which is basically consistent with the DP_UR scheme. The only difference is that equation Eq. (7) is used instead of the equation in DP_UR when predicting the user rating.
- **KDPCF**: This is the system proposed in this paper. It should be noted that in order to ensure the comparison fairness, in the experiments comparing with PNCF, the local sensitivities are used, while in those comparing with DP_UR, global sensitivities are still used.

In Figs. 3 and 4, we compare the recommendation performances among four systems, CF, DPCF, PNCF and KDPCF under the following conditions: (1) when the neighbor set size $N$ varies; and (2) when the privacy budget $\epsilon$ varies. Similarly, we also explored the recommendation performance of the systems CF, DP_UR, COS_UR and KDPCF when the privacy budget $\epsilon$ and the recommendation list length $m$ varies in Figs. 5 and 6. In default, we set recommendation list length $m = 30$, the neighbor set size $N = 30$, and the privacy budget $\epsilon = 1$.

Figs. 3 and 4 compare the performances of the four systems in term of the MAE. It is easy to find that no matter using the MovieLens dataset or the Netflix dataset, our system is significantly better than DPCF that directly uses differential privacy and PNCF proposed by Zhu et al. (2013). It should be noted that with the privacy budget $\epsilon$ and the number of neighbors $N$ increase, PNCF is even worse than DPCF.



(a) Precision-$\epsilon$

(b) Recall-$\epsilon$

(c) Precision-m

(d) Recall-m

**Fig. 6.** Performance comparison of Precision and Recall on the Netflix dataset.

The main reason for this is that in addition to the noise caused by the exponential mechanism, a large amount of noise generated by the Laplace mechanism is also added to PNCF, which seriously affects the accuracy of the rating prediction.

In Figs. 5 and 6, we can see that as the increase of privacy budget $\epsilon$, both precision and recall values raise. This is because the increase of privacy budget results in the reduction of noise. However, for the recommended item size $m$, the precision and recall have the opposite trends. The reason is that, increasing $m$ means to enlarge the list of $R_u$, and due to Eqs. (9) and (10) the precision would drop but the recall would increase. These trends indicate that the two performance indicators, recall and precision are complementary. Additionally, although the performance of DP_UR is slightly better than that of COS_UR, it is much worse than that of our system. This result benefits from the preprocessing process which ensures that the selected neighbors through the exponential mechanism have relatively high similarities.

## 6. Conclusion

In this paper, we have proposed a novel differentially private user-based collaborative filtering recommendation system based on $k$-means clustering, KDPCF, addressing the performance degradation issue. Specifically, we apply $k$-means clustering, and adjust the target category to an appropriate size, so that recommendations are limited within the well-sized target category. As a result, we can balance well between recommendation performance and user privacy. We also employ a subset sampling to randomly select a neighbor set from the target category with a single application of the exponential mechanism, and perform recommendations based on this neighbor set, which further improves the recommendation performance under the same privacy levels. Theoretically analysis shows that our system achieves differential privacy, and experimental evaluations demonstrate that our system improves the performance significantly upon previous systems. In this work, the way of sampling a family of neighbor sets suffers a certain loss of recommendation performance due to the random selection of a user set $U^*$. The future work is to improve the sampling way to further improve the recommendation performance.

## CRediT authorship contribution statement

**Zhili Chen:** Conceptualization, Methodology, Formal analysis, Writing - original draft. **Yu Wang:** Data curation, Software, Writing - original draft. **Shun Zhang:** Visualization, Investigation. **Hong Zhong:** Supervision. **Lin Chen:** Writing - review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The authors thank the anonymous reviewers for their helpful comments. The work of Zhili Chen, Yu Wang and Hong Zhong is partially supported by the Support Program for Outstanding Young Talents in Anhui Universities, China (No. gxyq2019001), the Special Fund for Key Program of Science and Technology of Anhui Province, China (No. 18030901027), and the National Natural Science Foundation of China (NO. 61572031). The work of Shun Zhang is partially supported by the National Natural Science Foundation of China (No. 11301002), Anhui Provincial Natural Science Foundation, China (2008085MF187), and Natural Science Foundation for the Higher Education Institutions of Anhui Province of China under Grant KJ2018A0017.

## References

Armknecht, F., & Strufe, T. (2011). An efficient distributed privacy-preserving recommendation system. In *Ad hoc networking workshop (Med-Hoc-Net), 2011 the 10th IFIP annual mediterranean* (pp. 65–70). IEEE, http://dx.doi.org/10.1109/Med-Hoc-Net.2011.5970495.

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms* (pp. 1027–1035). Society for Industrial and Applied Mathematics, http://dx.doi.org/10.1145/2492517.2492519.

Bilge, A., Kaleli, C., Yakut, I., Gunes, I., & Polat, H. (2013). A survey of privacy-preserving collaborative filtering schemes. *International Journal of Software Engineering and Knowledge Engineering*, *23*(08), 1085–1108.

Brickell, J., & Shmatikov, V. (2008). The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 70–78). ACM, http://dx.doi.org/10.1145/1401890.1401904.

Casino, F., Patsakis, C., & Solanas, A. (2019). Privacy-preserving collaborative filtering: A new approach based on variable-group-size microaggregation. *Electronic Commerce Research and Applications*, *38*, Article 100895.

Chang, C.-C., Thompson, B., Wang, H. W., & Yao, D. (2010). Towards publishing recommendation data with predictive anonymization. In *Proceedings of the 5th ACM symposium on information, computer and communications security* (pp. 24–35). ACM, http://dx.doi.org/10.1109/Med-Hoc-Net.2011.5970495.

Chen, J., Uliji, Wang, H., & Yan, Z. (2018). Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering. *Swarm and Evolutionary Computation*, *42*, 173. http://dx.doi.org/10.1016/j.swevo.2018.08.008.

Chu, P.-M., Tsai, H.-R., Lee, S.-J., & Pan, S.-T. (2018). Improving collaborative filtering recommendation. In *2018 3rd international conference on control, automation and artificial intelligence (CAAI 2018)*. Atlantis Press, http://dx.doi.org/10.2991/caai-18.2018.25.

Dwork, C. (2008). Differential privacy: A survey of results. In *International conference on theory and applications of models of computation* (pp. 1–19). Springer, http://dx.doi.org/10.1007/978-3-540-79228-4_1.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference* (pp. 265–284). Springer, http://dx.doi.org/10.1007/11681878_14.

Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, *9*(3–4), 211–407. http://dx.doi.org/10.1561/0400000042.

Feng, T., Guo, Y., & Chen, Y. (2016). A differential private collaborative filtering framework based on privacy-relevance of topics. In *Computers and communication (ISCC), 2016 IEEE symposium on* (pp. 946–951). IEEE, http://dx.doi.org/10.1109/ISCC.2016.7543858.

Frey, D., Guerraoui, R., Kermarrec, A. M., & Rault, A. (2018). Collaborative filtering under a sybil attack. In *IEEE/IFIP international conference on dependable systems and networks*. http://dx.doi.org/10.1145/2751323.2751328.

Hua, J., Xia, C., & Zhong, S. (2015). Differentially private matrix factorization. In *IJCAI* (pp. 1763–1770). http://dx.doi.org/10.1145/2792838.2800191.

Kant, S., Mahara, T., Jain, V. K., Jain, D. K., & Sangaiah, A. K. (2017). Leaderrank based k-means clustering initialization method for collaborative filtering. *Computers and Electrical Engineering*, Article S0045790617312909. http://dx.doi.org/10.1016/j.compeleceng.2017.12.001.

Kaur, H., Kumar, N., & Batra, S. (2018). An efficient multi-party scheme for privacy preserving collaborative filtering for healthcare recommender system. *Future Generation Computer Systems*, Article S0167739X17327012. http://dx.doi.org/10.1016/j.future.2018.03.017.

Li, N., Lyu, M., Su, D., & Yang, W. (2016). Differential privacy: From theory to practice. *Synthesis Lectures on Information Security, Privacy, & Trust*, *8*(4), 1–138. http://dx.doi.org/10.2200/S00735ED1V01Y201609SPT018.

Lyu, L., Bezdek, J. C., Law, Y. W., He, X., & Palaniswami, M. (2018). Privacy-preserving collaborative fuzzy clustering. *Data &amp; Knowledge Engineering*, *116*, 21–41.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1*, Oakland, CA, USA (pp. 281–297.

McSherry, F., & Mironov, I. (2009). Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 627–636). ACM, http://dx.doi.org/10.1145/1557019.1557090.

McSherry, F., & Talwar, K. (2007). Mechanism design via differential privacy. In *Foundations of computer science, 2007. FOCS'07. 48th annual IEEE symposium on* (pp. 94–103). IEEE, http://dx.doi.org/10.1109/FOCS.2007.66.

Meng, X., Wang, S., Shu, K., Li, J., Chen, B., Liu, H., et al. (2018). Personalized privacy-preserving social recommendation. In *Proc. AAAI conf. artif. intell.*.

Ortega, F., Rojo, D., Valdiviezo, P., & Raya, L. (2018). Hybrid collaborative filtering based on users rating behavior. *IEEE Access*, *6*, 69582–69591. http://dx.doi.org/10.1109/ACCESS.2018.2881074.

Ozturk, A., & Polat, H. (2015). From existing trends to future trends in privacy-preserving collaborative filtering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *5*(6), 276–291.

Polat, H., & Du, W. (2003). Privacy-preserving collaborative filtering using randomized perturbation techniques. In *Third IEEE international conference on data mining* (pp. 625–628). IEEE.

Polat, H., & Du, W. (2005). Svd-based collaborative filtering with privacy. In *Proceedings of the 2005 ACM symposium on applied computing* (pp. 791–795).

Polatidis, N., Georgiadis, C. K., Pimenidis, E., & Mouratidis, H. (2017). Privacy-preserving collaborative recommendations based on random perturbations. *Expert Systems with Applications*, *71*, 18–25.

Sai, L. N., Shreya, M. S., Subudhi, A. A., Lakshmi, B. J., & Madhuri, K. B. (2017). Optimal K-means clustering method using silhouette coefficient. http://dx.doi.org/10.5958/0975-8089.2017.00030.6.

Syakur, M., Khotimah, B., Rochman, E., & Satoto, B. (2018). Integration K-means clustering method and elbow method for identification of the best customer profile cluster. *IOP Conference Series: Materials Science and Engineering*, *336*, Article 012017. http://dx.doi.org/10.1088/1757-899X/336/1/012017.

Zhu, T., Li, G., Ren, Y., Zhou, W., & Xiong, P. (2013). Differential privacy for neighborhood-based collaborative filtering. In *Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining* (pp. 752–759). ACM, http://dx.doi.org/10.1145/2492517.2492519.

Zhu, X., & Sun, Y. (2016). Differential privacy for collaborative filtering recommender algorithm. In *Proceedings of the 2016 ACM on international workshop on security and privacy analytics* (pp. 9–16). ACM, http://dx.doi.org/10.1145/2875475.2875483.