Deriving ratings from a private P2P collaborative scheme

Murat Okkalioglu[1] and Cihan Kaleli[2*]
[1] Computer Engineering Department, Yalova University
Yalova, 77200 - Turkey
[e-mail: murat.okkalioglu@yalova.edu.tr]
[2] Computer Engineering Department, Eskisehir Technical University
Eskisehir, 26470 - Turkey
[e-mail: ckaleli@eskisehir.edu.tr]
*Corresponding author: Cihan Kaleli

Received March 12, 2018; revised December 21, 2018; revised March 1, 2019; accepted April 8, 2019; published September 30, 2019

Abstract

Privacy-preserving collaborative filtering schemes take privacy concerns into its primary consideration without neglecting the prediction accuracy. Different schemes are proposed that are built upon different data partitioning scenarios such as a central server, two-, multi-party or peer-to-peer network. These data partitioning scenarios have been investigated in terms of claimed privacy promises, recently. However, to the best of our knowledge, any peer-to-peer privacy-preserving scheme lacks such study that scrutinizes privacy promises. In this paper, we apply three different attack techniques by utilizing auxiliary information to derive private ratings of peers and conduct experiments by varying privacy protection parameters to evaluate to what extent peers' data can be reconstructed.

Keywords: Privacy, data reconstruction, auxiliary information, peer-to-peer, collaborative filtering

http://doi.org/10.3837/tiis.2019.09.008 **ISSN : 1976-7277**

1. Introduction

Since most of the human routines have already moved to online avenues and people prefer to perform their daily activities at their comfort zone almost any time, the Internet is an indispensable part of daily life. One can watch movies, order a meal, perform banking transactions, arrange a business meeting, read the news or get in touch with their acquaintances without even bothering oneself to go out. Such ease in any transaction attracts people to prefer online transactions. Therefore, online transaction volume has been continually increasing. Today, most of the online services offer almost unlimited contents for its users. Remember that a person visiting a movie or book rental shop before the Internet era could access limited content provided by the local store. On the other hand, online services offer various options that anyone is flooded with the abundance and variety of options. When it comes to deciding to pick which item to buy or prefer, people become hesitant because they are overwhelmed by the number of options. *Information overload* or *infobesity* defines this phenomenon; it becomes challenging for people to decide when they are surrounded by too much information.

Collaborative filtering (CF) systems aim to overcome the information overload problem by utilizing different algorithms. A mainstream direction in CF is to find out similar users (neighbors), for an active user (AU) who is looking for a prediction with a claim that an unobserved behavior (rating) of AU would be discovered by examining tastes of similar-minded users. In a traditional CF system, users provide their taste of preferences about different items in terms of scalar or binary ratings. These ratings are usually kept in $m \times n$ matrix, where m is the number of users while n is the number of items. A large and sparse matrix is very common in CF systems since there might be plenty of items, and users are not expected to have an idea for a variety of different items. The accuracy of a CF system mostly relies on enough participation: the denser the matrix is, the more relations can be explored between users. Even if some users had shared their ratings with a CF system, they might have concealed their real opinions. Users might not feel comfortable to provide their true opinions due to privacy concerns. Users' data could be compromised, and users might face with different threats such as government surveillance, price discrimination, unsolicited marketing or even their data can be sold in case of bankruptcy [1, 2].

Privacy-preserving collaborative filtering (PPCF) systems take privacy into their primary consideration while producing CF result. Main concentration in PPCF systems is to produce predictions with privacy without neglecting accuracy. In a typical scenario in PPCF, each user perturbs her ratings and sends it to a central server. Since predictions are produced based on perturbed data, accuracy is affected by the privacy level desired by users. As the privacy level increases, accuracy is expected to decline because predictions are produced on degenerated data.

Central server-based systems might still lack enough data to produce predictions although privacy is promised. In some cases, new companies in the market or start-ups could need data for more reliable predictions. In such cases, companies can collaborate to form a richer matrix, which can enhance accuracy. They could collaborate on horizontally, vertically or arbitrarily distributed data [3-5]. The general idea in distributed PPCF schemes is that data holders keep users' original data and try to prevent their disclosure from the parties they collaborate. Therefore, each party applies data perturbation methods to their data; as a result, data disclosure could be kept at a minimum rate. However, users supplying ratings for such data

holders, either central or distributed manner, could decide to operate their peer-to-peer (P2P) network in order to generate prediction without the need for data holders. Contrary to central or distributed schemes, users (peers) hold their data and take part in the recommendation process by sharing computational workload [6]. P2P CF with privacy concern on binary ratings is proposed in [6], and scholars employ a Naïve Bayes Classifier (NBC) approach for private prediction by utilizing randomized response technique (RRT) [7] to ensure privacy.

Although privacy promises are made in PPCF schemes, the degree of privacy should be scrutinized. Some scholars in the PPCF community study and devise different attacks to derive the original data from the perturbed one masked by different PPCF schemes. These studies so far focus on central and partitioned PPCF schemes [8-13] for binary and numerically rated data. Inspired by these studies, this paper aims to focus on a binary P2P PPCF scheme proposed in [6] to investigate to what extent privacy promises are kept. The degree of privacy provided in the targeted PPCF scheme is investigated by three different data reconstructions attacks to derive the original rating vectors of each peer. Attacks in this paper are performed by an active peer (AP) who is looking for a prediction. The targeted PPCF scheme [6] operates on a P2P network with binary ratings. The scheme in detail requires that collaborating peers send their partial conditional probability values to the *AP*. Therefore, data reconstruction attacks, which aim to build peers' rating vectors, in this study exploit such a transfer of interim values in different scenarios to derive private ratings of collaborating peers. In the first data reconstruction attack, *alienate the victim*, a malicious *AP* dispatches a contemplated query where a single item is differentiated from the rest by an opposite rating. It can be easily achieved thanks to binary ratings. Therefore, the malicious *AP* can analyze partial conditional probability values received from each collaborating peer to derive the alienated item's rating. The second attack, *perfect match*, exploits specific partial probability values. The attacker obtains that the related collaborating peer with a perfect match has either identical or opposite ratings with the *AP* for the commonly rated items. The original ratings of these items can be derived if new perfect matches can be captured in repeated queries. The last attack, *acting as an active user*, which the targeted private binary scheme has measures against, is well-known in the literature and can be considered as a benchmark. The attacker stores partial similarity values for a reference query and alters a rating from the reference query each time a new query is dispatched. Such a change in the reference query reveals the original rating of the altered item by monitoring the temporal change in the partial conditional probability compared to the initially stored value. The attacker does not know the peers' rating of the queried item, q, which is crucial for the reconstruction attacks; therefore, the auxiliary information is utilized in this paper to overcome this bottleneck. The contribution of this study is essential for the PPCF community because, to the best of our knowledge, this is the first study covering a data reconstruction attempt to derive confidential peer vectors perturbed by a P2P PPCF scheme. In this paper, the attacks are experimentally analyzed with two data sets by varying privacy protection parameters offered by the targeted private P2P scheme. Our empirical outcome shows that our data reconstruction attacks outperform *acting as an active user* attack.

The paper is organized as follows. In the following section, the related work in the literature is given. Section 3 describes the targeted P2P binary PPCF scheme. Section 4 gives our attack techniques and their applicability for the targeted scheme. Section 5 lays out the experimental results while the last section covers the conclusion of the study.

2. Related Work

CF systems might lack true user participation if users believe that their privacy could be compromised. Privacy could be regarded as a crucial issue to ensure user participation. Canny [1] proposes a scheme in which each user computes public aggregate data without disclosing own privacy. Polat and Du [14] utilize randomized perturbation to offer predictions for numerically rated data. The authors convert numeric ratings into z-scores and add a random number drawn from uniform or Gaussian distribution to them. Polatidis et al. [15] utilize randomized perturbation with multi-level privacy, where users pick a random privacy level. A random noise associated with the level of privacy is added to the original rating. The scholar in [16] state that uniform privacy guarantee for all users is not necessary; therefore, they propose a personalized private protocol where users individually calibrate their privacy level. Based on stated user preferences, the proposed protocol add different Laplace noise to perturb individual similarities. Johnson-Lindenstrauss transform is applied in [17] to preserve privacy, and the authors theoretically show that their method satisfies ϵ-privacy. Xiong et al. [18] propose a private recommendation scheme where each user defines a symbol set and maps their ratings onto their symbol set by utilizing an exponential mechanism for differential privacy. They propose a private pattern correlation metric to find similarities between two users whose symbol sets differ. A central server-based binary PPCF scheme is conducted in [19]. The authors use RRT, which is originally a survey technique to determine the percentage of a population who has a sensitive attribute [7]. NBC is employed in [20] for binary rated PPCF systems. This study could be considered as the earlier version of the P2P binary PPCF scheme targeted in this paper. The authors assume that users are the features and ratings are the feature values. They also use RRT for privacy purposes and discuss how NBC-based predictions can be carried out in such a scenario. Since data sparsity is a significant bottleneck in producing recommendations for PPCF systems, new companies in the market would like to collaborate with other parties to enhance their data sets. In the PPCF community, scholars propose two-, multi-party, and P2P solutions. In [21, 3], two-party binary PPCF schemes are studied for horizontally (HPD) and vertically partitioned data-based (VPD) scenarios. The proposed algorithms select qualified users as neighbors, and the authors provide predictions considering two different neighbor selection scenarios for the HPD-based scheme. In the VPD-based PPCF scheme, the authors handle two cases depending on which party or parties hold items among which the predictions are being made. Kaleli and Polat [5] propose multi-party binary PPCF schemes, which utilize NBC. They devise schemes that allow parties to share partial conditional probabilities while preserving privacy. The context of this study is related to binary PPCF systems; further details about the PPCF literature can be found in a comprehensive survey by Bilge et al. [22].

Although privacy-enhanced schemes proposed in PPDM, these schemes are studied by various scholars in terms of the claimed privacy promises. Agrawal and Srikant [23] and Agrawal and Aggarwal [24] study to reconstruct the original distribution from the perturbed data. Kargupta et al. [25, 26] argue that randomness does not necessarily introduce uncertainty. They study a reconstruction approach based on spectral filtering (SF) and claim that theoretical boundaries of eigenvalues of the noise matrix can be obtained in randomization. Similarly, other scholars [27] conduct principal component analysis to exploit data correlations to reconstruct the original data. Guo et al. [28] study the bounds of reconstruction error when SF-based approaches are utilized. In terms of PPCF, various scholars devise attack techniques to derive the original data as well. Zhang et al. [8] perform singular value decomposition (SVD) and k-means clustering-based attacks to reconstruct original ratings

perturbed by randomization [14]. Auxiliary information could be critical to derive information from CF systems, which intend to keep users' data confidential. Calandrino et al. [29] target online CF systems, infer information by utilizing auxiliary information and monitoring the public output of such systems. In [9], authors target a central server based PPCF scheme to derive rated items only by exploiting auxiliary information. Similarly, scholars aim to find out which items are rated in [11] if ratings are numeric. They enhance their method with auxiliary information. Other than central-based schemes, the claimed privacy of partitioned PPCF schemes is also studied. Numerically rated HPD- and VPD-based schemes are studied under different attack scenarios in [11]. They aim to design attack scenarios by utilizing auxiliary information. Binary partitioned VPD- and HPD-based PPCF schemes are targeted in [12, 13] with different attacks. Attacks proposed in these studies form a basis for this study. Two attacks in that work can be extended to our targeted P2P scheme with NBC. In [30], a detailed survey is given about reconstruction techniques. The study breaks down the reconstruction techniques based on their method of application.

3. Preliminaries

The targeted P2P binary PPCF scheme proposed by Kaleli and Polat [6] utilizes NBC to provide predictions. Their approach is based on an NBC-based algorithm [31] for central CF systems. In this NBC scheme, users are features, and their ratings for items are feature values. Kaleli and Polat [6] apply this notion for P2P PPCF systems. Given the features, the probability of an item belonging to a class, c_j, where j is *dislike* or *like*, is given in Eq. (1) [6]:

$$p(c_j \mid f_1, f_2, \cdots, f_n) \propto p(c_j) \prod_{u=1}^{n} p(f_u \mid c_j) \tag{1}$$

In Eq. (1), $p(c_j)$, the prior probability of each class can be calculated from the active query, and f_u is the rating of q by peers where q is the item to be predicted. The scheme is initiated by request from *AP* to other peers in the prediction process. Peers who rate q are eligible to participate in the prediction process. Peers let *AP* know if they would like to join. Then, *AP* sends her rating vector and q to participating peers. Each peer calculates $p(f_u|c_j)$ and sends it to *AP*. *AP* assigns q either *like* or *dislike* after collecting the partial probability values from peers and calculating $p(c_j)$. However, the exchange between *AP* and other peers must be performed privately to avoid any disclosure. In this regard, privacy is generally handled in two aspects in PPCF [19]. The first aspect of privacy is that actual ratings should be masked so that other peers are not allowed to know them. The second aspect of privacy should prevent others from disclosing if an item is rated or not.

The scholars in [6] propose to use RRT [7] for data manipulation to offer the first aspect of privacy. RRT is a survey technique to discover the prevalence of a sensitive attribute by generally asking a polar question whose answer is either yes or no. Respondents of the survey use a random device generating an output between 0 and 1. If the output is less than a predefined threshold value, θ, the respondent gives a correct answer. Otherwise, an opposite answer is given for the sensitive question. *AP* adapts RRT by splitting the vector into multi-groups [6]. For each peer, *AP* first picks a random M_i from the range [2, M] to partition the rating vector where i is the related peer and M is the maximum number of groups that the vector can be split. After M_i is determined, *AP* picks a random θ_{ik} for each group to determine a threshold where k=1, 2, …, M_i – 1, M_i. Then, r_{ik} is picked for each group, and the rating vector for the k-th group is preserved if $r_{ik} \leq \theta_{ik}$. Otherwise, ratings are reversed, which means

that *like*s are converted to *dislike*s and vice versa. In addition to the data masking, Kaleli and Polat [6] also consider the second aspect of privacy by a data hiding method where unrated items are filled up to a random percentage between 0 and 100 (δ_{AP}). The steps of P2P CF algorithm with privacy are listed below.

Data hiding

- *AP* finds the number of her unrated items m_{ur}.
- *AP* pick random integers α_{AP} and δ_{AP} between the range of [0, 100] and [0, α_{AP}], respectively.
- *AP* picks δ_{AP} percent ($m_{ur} \times \delta_{AP}/100$) of unrated items and fills half of them with *dislikes* and the rest with *likes*.

Data hiding step fills unrated cells up to a random percentage without considering *AP*'s rating density. There might be a case where *AP*'s rating vector is very sparse, and α_{AP} and δ_{AP} are very close to 100. In such a case, most of the ratings in the vector are filled with unrelated random *like*s and *dislike*s. To avoid such an issue, scholars in [5] presents *hiding rated items* (HRI) protocol to associate the size of inserted ratings with the rating vector density, *d*. In HRI, *AP* determines *d* for its vector and chooses a random number, *L*, over the range (1, δ_{AP}], where δ_{AP} is a factor of *d* such as 0.5*d*, *d*, 2*d*, 4*d*, or 8*d*. *L* percent of unrated cells of the rating vector is filled with *like*s and *dislike*s based on a filling method, which could be either *random filling* (RF) or *default voting* (DV). RF method randomly fills unrated entries either *like* or *dislike* while DV fills a corresponding cell with a default value, which is dominant in the rating vector. For example, if *AP*'s rating vector has more *like*s than *dislike*s, then the default vote will be *like*. δ_{AP} determines the privacy level; however, note that larger values would diminish accuracy.

Although privacy measures are taken to keep *AP*'s ratings confidential, the authors discuss a possible data disclosure scenario if *AP* acts in a malicious manner [6]. Since peers who only have *q* can participate in the prediction process, *AP* could disclose the second aspect of privacy, rated items by peers, by asking repeated queries. The authors propose an extra privacy measure for peers to avoid such an incident. Each peer in the P2P network performs a random coin toss to join the prediction process. Therefore, half of the peers join the prediction process regardless of the status of *q*, rated or not. Peers who did not rate *q* yet joining the prediction process fill the unrated *q* with their default vote. In another scenario, *AP* could act maliciously by manipulating one item each time a new query is dispatched. *AP* could learn other peers' rating for the manipulated item by tracking temporal changes in subsequent queries. This attack is known as *acting as an active user* in multiple scenarios [6]. The authors indicate that such a case could be avoided if peers apply a data hiding method like HRI. Each time *AP* asks for a prediction, peers respond with a different vector by inserting some ratings into their vector. Henceforth, the first privacy measure for participating peers will be called *peer privacy for participation* (PPP), and the second privacy measure will be called *peer privacy for ratings* (PPR).

4. Deriving Private Data from P2P PPCF

AP collects interim probability values from other participating peers, and such an interchange can be exploited to derive confidential data. In this section, three different attacks are discussed, and their application on a binary P2P CF system with privacy [6] is examined. These attacks initially designed as if there were no privacy measures. We aim to show to what extent privacy metrics prevent from these attacks. The attacks in this paper exploit conditional

probability values calculated by collaborating peers. In order to compute the final probability value of *q*, belonging to class *like* or *dislike*, Eq. (1) requires collaborating peers to send their conditional probability values to *AP*. Although such partial conditional probabilities are aggregate values, we argue that a malicious *AP* can utilize them to derive peers' ratings. The attacks, on the other hand, differ in how they exploit partial conditional probability values. The first attack, *alienate the victim*, singles out the rating of a victim item from the rest. Such an alienation of an item in the query reveals its partial conditional probability value. This attack can be used in NBC-based schemes. The second attack, *perfect match attack*, exploits condition probability values if the transferred conditional probability reveals a perfect match which indicates that every corresponding item of an *AP* and a collaborating peer is identical or opposite. This attack can be applied to various binary private schemes [3, 21] under different partitioning scenarios. The last attack in this section, *acting as an active user*, is well known in the community. It monitors temporal changes in conditional probabilities when the rating vector is manipulated by one item rating. Besides, the targeted private scheme [6] only takes privacy measures considering *acting as an active user* attack. This attack can be considered as a benchmark to analyze our proposed attacks.

4.1 Alienate the Victim Attack

In this type of attack, a malicious *AP* picks a victim item and discloses it by checking the partial conditional probability on that item. *AP* constructs a query vector by alienating the victim item from the remaining items of the query with an opposite rating. To illustrate, imagine that the victim item is rated as *like* while the rest of the rated items in the query are rated as *dislike*. Peers participating in a prediction process calculate the partial conditional probabilities for each group. If there is a victim item in one of the groups, one of the conditional probabilities, $p(f_u|like)$ or $p(f_u|dislike)$, in that group is calculated by only using the victim item so that *AP* can figure out the rating of the victim item relative to the value of *q*.

Alienate the victim attack for P2P CF scheme is depicted in **Fig. 1**. Notice that *AP*'s vector has two groups for each peer for convenience and there is only one victim item for each group marked by a star (*). Partial conditional probabilities marked with red are calculated for each group by each peer for illustration purposes. If *q* is rated by the related peer, the peer can join and calculate the partial conditional probability. When *AP* receives a conditional probability from a peer, *AP* knows that the related peer rated *q* because only peers who rate *q* can participate in the prediction process. In **Fig. 1**, the partial conditional probabilities for p_1 are calculated for the first and the second group. $p(f_u|like)$ is calculated by utilizing the victim item in the first group because the only item rated (*like*) is the victim item in the query. Therefore, the denominator part of the partial conditional probability could be 0 or 1. If it is 0, *AP* can disclose that the related peer does not rate the victim item. Otherwise, *AP* discloses that the victim item is rated. After realizing that the victim item is rated, *AP* could check the numerator to reveal the rating for the victim item. The numerator could also be either 1 or 0. If it is 1, it means that the victim item is rated identically with respect to *q*; otherwise, it is rated opposite to *q*. To sum up, the denominator tells that whether the victim item is rated; the numerator tells the value of the victim item relative to *q*. However, *AP* cannot figure out the actual rating made for the victim item because *AP* does not know the rating that p_1 has for *q*. Therefore, we utilize auxiliary information, which is discussed in Section 4.4, to overcome this problem.

Alienate the victim attack could also be applied in multiple scenarios. A different victim item is picked in every successive active query. The malicious *AP* can learn about many victim items so that a portion of the user-item matrix can be built. The malicious *AP* can also create a relative coding map of items for each peer.

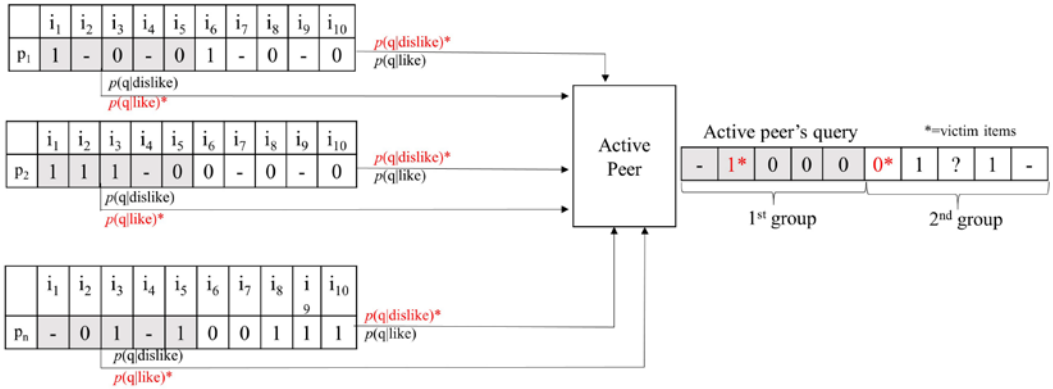

Fig. 1. Alienate the victim attack

4.2 Perfect Match Attack

In *perfect match attack*, *AP* initiates a prediction process as usual and tracks specific partial conditional probability values where the denominator is not 0, and denominator and numerator are equal or additive inverse. Such a condition will hereafter be called as a *perfect match*, and it reveals that the similarity between peers who have perfect matches is 1 or -1. A perfect match can be illustrated as in **Fig. 2**. In the figure, every corresponding rating in *AP*'s and related peers' query is either identical or opposite unless any of them is unrated. If every item is identical, such a relationship is a positive perfect match. If they are opposite to each other, such a relationship is a negative perfect match.

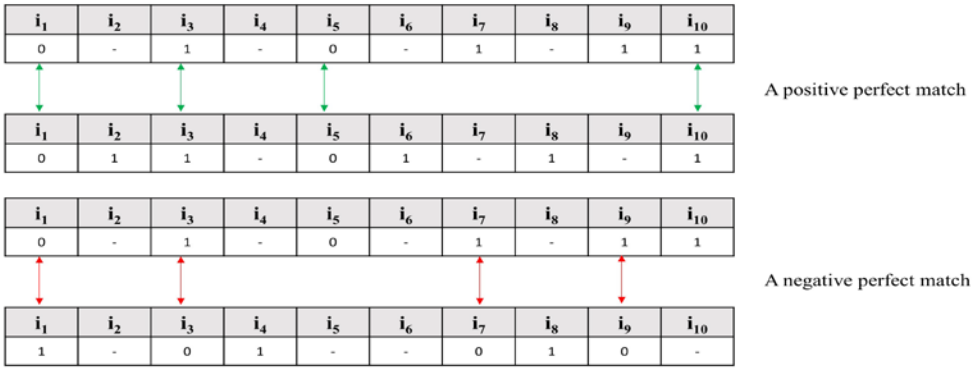

Fig. 2. A perfect match

A *perfect match attack* is depicted in **Fig. 3**. *AP* asks for a participation request from all other peers, then monitors partial conditional probability values to capture a perfect match between its rating vector and the participating peers. In **Fig. 3**, once *AP* captures a positive perfect match with p_1, the first query in **Fig. 3**, *AP* creates a vector for p_1 and marks its items either with the corresponding values of *AP*'s query or unrated. For example, i_1, i_2, i_4, and i_5 are filled with 0-, 1-, 1- and 1- where 0- and 1- mean *dislike* or unrated and *like* or unrated, respectively. When the second perfect match with p_1 is captured, i_3 is marked 1- and i_5 is proved to be unrated. Notice that i_5 was previously marked as 1- and it must be marked 0- after the second perfect match. Since a rating for an item cannot be *like* and *dislike*, it is now proved to be unrated. The repeated process of this attack will help infer the information of actual ratings or rating statuses of items. Since all participating peers send their partial conditional probabilities, *AP* can recover this information for all peers. *AP* does not know the related

peer's rating for *q*; therefore, *AP* cannot figure out the exact rating for items in the perfect match. Items are marked relative to *q*. Thus, we utilize auxiliary information similar to *alienate the victim* attack. *Perfect match* attack utilized for two-party binary PPCF schemes [12, 13]. In this paper, it is extended for the P2P binary PPCF scheme [6] as well with auxiliary information.

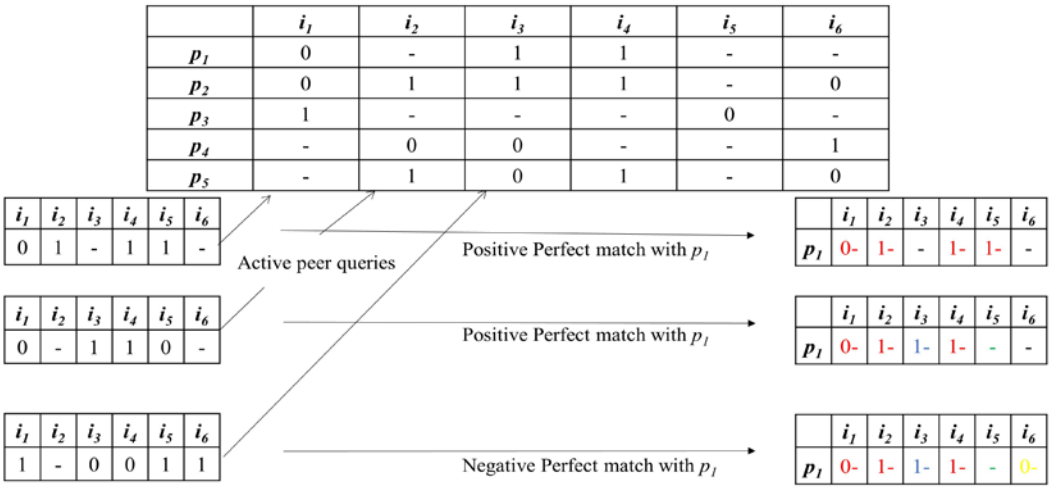

Fig. 3. Perfect match attack

4.3 Acting as an Active User Attack

Acting as an active user attack is well-known in the PPCF literature, and the targeted scheme [6] has measures considering this attack technique. The idea in this attack is that *AP* sends multiple queries to infer private information by monitoring temporal changes in the interim results. A malicious *AP* starts with an initial rating vector and stores returning conditional probabilities for future references. One item in the initial query is reversed in the next query so that the rating of the manipulated item can be derived by comparing the incoming conditional probability with one stored from the initial query.

Fig. 4 illustrates this attack in detail. *AP* sends subsequent queries to collaborating peers that differ by only one rating from the reference query. When *AP* receives conditional probabilities for each subsequent query, she compares it with the probabilities that are stored for reference. If there is an increase in the probability, the related item is identical to the value of *q*. If there is a decrease, the related item is opposite of the value of *q*. If it stays the same, then it is unrated. Similar to the other two attacks, the auxiliary information is utilized to estimate the rating of *q*.

4.4 Exploiting Auxiliary Information

For all attack types in this study, *AP* needs to know the rating of *q* by the related peer to disclose the actual rating. This rating of *q* cannot be obtained by *AP*. Each peer holds *q*, and it is not transferred to *AP* because *AP* does not need it to calculate the final probability. However, the use of auxiliary information about the data set might help overcome this issue. The data sets used for this study are movie related, MovieLensMillion, MLM, (www.cs.umn.edu/research/GroupLens) and YahooMovie [32]. Both data sets are on a numeric scale. Internet Movie Database (IMDB) is a well-known, reliable and reference website that movies are hosted and rated by a large number of users. For MLM data set,

movie-related basic statistics such as average movie ratings, the number of user votes and number of awards are collected almost for all movies from IMDB. To test another auxiliary information, we utilize the global non-personalized popularity (GNPP) score calculated by Yahoo for YahooMovie data set. GNPP is included in the data set.

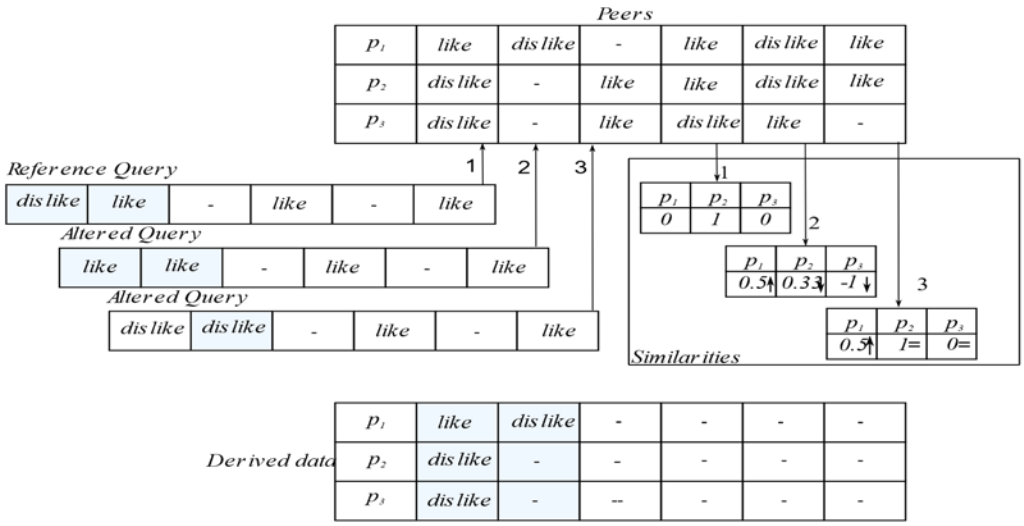

Fig. 4. Acting as an active user attack

Since *AP* does not know the rating of *q* by other peers, the auxiliary information can be utilized to have an idea about the rating of *q*. Having initiated a query, *AP* might assume that the rating for *q* by a peer is similar to the average rating collected from IMDB and GNPP score for MLM and YahooMovie data sets, respectively. Although such auxiliary information is helpful for *AP* to suggest a rating value about *q*, the downside is that each peer is assumed to rate *q* with the same value. To alleviate this obstacle, movies with the highest number of votes and ratings in IMDB are selected as *q* so that *AP* does not ask for a prediction for a random *q* whose rating might frequently differ among peers. For example, if *AP* asks for a prediction for a movie whose average rating is 6 out of 10 in IMDB and rated by few users, it is possible that the movie is not rated similarly or even unrated by many peers. Therefore, movies whose number of votes are higher than 500,000 in IMDB are selected for MLM data set. Among these movies, ones whose ratings are higher than 8.5 or less than 4.0 are selected to promote a consensus among peers. However, no movie less than 4.0 meets this criterion. For YahooMovie data set, movies whose GNPP value, which is between 1.29 and 13, is greater than ten are selected as eligible movies for *q*.

5. Experiments

In this section, we performed experiments to test the success of the reconstruction methods. Both *AP*'s and collaborating peers' privacy measures in [6] are considered to analyze the performance of the attacks. In the following subsections, data sets, the methodology of the experiments and experimental results are presented.

5.1 Data sets

Experiments are performed using MovieLens Million (MLM) and YahooMovie [32] data sets. MLM was collected by GroupLens research group (www.cs.umn.edu/research/GroupLens).

Its density is about 4.26%. MLM contains about a million ratings from 6,040 users for 3,883 items. YahooMovie data set is a sparse data set with a density of 0.23%. It contains 211,231 ratings from 7,642 users for 11,916 movies. Both data sets are on a numeric scale. MLM ratings are between 1 and five while YahooMovie ratings are between 1 and 13. Besides, YahooMovie data set contains converted ratings from 1-13 to 1-5. However, the concentration in this paper is on binary ratings. Thus, ratings were converted to a binary scale [31]. Ratings greater than or equal to 3 are converted to *like* (1), and the rest are converted to *dislike* (0).

5.2 Evaluation criterion and methodology

Precision (*prec*) is used as an evaluation criterion. *Prec* is the ratio of how much of the reconstructed items are identical to the original ones. *Prec* is only calculated for reconstructed *like*s and *dislike*s. Since data sets are sparse, integrating unrated items into evaluation criteria would be misleading. **Table 1** displays the confusion matrix after the reconstruction, and Eq. (2) gives *prec* calculation. We do not include *accuracy* as our evaluation criteria because both data sets are sparse, and accuracy does not tell much about the success of our reconstruction methods. For example, YahooMovie's density is 0.23% even if our reconstruction methods do not achieve to derive a single rating; the accuracy would be greater than 99% due to the overwhelming majority of unrated items.

In the experiments, we assume that *AP* is semi-trusted. *AP* fulfills private P2P binary PPCF scheme requirements, but she exploits inherent weaknesses in the protocol. In the first two experiments, we test how well the reconstruction attacks perform against *AP*'s privacy measures. In this context, privacy parameters of *AP*, which are δ_{AP}, the filling methods and *M* (number of groups) are examined. The *AP* fills her rating vector up to δ_{AP} density, which is associated to *AP*'s density. The filling methods, DV and RF, determine how selected unrated items will be filled. DV fills with default ratings while RF fills with random ratings. *M*, determines how many groups the *AP* will create for the rating vector while applying RRT. In third experiment, we include θ parameter. Although it is randomly determined for each peer, studies in the literature [19, 20] use constant θ values. We will examine the effects of random and constant θ values on the reconstruction. When applied alone, *AP*'s privacy measures provide a private protocol; however, the *AP* can act maliciously. Authors [6] claim that some extra measures can be adapted to preserve collaborating peers' privacy. The fourth experiment examines PPR, where collaborating peers hide their rating vectors using HRI method like the *AP*. In this experiment, δ_{peer} values of participating peers are varied to analyze how an increasing number of introduced fake ratings in peers' vectors affect the reconstruction results. The fifth experiment includes PPP, as well. In the original scenario, peers who rate *q* can join the prediction process. The *AP* can disclose who rate *q* by checking participating peers. PPP lets half of the collaborating peers to participate regardless of whether they rate *q* or not. In this experiment, PPP and PPR are applied both individually and together with *AP*'s privacy measures to test their effectiveness compared to an experimental reference setting, where only *AP*'s privacy measures are applied.

Table 1. The confusion matrix

		Original		
		Likes	**Dislikes**	**Unrated**
Classified	**Likes**	V_{11}	V_{12}	V_{13}
	Dislikes	V_{21}	V_{22}	V_{23}
	Unrated	V_{31}	V_{32}	V_{33}

$$prec = \frac{\sum_{i=1}^{2} V_{ii}}{\sum_{i=1}^{2}\sum_{j}^{3} V_{ij}} \tag{2}$$

Attacks in this paper must be operated in a repeated manner to build a matrix of collaborating peers' vector. Therefore, a random q among selected eligible movies is selected every time *acting as an active user* and *alienate the victim* attacks perform, and these attacks are repeated in m times. On the other hand, *perfect match* attack can reveal that an item's rating is either unrated or its value relative to q. In *perfect match* attack, each item needs to be queried to disclose which items are rated, the second aspect of privacy. After mapping of rated and unrated items is revealed, a second run is performed for m times with the selected movies as done in *acting as an active user* and *alienate the victim* attack. In order to derive the original rating matrix of peers, we introduce a new peer into the network. The introduced peer generates a query vector and asks for a prediction to derive peers' ratings in a repeated manner. Experiments have been run five times, and their average is evaluated.

5.3 Experimental Results

Effects of δ_{AP} and the filling method

In this experiment, one of AP's privacy measures, HRI protocol, is examined by varying the HRI parameters. Recall that *AP* applies the HRI protocol to hide her original ratings. The primary parameters in HRI are δ_{AP} and the filling methods, DV and RF. Reconstruction attacks have been performed with varying δ_{AP} values and two filling methods, DV and RF. For comparison purposes, we also included a plain P2P NBC-based CF with no privacy (No Filling in **Table 2**). In our experimental setting, δ_{AP} changes between $0.125d$ and $1d$, where d is the density of *AP*'s rating vector. When δ_{AP} goes beyond $1d$, *AP*'s rating vector would contain more fake ratings than authentic ones; therefore, it is the upper limit in the experiments. M, which is the *AP*'s privacy parameter for RRT, is set to 1 to eliminate any unexpected consequences that might result from RRT. Similarly, collaborating peers' privacy protocols, PPP and PPR, are not activated because HRI is analyzed during this experiment.

δ_{AP} is a factor determining how many unrated items will be filled by *AP*. Before giving the experimental results, we will first discuss the potential effects of varying δ_{AP} values concerning the reconstruction. Reconstruction results for *acting as an active user* attack could be affected by increasing δ_{AP} values regardless of which filling method is utilized. This attack relies on subsequent queries that differ by only one item rating at a time. When δ_{AP} and a filling method are utilized for a query, the next queries will be much more different from the intended one that is expected to differ only one item rating. Therefore, our intuition is that altering each departing query for participating peers by δ_{AP} and the filling method will diminish the results. The resilience of *alienate the victim* attack against privacy measures, δ_{AP}, and the filling method, could depend on the filling method that is used. This attack relies on singling out the victim item's rating from the rest. In DV, the dominant rating value in the vector is appended into unrated item cells as fake ratings. Such a way of filling unrated item cells does not affect the alienated status of the victim item. Therefore, we do expect that increasing δ_{AP} with DV would not make a prominent effect on the reconstruction result. RF method randomly fills unrated item cells. On the contrary, this breaks the alienated status of the victim item. Therefore, a declining trend is expected for growing δ_{AP} with RF. Remember that *perfect match* attack dispatches a query and tracks peers who have a perfect match with the dispatched

query. Appending new ratings into *AP*'s rating vector does not alter the basic principle of the attack. *AP* continues to monitor perfect matches in an unaffected way because this protocol only modifies *AP*'s rating and *AP* knows that no data is hiding at peers' sites. Appended ratings into a vector could either create a new perfect match or spoil a current perfect match. Therefore, the number of captured perfect matches might stay similar in size. As a result, similar results are expected as δ_{AP} grows with RF and DV. **Table 2** displays the results.

When any filling method is not employed (No Filling) for MLM, *prec* is 0.831, 0. 842 and 0.915 for *acting as an active user*, *alienate the victim* and *perfect match* and attacks, respectively. For YahooMovie data sets, the results vary between 0.906 and 0.937. Even filling methods are not used, where no privacy is applied, the reconstruction attacks in the paper need the value of *q* for the reconstruction. *AP* does not know the rating of *q* in peers' vectors and utilizes auxiliary information to speculate the value of *q* at peers' site; therefore, this experimental result is important to stress that exploiting auxiliary information could be very useful to reconstruct with decent *prec*. When privacy is considered, **Table 2** clearly shows that increasing δ_{AP} values have a dramatic effect on *acting as an active user* attack. There is a sharp decrease in *prec* for both data sets as soon as privacy measures are introduced, $\delta_{AP} = 0.125d$. As δ_{AP} is increased up to $1d$, the decline continues for DV and RF. Results for *acting as an active user* attack are in accordance with our expectation stated previously. *Alienate the victim* attack displays a constant trend in terms of *prec* for growing δ_{AP} with DV. As initially stated, when DV is utilized as the filling method, increasing δ_{AP} does not affect this attack since it has no particular damage in the alienated status of the victim item. Therefore, results for DV with larger δ_{AP} confirms our argument. However, when RF is utilized to fill unrated items' cells, a steep decline is recorded as δ_{AP} gets larger. *Perfect match* attack maintains a stable record in terms of both metrics. Especially, when DV is applied, *prec* almost remains defiant for larger δ_{AP} values. When RF is utilized, *prec* is relatively lower since inserted default ratings of a peer in DV method are intuitionally more inclined to be in harmony with peers' ratings. As a result, increasing δ_{AP} does not hinder *AP* from discovering perfect matches regardless of filling methods; therefore, the evaluation metric follows a more constant trend for DV and RF.

Table 2. Effects of varying δ_{AP} and filling methods

		Acting as an active user		Alienate the victim		Perfect match	
Filling Method	δ_{AP}	**MLM**	**YahooMovie**	**MLM**	**YahooMovie**	**MLM**	**YahooMovie**
No Filling	$0d$	0.831	0.906	0.842	0.908	0.915	0.937
DV	$0.125d$	0.107	0.336	0.843	0.905	0.914	0.943
	$0.25d$	0.092	0.280	0.844	0.905	0.913	0.940
	$0.5d$	0.077	0.212	0.842	0.906	0.913	0.944
	$1d$	0.063	0.141	0.844	0.906	0.913	0.940
RF	$0.125d$	0.117	0.356	0.374	0.749	0.913	0.943
	$0.25d$	0.101	0.323	0.250	0.622	0.904	0.931
	$0.5d$	0.085	0.261	0.158	0.462	0.897	0.937
	$1d$	0.071	0.199	0.100	0.305	0.891	0.935

Effects of varying *M*

In addition to the data hiding by HRI, *AP* masks the rating vector by utilizing RRT method as well. In RRT, *AP* uniformly generates a random group number for each peer independently, M_i,

so that whole rating vector is not revealed if an item is disclosed. As M_i grows for a peer, the rating vector is split more**.** In this experiment, M_i is varied between 1 and 20 to test how splitting the rating vector affects the reconstruction results. δ_{AP} is set to 0.25d, and the filling method is DV. Since *AP*'s privacy is handled, PPP and PPR (collaborating peers' privacy) are not activated.

In *acting as an active user* attack, we hypothesize that growing M_i values could introduce some improvement on the reconstruction. It is important for this attack to be successful that subsequent queries are only different by one item because temporal changes are monitored between subsequent queries. When M_i gets larger, the size of each group shrinks. As a result, the possibility of appending a fake rating into the group that contains the manipulated item decreases. Therefore, the group containing the manipulated item might remain unaltered after HRI, and it would help the reconstruction accuracy. In terms of *alienate the victim* attack, we anticipate that it remains unaffected from larger group sizes if DV is exploited as the filling method. This attack performs well as long as the victim item remains isolated from the rest and increasing M_i does not affect the victim item to take away its isolation status. Thus, a steady trend is expected. In terms of *perfect match* attack, introducing larger groups, principally, should have no adverse effect on the basic principle of the attack. Perfect matches can still be matched.

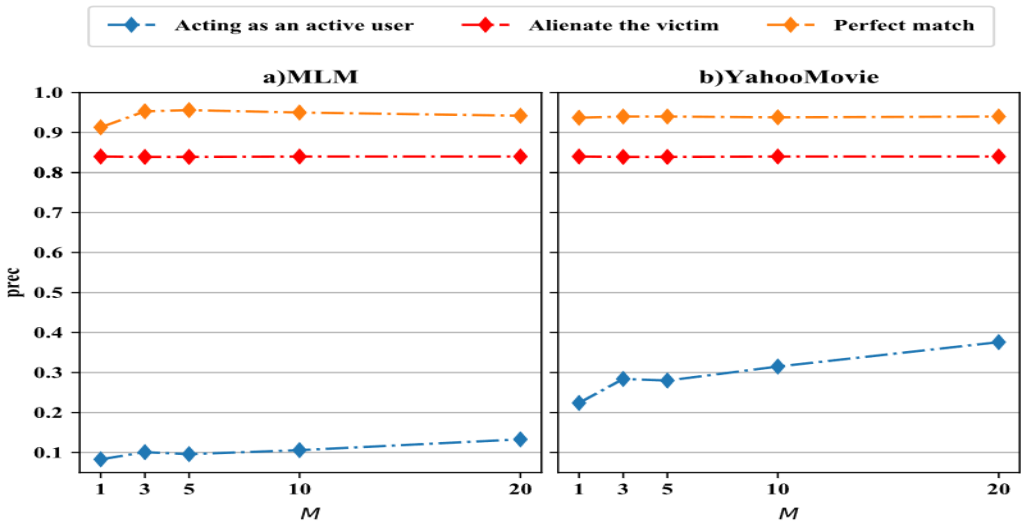

Fig. 5. Effects of varying M_i

Fig. 5 displays *prec* results for MLM and YahooMovie data sets. For both data sets, an increasing trend is recorded for *acting as an active user* attack although it is more prominent for YahooMovie. This increase in the results approves our intuition about larger groups. A larger group number means fewer items for each group; therefore, the manipulated item's group could remain same after HRI which leads to increase in *prec* results. Concerning the second attack type, *alienate the victim*, increasing M_i does not affect the victim item's probability results returned from other peers. Therefore, *prec* metric follows a constant trend for all values of M_i as we initially hypothesized. Similarly, *prec* metric for *perfect match* attack remains stable. The minor exception is the increase in between one-group and three-groups for MLM data set. Remaining groups larger than 3 continue a relatively steady trend for *prec*. Such an outcome also confirms our intuition about larger groups for *perfect match* attack.

When attacks are compared, although *acting as an active user* attack performs an upward trend for larger groups, its *prec* is very low when compared with our proposed attacks, *alienate the victim* and *perfect match*.

Effects of θ, constant or random

In [6], the *AP* utilizes RRT to reverse or keep original ratings by dividing her rating vector into up to *M*, which is analyzed in the previous experiment. In this experiment, we analyze how our attacks performs against different values of θ. Although the authors propose to use random θ values for each group and peer (θ_{ik} in Section 3), we compare it with constant values of θ varying from 0.51 to 0.90. Other experimental settings are set as follows: δ_{AP} is 0.25*d*, M_i is 5, the filling method is DV and PPP and PPR are not activated.

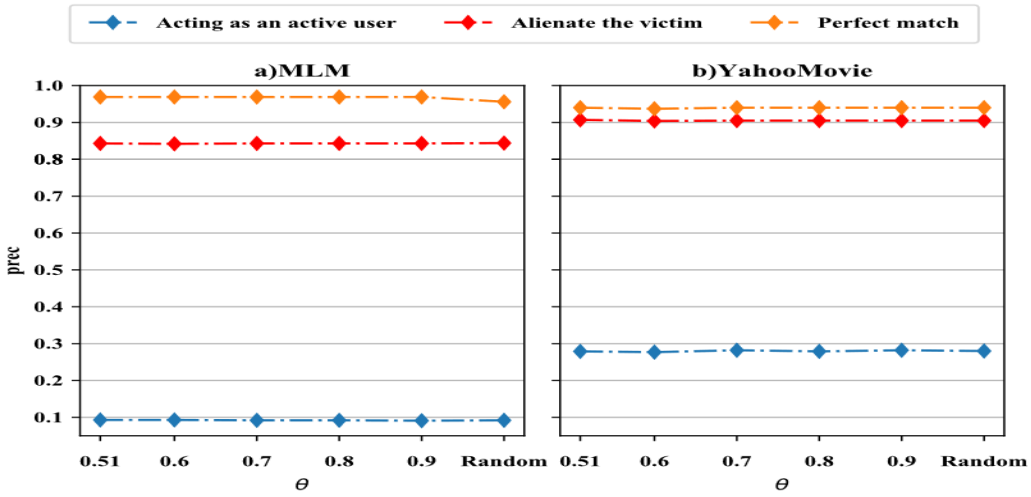

Fig. 6. Effects of varying constant and random θ

θ is associated with the *AP*. The *AP* either reverses or preserves her rating vector by comparing randomly generated r_{ik} value with θ and sends her masked rating vector to the related peer. The *AP* can easily determine whether the incoming value from a peer is calculated by using a reversed or original rating vector. If the *AP* finds out it is computed for a reversed rating, she can easily convert it what it had to be. As a result, using random or constant values of θ does not prevent the *AP* from discovering partial conditional probability values. It is very crucial for the attacks in this paper because they rely on obtaining true partial conditional probability values. Since random or different constant θ values does not perturb the partial similarities received from peers, we anticipate that our attacks would perform similarly in terms of reconstruction accuracy. The **Fig. 6**, displays the experimental results.

Fig. 6 confirms our intuition that varying constant or random values of θ does not affect the reconstruction of the attacks in this paper. The figure follows a constant trend for larger values of constant θ and introducing random θ value does not alter this trend. A malicious *AP* should select random or values around 0.51 as θ because such values offer more privacy for her and do not alter her accuracy on reconstruction.

Effects of PPR parameter, δ_{peer}

Up to now, privacy is always viewed from *AP*'s point of view. However, peers can also apply some privacy measures as discussed in Section 3 to prevent possible data disclosure. In this

experiment, we evaluate the performance of PPR against the reconstruction attacks. Partial conditional values calculated by peers are perturbed due to appended fake ratings up to δ_{peer} density in PPR. We associate δ_{peer} with vector density similar to HRI used by the *AP*. Therefore, δ_{peer} is varied between 0.125*d*, 0.25*d*, 0.5*d* and 1*d* in this experiment. Other parameters are set as follows: δ_{AP} is 0.25*d*, M_i is 5, the filling method is DV and PPP is not activated. Results are given in **Fig. 7**.

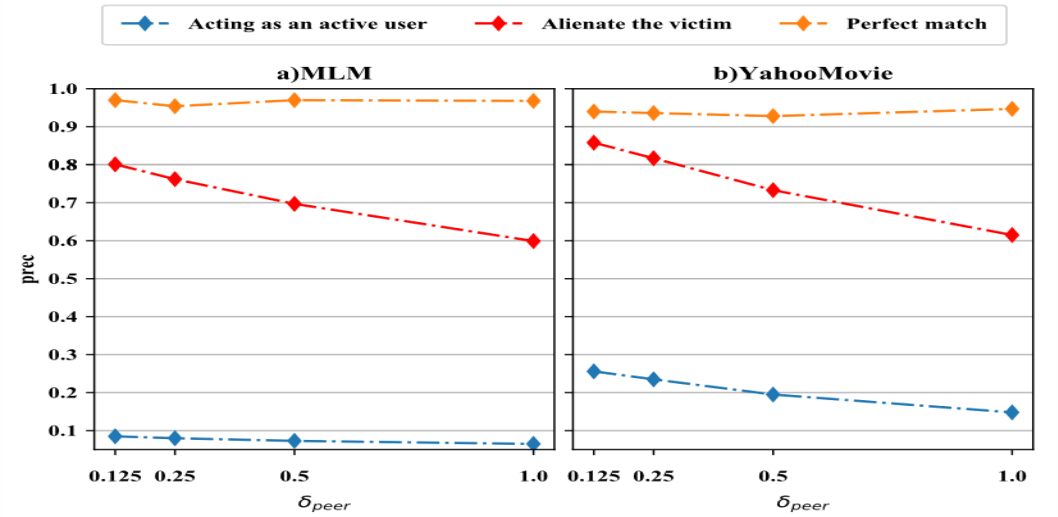

Fig. 7. Effects of varying δ_{peer}, PPR

Increasing values of δ_{peer} add randomness to collaborating peers' rating vectors; therefore, it diminishes the partial conditional probability values transferred to the malicious *AP*. *Acting as an active user* tracks temporal changes in those results in subsequent queries. PPR (δ_{peer}) causes inconsistencies in subsequent transferred values. *Alienate the victim* attack relies on similarity values returned from collaborating peers for the victim item. However, the corresponding item rating at peers' vectors might be appended due to PPR, and such a coincidence confuses the *AP*. The *AP* might derive a rating for the unrated item due to PPR. Since these two attacks depend on the partial probability values, there is a clear downward trend for the larger values of δ_{peer} in **Fig. 7**. In terms of *perfect match* attack, **Fig. 7** displays a rather stable trend like the first experiment, where δ_{AP} is tested. Remember that the *AP* repeatedly looks for perfect matches in different queries. One appended rating in a peer-vector can be detected in other perfect matches so they can be marked as unrated. This attack is, therefore, successful in determining appended ratings due to PPR. It is also clear in **Fig. 7** that larger δ_{peer} values do not negatively affect the reconstruction results for perfect match attack.

Effects of peer privacy, PPP and PPR

In addition to PPR, collaborating peers apply PPP for privacy protection, as well. Recall that PPP regulates the participation in a prediction process by a coin toss so that *AP* cannot reveal who rate *q*. On the other hand, PPR is a protocol that lets each collaborating peer apply HRI to perturb the reciprocal similarity (conditional probability). The effects of integrating peers' privacy are examined in this experiment. Therefore, we activate PPP and PPR. PPR fills unrated items up to a certain density (δ_{peer}) like HRI, and δ_{peer} is set to 0.25*d*. *AP*'s privacy parameters, which are δ_{AP}, M_i, and the filling method, are set to 0.25*d*, 5 and the filling method,

respectively. **Fig. 8** displays a bar graph of the results. In **Fig. 8**, PPP or PPR shows that only PPP or PPR is applied, respectively. PPP&PPR shows that both of PPP and PPR are applied. Reference column displays a default experimental setting ($\delta_{AP} = 0.25d$, $M_i = 5$ and the filling method are DV) without collaborating peers' privacy considered for comparison purposes.

When PPR is considered, half of the peers participating in the prediction process including peers who did not rate *q*, if a non-rater peer joins the prediction, *q* is filled with the default vote of the peer, and this adds another uncertainty to the reconstruction process. *AP* assumes that the rating of *q* would be correlated to the average rating collected from IMDB or GNPP (Section 4.4). Furthermore, PPR lets peers mask their ratings by utilizing HRI. PPR turns the original peers' vectors into another one by appending fake ratings. Due to uncertainties introduced by PPP and PPR, we anticipate that both PPP and PPR will negatively affect the attacks regarding *prec* results.

In terms of PPP, the most noticeable point in **Fig. 8** for both data sets is the dramatic declines recorded for *perfect match* attack compared with the other two attacks. In *perfect match* attack, *AP* first marks rated items based on peers' participation by dispatching queries with different *q*. Then, the attacker performs a second run to reveal actual rating values. Random peers join the prediction in PPP, and peers who did not rate *q* yet joining the prediction use default value instead. Therefore, an attacker cannot form a true mapping for rated items, and it leads a dramatic decline in *prec* when PPR is integrated for *perfect match* attack. Compared to other two attacks, the relatively greater decline of *prec* in *perfect match attack* could be attributed owing to the larger item set that this attack deals with each time. *AP* marks all items in its rating vector based on *q* in *perfect match attack*; however, the first two attacks deal with only one item, the manipulated or victim item, at each iteration of the attack. In *acting as an active* user attack, *AP* exploits changes in peers' probability values for subsequent queries. Since participating peers constantly change due to PPP, some peers' probabilities that *AP* is monitoring to exploit in the next query might not be captured, which might cause the decline. In terms of *alienate the victim* attack, it is more resistant to PPP for both data sets. Primarily, it slightly achieves to outperform the reference setting with YahooMovie data set. This attack could be successful unless the isolated status of the victim item is not broken. PPP does not explicitly offer such protection; therefore, *prec* remains relatively stable.

The main reason that makes sense about the decline in the results for all attack types when PPR is applied is that original data held by peers is masked. In *alienate the victim* attack, the conditional probability calculated by peers for the victim item could be affected by appended fake ratings due to PPR. The corresponding victim item might be unrated in an original peer-vector; however, it might have been filled due to PPR protocol. Thus, the related conditional probability returned by the peer misleads the attacker in *alienate the victim* attack. We believe the significant decline compared to the reference and PPP case in *alienate the victim* attack is due to change in the conditional probability calculation. *Perfect match* attack seems to be resilient to PPR when compared to the reference setting and PPP case. Appending fake ratings by collaborating peers do not affect the basic idea of this attack, perfect matches can still be captured. Like the first experiment, there are two cases in PPR scenario affecting the number of perfect matches. The first is that a possible perfect match could be lost due to appended ratings. In another case, a new perfect match can be captured. Because some perfect matches are lost, and some are gained, we believe that perfect match attack is resilient to PPR owing to this factor. *Acting as an active user* attack performs a decline for both data sets due to HRI protocol applied by peers. Due to PPR, peers fill random unrated items each time a new query is received. Such a change in peers' vector when a new query received violates the

underlying assumption of *acting as an active user* attack. Change in the peer-vector means that the manipulated query by one rating is compared with a different peer vector because of the appended ratings. This attack relies on the change in conditional probability values in subsequent queries which differ in only one rating.

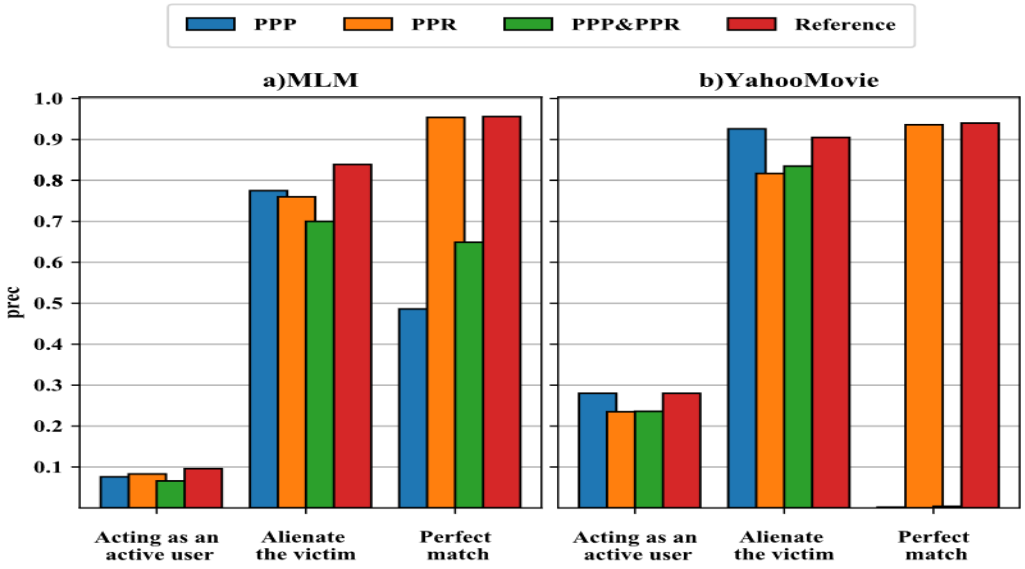

Fig. 8. Effects of peer privacy measures

6. Conclusion

In this study, we scrutinize privacy promises made by an NBC-based P2P PPCF scheme [6]. Three different attack techniques have been tested with different privacy protection parameters that the targeted scheme offers. Although a malicious *AP* does not know the rating of q held by peers, which is crucial to derive original ratings, we overcome this problem by utilizing auxiliary information. This scheme handles privacy in both *AP*'s and peers' perspective. Both *AP*'s and peers' privacy measures are examined in the experiments. Our experimental results show that *acting as an active user* attack is not successful for increasing δ_{AP} values. While *alienate the victim* attack, proposed in this paper, is not affected by larger δ_{AP} values if the filling method is default voting. However, this attack records a declining trend similar to *acting as an active user* attack when the filling method is random filling. *Perfect match* attack presents almost a stable trend for increasing δ_{AP} for both filling methods. When the number of groups is increased, *acting as an active* user attack performs some improvements; however, its precision results are already too low to be considered applicable. On the other hand, *alienate the victim* attack performs a stable trend in terms of precision. *Perfect match* attack is very resilient to protect its precision value for larger groups. It is also discussed that θ parameter of the randomized response does not affect the reconstruction accuracy for the attacks. In terms of peers' privacy, which is primarily designed to prevent from *acting as an active user* attack, integrating PPR with growing δ_{peers} leads to a clear decrease for *acting as an active user* and *alienate the victim* attacks while *perfect match* attack remains stable. When both PPP and PPR are applied, *alienate the victim* attacks seems to be auspicious results while the other two attacks dramatically decline in terms of precision. In

terms of PPR, *perfect match attack* displays very similar results to the reference setting where no peer privacy is utilized.

To sum up, privacy measures taken by targeted PPCF scheme [6] is designed to avoid from *acting as an active user* attack and the claimed-privacy is successful against this attack. However, *alienate the victim* attack can reconstruct with very high precision unless the random filling is utilized. *Perfect match* attack also reconstructs with very high precision unless peers decide to protect themselves by PPP. However, utilizing PPP will harm prediction results dramatically because it lets peers take part in the prediction without rating *q*. The attacks described in this paper cannot be generalized to all privacy promises made for binary rated data. The reason is that they are designed to monitor similarity values exchanged by peers or parties while producing predictions. However, if similarities are exchanged, the idea in *perfect match* attack reveals that the corresponding rating vectors are alike, and an attacker might exploit such a disclosure. In such a case, one should consider the details of the protocol to adapt *perfect match* attack.

Similarly, an attacker in *acting as an active user* attack might alter one cell rating at a time and monitor similarity values to reveal the altered rating. Therefore, these two attacks might be generalized if parties exchange similarities in a collaborating filtering protocol. *Alienate the victim* attack, on the other hand, utilize drawbacks of the targeted scheme [6] where the related conditional probability is calculated only for the victim item, which is singled out by its rating.

Since *alienate the victim,* and *perfect match* attacks are very resilient under different privacy settings, we believe extra measures should be developed. This study also confirms that exploiting auxiliary information could be very crucial to infer confidential data. We plan to investigate NBC-based P2P PPCF scheme [6] in detail to take it one step further in terms of attack types given in this study as our future motivation.

References

[1] John Canny, "Collaborative filtering with privacy," in *Proc. of IEEE Symposium on Security and Privacy*, pp. 45-57, May 12-15, 2002. Article (CrossRef Link).

[2] Lorrie F. Cranor, "I did not buy it for myself," *Human-Computer Interaction Series*, vol 5, pp. 57-73, 2004. Article (CrossRef Link).

[3] Huseyin Polat and Wenliang Du, "Privacy-preserving top-*N* recommendation on distributed data," *Journal of the Association for Information Science and Technology*, vol. 59, no. 7, pp.1093-1108, 2008. Article (CrossRef Link).

[4] Cihan Kaleli and Huseyin Polat, "Providing naïve Bayesian classifier-based private recommendations on partitioned data," *Lecture Notes in Computer Science*, vol. 4702, pp. 515-522, 2007. Article (CrossRef Link).

[5] Cihan Kaleli and Huseyin Polat, "Privacy-Preserving Naïve Bayesian Classifier – Based Recommendations on Distributed Data," *Computational Intelligence*, vol. 31, no. 1, pp. 47-68, 2015. Article (CrossRef Link).

[6] Cihan Kaleli and Huseyin Polat, "P2P collaborative filtering with privacy," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 18, no. 1, pp. 101-116, 2010. Article (CrossRef Link).

[7] Stanley L Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp.63-69, 1965. Article (CrossRef Link).

[8] Sheng Zhang, James Ford and Fillia Makedon, "Deriving private information from randomly perturbed ratings," in *Proc. of the 2006 SIAM International Conference on Data Mining*, pp. 59-69, April 20-22, 2006. Article (CrossRef Link).

[9] Murat Okkalioglu, Mehmet Koc and Huseyin Polat, "On the discovery of fake binary ratings," in *Proc. of the 30th Annual ACM Symposium on Applied Computing*, pp. 901-907, April 13-17, 2015. Article (CrossRef Link).
[10] Burcu Demirelli Okkalioglu, Mehmet Koc and Huseyin Polat, "Reconstructing rated items from perturbed data," *Neurocomputing*, vol. 207, pp.374-386, 2016. Article (CrossRef Link)
[11] Burcu Demirelli Okkalioglu, Mehmet Koc and Huseyin Polat, "Deriving private data in partitioned data-based privacy-preserving collaborative filtering," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 32, no. 1, pp. 53-64, 2017. Article (CrossRef Link).
[12] Murat Okkalioglu, Mehmet Koc and Huseyin Polat, "On the privacy of horizontally partitioned binary data-based privacy-preserving collaborative filtering," *Lecture Notes in Computer Science*, vol. 9481, pp. 199-214, 2015. Article (CrossRef Link).
[13] Murat Okkalioglu, Mehmet Koc and Huseyin Polat, "A privacy review of vertically partitioned data-based PPCF schemes," *International Journal of Information Security Science*, vol. 5, no. 3 pp. 51-68, 2016.
[14] Huseyin Polat and Wenliang Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *Proc. of Third IEEE International Conference on Data Mining, ICDM 2003*, pp. 625-628, Nov. 22, 2003. Article (CrossRef Link).
[15] Nikolas Polatidis, Christos K. Georgiadis, Elias Pimenidis and Haralambos Mouratidis, "Privacy-preserving collaborative recommendations based on random perturbations," *Expert Systems with Applications*, vol. 71, pp. 18-25, 2017. Article (CrossRef Link).
[16] Mengmeng Yang, Tianqing Zhu, Yang Xiang and Wanlei Zhou, "Personalized privacy preserving collaborative filtering," *Lecture Notes in Computer Science*, vol. 10232, pp. 371-385, 2017. Article (CrossRef Link).
[17] Mengmeng Yang, Tianqing Zhu, Lichuan Ma, Yang Xiang and Wanlei Zhou, "Privacy preserving collaborative filtering via the Jonhson-Lindenstrauss transform," in *Proc. of 2017 IEEE Trustcom/BigDataSE/ICESS*, pp. 417-424, Aug. 1-4, 2017. Article (CrossRef Link).
[18] Ping Xiong, Lefeng Zhang, Tinaqing Zhu, Gang Li and Wanlei Zhou, "Private collaborative filtering under untrusted recommender server," *Future Generation Computer Systems, in press*, 2018. Article (CrossRef Link).
[19] Huseyin Polat and Wenliang Du, "Achieving private recommendations using randomized response techniques," *Lecture Notes in Computer Science*, vol. 3918, pp. 637-646, 2006. Article (CrossRef Link).
[20] Cihan Kaleli and Huseyin Polat, "Providing private recommendations using naive Bayesian classifier," *Advances in Soft Computing*, vol. 43, pp. 168-173, 2007. Article (CrossRef Link).
[21] Huseyin Polat and Wenliang Du, "Privacy-preserving top-n recommendation on horizontally partitioned data," in *Proc. of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 725-731, Sept. 19-22, 2005. Article (CrossRef Link).
[22] Alper Bilge, Cihan Kaleli, Ibrahim Yakut, Ihsan Gunes and Huseyin Polat, "A survey of privacy-preserving collaborative filtering schemes," *International Journal of Software Engineering and Knowledge Engineering*, vol. 23, no. 08, pp. 1085-1108, 2013. Article (CrossRef Link).
[23] Rakesh Agrawal and Ramakrishnan Srikant, "Privacy-preserving data mining," in *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, vol. 29, no. 2, pp. 439-450, May 15 - 18, 2000. Article (CrossRef Link).
[24] Dakshi Agrawal and Charu C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proc. of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 247-255, 2001. Article (CrossRef Link).
[25] Hillol Kargupta, Souptik Datta, Qi Wang and Krishnamoorthy Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Proc. Third IEEE International Conference on Data Mining*, pp. 99-106, Nov. 22, 2003. Article (CrossRef Link).
[26] Hillol Kargupta, Souptik Datta, Qi Wang and Krishnamoorthy Sivakumar, "Random-data perturbation techniques and privacy-preserving data mining," *Knowledge and Information Systems*, vol. 7, no. 4, pp. 387-414, 2005. Article (CrossRef Link).

[27] Zhengli Huang, Wenliang Du and Biao Chen, "Deriving private information from randomized data," in *Proc. of the 2005 ACM SIGMOD international conference on Management of Data*, pp. 37-48, June 13 - 17, 2005. Article (CrossRef Link).
[28] Songtao Guo, Xintao Wu and Yingjiu Li, "Determining error bounds for spectral filtering based reconstruction methods in privacy preserving data mining," *Knowledge and Information Systems*, vol. 17, no. 2, pp. 217-240, 2008. Article (CrossRef Link).
[29] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov, ""You Might Also Like:" Privacy Risks of Collaborative Filtering," in *Proc. of 2011 IEEE Symposium on Security and Privacy*, pp. 231-246, 2011. Article (CrossRef Link).
[30] Burcu Demirelli Okkalioglu, Murat Okkalioglu, Mehmet Koc and Huseyin Polat, "A survey: deriving private information from perturbed data," *Artificial Intelligence Review*, vol. 44, no. 4 pp. 547-569, 2015. Aticle (CrossRef Link).
[31] Koji Miyahara and Michael Pazzani, "Collaborative filtering with the simple Bayesian classifier," *Lecture Notes in Computer Science*, vol. 1886, pp. 679-689, 2000. Article (CrossRef Link).
[32] Yahoo!, "Yahoo! movies user ratings and descriptive content information, version 1.0,". URL: https://webscope.sandbox.yahoo.com/

Murat Okkalioglu received his BSc degree from Computer Engineering Department at Pamukkale University, Denizli, Turkey, MSc degree from The University of Texas at San Antonio, Texas, The United States, and PhD degree from Anadolu University, Eskisehir, Turkey in 2008, 2012 and 2017, respectively. Currently, he is an Assistant Professor in Computer Engineering Department at Yalova University, Turkey. His research interest is privacy in recommender systems.

Cihan Kaleli is an Associate Professor in Computer Engineering Department at Eskisehir Technical University, Turkey. He got his Master's degree and PhD from Computer Engineering Department of Anadolu University in 2008 and 2012, respectively. His research interest is privacy-preserving data mining in general; and specifically, studies distributed data-based collaborative filtering with privacy.