On the Discovery of Fake Binary Ratings

Murat Okkalioglu
Yalova University
Computer Engineering Department
77100 Yalova, Turkey
+90 226 815 5000
murat.okkalioglu@yalova.edu.tr

Mehmet Koc
Bilecik Seyh Edebali University
Electrical & Electronics Engineering Department, 11210 Bilecik, Turkey
+90 228 214 1111
mehmet.koc@bilecik.edu.tr

Huseyin Polat
Anadolu University
Computer Engineering Department
26470 Eskisehir, Turkey
+90 222 321 3550
polath@anadolu.edu.tr

ABSTRACT
Privacy-preserving collaborative filtering methods promise to preserve privacy of individuals. In general, privacy has two aspects, preserving the rating values of users and masking who rated which items. In this study, we analyze a privacy-preserving collaborative filtering method for binary data referred to as randomized response technique. We develop a method targeting the second aspect of privacy to discover fake binary ratings using auxiliary and public information.

Categories and Subject Descriptors
• Security and privacy: Human and societal aspects of security and privacy: Privacy protections

• Information systems: Information systems applications: Data mining: Collaborative filtering

General Terms
Security, Algorithms, Experimentation.

Keywords
Privacy analysis, binary data, auxiliary information, fake ratings

1. INTRODUCTION
The Internet and information technologies have great effects onto our habits. Many traditional ways of people's daily routines from basic social activities such as gathering together for a cup of coffee and conversation to going out for shopping have moved to online channels. The convenience of online activities has been indispensable for people of this age, especially for today's teenagers growing up with technology. This natural dependency of people to the Internet has caused huge amount of data to be created every second. This phenomenon has caused a new term, called *information overload*, to emerge. People have to deal with great amount of information to make a decision. At this point, collaborative filtering (CF) techniques are popular approaches help people make a decision by producing recommendations. In a traditional setting, a CF system is composed of n users and m items consisting a matrix of $n \times m$. Such a matrix is usually very sparse because there are many items to rate and users only rate for the items based on their interests.

SAC'15, April 13-17 2015, Salamanca, Spain

http://dx.doi.org/10.1145/2695664.2695866

Accuracy of a CF system relies on a considerable participation of its users. However, CF systems have some disadvantages including individual privacy [1, 2]. Individual user data is considered valuable asset and it can be sold in case of bankruptcy. Additionally, user data can be processed by data owners and is subject to various threats such as unsolicited marketing, price discrimination, information inference, government surveillance, unauthorized access, and so on [3]. Therefore, users might be unwilling to share their true preferences unless they are convinced that their data is protected.

Users would be more eager to participate in a CF process if they believe that their privacy is preserved. Privacy-preserving collaborative filtering (PPCF) addresses the problem of preserving individual privacy while still providing accurate predictions. Conceptually, privacy and accuracy are conflicting goals and the dominating challenge for PPCF systems is to balance an equilibrium between them. User input could vary such as numeric, categorical, or binary. These data types measure users' attitude in different scales. For example, numeric metrics express how much a particular item is liked by a user while binary metrics show if an item is liked by a user or not. Different perturbation schemes are available such as randomized perturbation technique [4, 5] and randomized response technique (RRT) [6]. For more details about PPCF schemes, one can refer to the study presented by Bilge et al. [7]. Perturbation can also be reinforced by random filling of empty cells to masks which items are rated.

Our primary focus for this study is the RRT proposed by Polat and Du [6]. Their method perturbs user data by employing randomized response method by Warner [8]. However, perturbing user data is not enough for individual privacy because the server side can still observe which users rate which items. Random filling is applied to mask unrated items. We have already studied how to derive original binary ratings from RRT [9]. In this paper, we analyze random filling method presented by Polat and Du [6] to discover which items are fake or rated by users. We exploit auxiliary and public information to show "*having reliable prior knowledge about data would cause privacy breaches*." Random filling procedure for RRT fills $\beta\%$ of unrated items with 1 or 0. The effects of perturbing constraints like number of groups (M) and perturbing scale (θ) and random filling constraint (β) will be discussed throughout the analysis and experiments.

The rest of this study is organized as follows. In the next section, related work is given. Preliminaries for RRT are introduced in Section 3 in detail. In Section 4, analysis about discovering fake ratings and determining actual ratings are discussed. Section 5 presents our method of discovering fake items in detail while Section 6 covers real data-based experiments performed to evaluate the success of the proposed schemes. Finally, comments, outcomes, and future research directions are discussed about the study in the last section.

2. RELATED WORK

Privacy analysis should present to what extent privacy is protected by a proposed perturbation scheme. In other words, the objective is to analyze how closely original values can be determined from the perturbed data. Various researchers present privacy measures to quantify privacy [10, 11, 12, 13]. Agrawal and Srikant [10] discuss *c%* confidence level, if an attribute can be estimated that it lies in the interval $[x_1, x_2]$ with *c%* confidence. To illustrate, if a uniform distribution is between [-1, 1], then the intervals of [0, 1] and [0, 1.5] are the amount of privacy at confidence level 50% and 75%, respectively. Confidence level quantifies privacy in terms of interval range of distortion added to original data. However, Agrawal and Aggarwal [11] argue that *c%* confidence level does not consider the distribution of data and they state that distribution of original data can reveal a certain level of information that can cause to guess some unintended information. They quantify privacy by considering distribution of original data. They propose privacy measure based on differential entropy (h) of random variable ($\Pi(X) = 2^{h(X)}$). Differential entropy is a measure of uncertainty inherent in random variable. In general, this metric measures the length of interval over which a uniformly random variable has the same uncertainty. Furthermore, average conditional privacy is introduced which is the privacy of X given Y, $\Pi(X) = 2^{h(X|Y)}$. This metric motivates the fraction of privacy lost given Y, $P(X|Y) = 1 - 2^{-I(X;Y)}$. As a result, X has a privacy $\Pi(X|Y) = \Pi(X)(1-P(X|Y))$ after Y is revealed.

Rivzi and Harista [12] propose another privacy measure evaluating binary data. They question to what extent 1 and 0 can be reconstructed. They derive a reconstruction equation targeting their distortion procedure by evaluating probability of correct reconstruction. After determining reconstruction probability (p) specifically for their distribution, the privacy metric that measures the reconstruction probability is proposed as Eq. (1) that can be applied any distortion procedure once reconstruction probability has been identified.

$$P = (1 - p) \times 100 \quad (1)$$

Evfimievski et al. [13] propose a methodology to limit privacy breaches for categorical data. Privacy breach is defined as revealing a particular property of private information holds with high probability with the disclosure of its randomized form. They state *ρ1-to-ρ2* privacy breach occurs if the disclosure of randomized data increases the probability of original attribute, where $\rho1$ is the probability of discovering original data while $\rho2$ is the probability of original attribute after revealing randomized data and $0 < \rho1 < \rho2 < 1$. Originating from the definition of privacy breach, they introduce amplification proving that disclosure of randomized data has limited effect at privacy breaches, depending on the value of γ. Their proposed measure is applicable for categorical data, where private information belongs to some finite set of V_x.

Huang and Du [14] study RRT to obtain optimal scheme by quantifying both privacy and utility. Their methodology tries to find optimal disguising matrix to perturb data. They quantify privacy and utility as an estimation problem. Privacy metric is about how accurate individual estimation of original values can be derived given perturbed values. On the other hand, utility is a metric to reconstruct original distribution of original data.

Auxiliary data is utilized by Calandrino et al. [15] to infer information about users of CF systems. They conduct experiments to argue the information inference resilience of online CF websites. Zhang et al. [16] propose reconstruction methods targeting numerically perturbed data by randomization utilized by Polat and Du [4]. They exploit singular value decomposition and *k*-means algorithm. Although the abovementioned work studies numerically rated data, our earlier work [9] targets binary perturbed data by RRT to derive original ratings by exploiting auxiliary and public information. In this study, we are interested in discovering fake binary ratings made by users to mask which items are rated.

3. PRELIMINARIES

There are two aspects of preserving privacy of individuals in general. The first one is to perturb the rated items to disguise real rating values. Perturbation masks the rated items in order to preserve how items are rated. However, preserving which items are rated by a specific user is also confidential. Thus, it is also prominent to mask which items are rated by users. In this section, we handle two aspects of privacy for RRT proposed by Polat and Du [6].

3.1 Randomized Response Technique

RRT for binary rated data proposed by Polat and Du [6] is based on Warner's study [8], which is originally designed for surveys. This technique is devised to encourage survey respondent to answer their true opinion about a sensitive question. A threshold value is determined between 0 and 1 and respondents use a random device and picks a number. If the random device generates a number less than predetermined value, then respondents answer the sensitive question. Otherwise, they respond a question whose answer is exact opposite of the sensitive question.

Inspiring from randomized response model, Polat and Du [6] apply a similar technique for binary rated data. Remember that each user of CF systems has a vector of ratings of different items composing a large and sparse matrix due to overwhelming unrated items. Each user picks a uniform random number, r_u, between 0 and 1. Then, r_u is checked against the threshold value, θ. If $r_u \leq \theta$, then the user preservers her rating vector. Otherwise, whole rating vector is reversed. Suppose that the user has a vector of (0 1 * 1 * * * 0 *), where * means specified item is not rated. If $r_u > \theta$, then reversed version of vector (1 0 * 0 * * * 1 *) is sent to the server.

3.1.1 Determining θ

RRT has two different versions in terms of determining threshold value, θ. In the first setting, the server and the users agree on a predetermined θ value. Each user utilizes predetermined θ before sending her own vector to the server. Instead of using a common predetermined value, each user can pick a θ_u between 0.51 and 1.

Note that complementary θ values between 0.0 and 0.49 have the same effects [6]. After determining θ_u, each user utilizes θ_u values independently to decide if item vector will be reversed or preserved while sending to the server.

Determining θ value has a direct effect on how much data is preserved in RRT. Recall that main assumption is to reserve a user vector if $r_u > \theta$. Therefore, θ value specifies how much data is reversed and it can be considered as a perturbation scale. As θ approaches to 1, it is likely that the user vector will be preserved due to $r_u > \theta$ condition to reverse ratings. This means that privacy level decreases. Since more data is likely to be preserved, better accuracy results are expected as θ moves away from 0.51 to 1. It is also possible to strengthen privacy by approaching θ to 0.51. Hence, when considering θ value, a decision must be made by considering both privacy and accuracy.

3.1.2 One and Multi Group Schemes

Other options are also possible in RRT to enhance privacy like dividing user input into one or multiple groups. In one group case, user vector is considered as whole and data reversing or preserving operations affects all vector items. An operation in one group case is performed on whole vector. The shortcoming of this approach is that the server can infer true ratings of all items for a user, if it obtains a true rating about any item because same operation is performed on all items. In multi group or M-group case, where $1 < M << m$, each user divides her rating vector into M groups. RRT is performed for each group by each user independently. If the server somehow gets a true rating about an item, it can only obtain true ratings of a group to which the related item belongs. Other groups remain safe. M group option provides more privacy when compared to one group case since each group is handled independently.

3.2 Random Filling

As discussed earlier, the other aspect of privacy is to protect rated items from being discovered. If users are convinced that items which they rated are not disclosed, they would be more willing to rate for sensitive items such as books with politics, gay rights, or adult contents.

Polat and Du [6] propose a random filling scheme to provide full privacy. Empty items are filled with random ratings to mask which items are actually voted by a user. To accomplish random filling, a β value is determined. Each user randomly fills $\beta\%$ of her empty cells with equal number fake ratings of 0 and 1.

β value can be set to provide flexibility of deciding how much fake ratings to insert into user vector. A maximum upper bound, β_{max}, is determined beforehand. Then, each user picks a uniform random β_u from the range $(0, \beta_{max}]$. Finally, $\beta_u\%$ of empty cells are filled with fake ratings. Since the server does not know β_u, it does not have any idea how much of items are filled with fake votes.

4. PRIVACY ANALYSIS

Privacy has two aspects applied by a PPCF scheme. The first one is to protect user data by distorting or perturbing original data so that the server cannot access the raw data and the second aspect is to hide which items are voted by the users. In this section, an analysis for these two aspects in RRT will be given. We start with discovering actual ratings and then we analyze the possibility of determining real values of rated items. Discovering actual ratings are first analyzed because determining real values would be needless without knowing which items are really rated.

4.1 Analysis of Discovering Rated Items

Assume that m is the number of items, m_t is the number of ratings while m_e is the number of empty cells before item set is filled. When β is constant, each user will exploit same value and determining how many actual ratings have made is quite simple, $m_t = (m_f - m\beta) / (1 - \beta)$, where m_f is number of items being filled after random filling. Another option to fill empty ratings is to select independent β_u for each user. According to RRT scheme, $\beta_u\%$ of m_e will be filled with uniform selection over $(0, \beta_{max}]$ and there are will be m_f items filled with m_e' empty cells ($m = m_t + m_e = m_f + m_e'$). Intuitively, the probability of selecting β_u is 1 out of β_{max}. However, we can narrow down this probability if we assume that the user has at least some number of actual ratings. In our case, let us assume that each user has at least two actual ratings. Then, there will be at most $m_e' - 2$ fake ratings that are inserted. If $m_e' - 2$ is greater than β_{max} percent of m_e, then no useful inference is made. However, if it is less than β_{max} percent of m_e, then the probability to select β_u can be narrowed down 1 out of $\lfloor((m_e' - 2)/m)100\rfloor$. After determining β_u, m_t can be calculated easily $m_t = m_f - \lfloor\beta_u m_e'/100\rfloor$. Upon guessing m_t, the remaining task is to identify which m_t items among m_f are actually voted. Here, the probability guessing correct items is 1 out of $C_{m_t}^{m_f}$.

Another scheme to fill unrated cells could be density based as a variation of a method for partitioned data applied by Yakut and Polat [17]. Instead of filling unrated cells with half of 1s and 0s for binary data, the user can prefer to keep like/dislike ratio. In this case, the user selects a uniform β_u over the range $(0, \beta_{max}]$ and there are m items. Once the server receives the disguised rating vector, it can infer like/dislike ratio. Assume that like/dislike ratio is $k > 0$ and m_1, m_0, and m_d are the number of likes, dislikes and sum of them in the disguised vector, respectively. We assume that at least one authentic vote have been made for both m_1 and m_0. Since the server knows m_1 and m_0, it is easy to observe that $m_1/m_0 = k$ and the server can determine the smaller one. Determining smaller value is important because we assume that at least one authentic vote has been made for it. This enables us to know at least $1+k$ actual votes have been made and at most $m_f = m_d - (1+k)$ items are filled where m_f is the number of filled items. A maximum boundary can be derived if m_f is less than $\beta_{max} \times m_d$, then an upper bound for probability of β_u can be set at $\lfloor(m_f/m_d)100\rfloor$. Otherwise, probability of β_u is 1 out of β_{max} similar to previous paragraph. After determining β_u, the previous calculation about combination can be repeated.

4.2 Analysis of Determining Real Ratings

After the first step of trying to decide which items are actually voted by users, the second step is to determine the real values of items being voted. Privacy levels provided by these schemes should be analyzed by controlling how it changes for different perturbation values.

When considering privacy level for RRT for binary data, we can consider the very basic idea of accessing original ratings given perturbed ones. We need to analyze such a possibility. Remember that users preserve their ratings with the probability of θ ($r_u \leq \theta$). Assume that the original ratings of X are disclosed as Y to the

server after RRT is applied. Reconstruction probability can be stated as the conditional probability $P(X|Y)=\frac{p(Y|X)p(X)}{p(Y)}$, which then yields Eq. (1) [6]:

$$P(X|Y)=\frac{\theta^2+\theta p(Y)-\theta}{2\theta p(Y)-p(Y)} \quad (1)$$

Recall that data can be divided into different number of groups so the above formulation needs a small change to reflect the effect of grouping. Assuming there are M groups the above privacy measure becomes Eq. (2) as discussed in [6]:

$$P(X|Y)=\left[\frac{\theta^2+\theta p(Y)-\theta}{2\theta p(Y)-p(Y)}\right]^M \quad (2)$$

Fig. 1 shows how privacy level changes with varying M and θ values. Privacy level increases as number of group increases because more groups introduce more randomness and privacy level decreases as θ increases from 0.51 up to 1. This is because θ diminishes randomness as it grows while M adds randomness as it increases. Remember that complementary θ values have the same effect so there is no need to include them.

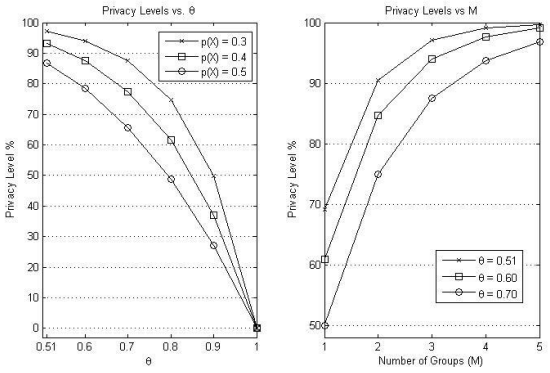

Figure 1. Privacy level with data disguising parameters

The trend in the Fig. 1 can be also confirmed with mutual information between original data, X, and perturbed data, Y. Mutual information defines how much uncertainty of X removed when Y is disclosed. Mutual information for between X and Y is presented as $I(X; Y) = H(X) - H(X|Y) = 1 - h(p)$, where H is entropy and $h(p)$ is binary entropy function. Binary entropy function $h(p) = -plog_2(p) - (1-p)log_2(1-p)$ and p is the probability of reversing binary input. In randomized response model, this value is modeled by θ. Fig. 2 displays mutual information, which indicates that the effect disclosure of Y on the reduction of uncertainty of X. As seen in the Fig. 2, mutual information is the lowest when θ is around 0.5, where uncertainty is maximum. Fig. 2 supports the privacy level depicted in Fig. 1. Similarly, as θ goes up, randomness diminishes and privacy level decreases. This trend is confirmed with mutual information graph in Fig. 2.

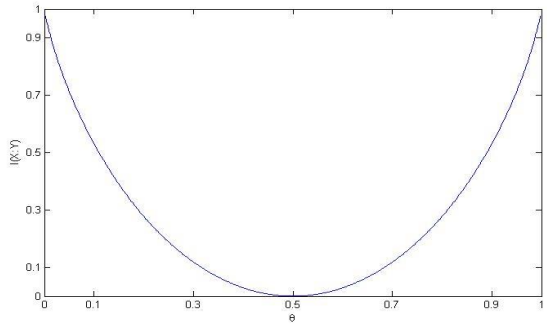

Figure 2. Mutual Information vs. θ

5. DISCOVERING FAKE RATINGS

When discovering which items are rated, exploiting noise elimination techniques can be useful if items are numerically rated for randomization process. Randomization proposed by Polat and Du [4] inserts some fake ratings into user-item matrix and this method of inserting fake ratings can be considered as noise to the original data. Noise elimination techniques can be useful to some extent to derive original ratings. RRT possibly creates multiple groups and reverses ratings if randomly selected value by each user is greater than the predetermined θ value. RRT alters the characteristics of original data deeply. Additionally, fake ratings inserted into empty cells and their characteristic is not different from original ratings, which makes them difficult to discover. Inserted items are just 1s and 0s just like other original items. On the other hand, reversing binary ratings distorts data pattern dramatically and this change cannot be easily referred noise as in randomization. Masked data in RRT takes a new form, which is very different from the original one. Assume that θ is 0.65 and disguised data would have about 0.35 percent different ratings from original matrix even though we have not considered the inserted items yet. If inserted items are taken into account, the change between masked and original data becomes larger. The comparison of original and disguised data can be seen in Fig. 3.

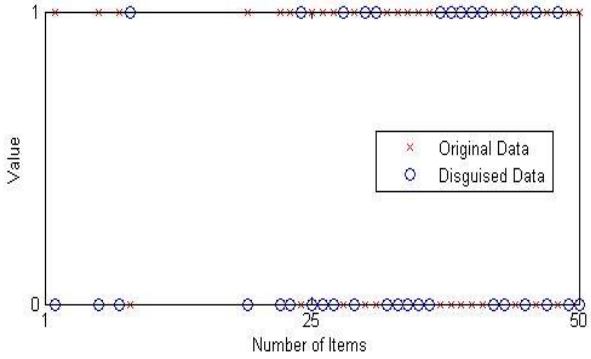

Figure 3. RRT distribution

Although we believe that noise elimination techniques do not help discover true rated items, exploiting public information could be useful. Since we are interested in if an item is rated or not, collecting auxiliary information could reveal high degree of useful information. Public information is collected about targeted data set to test our hypothesis that auxiliary information would identify true rated items with decent accuracy. The data set is MovieLens

(ML) with movie ratings of 943 users for 1,682 movies. Demographic data is already available with the data set. Internet Movie Database (IMDB) is a celebrated movie database and we have collected number of votes made for each movie in our data set. Number of votes is a good piece of information to determine if a movie is rated. Additionally, ML comes with demographic information of users.

5.1 Exploiting Auxiliary Information

Our intuition to discover rated items is based on an idea of collecting public auxiliary information about the data set we are working with. Our data set is movie related. First, we have collected number of votes being made for each movie in our data set from IMDB web site. Number of votes is an important piece of information to decide which items being voted by users regardless of whether items have been voted 1 or 0. The important point for this study is to identify if it has been voted, the exact value of the rating can be analyzed after determining rated items. Second, some information contained in the data set has been utilized. Demographic user data and movie-genre information are available in the data set. It means that the server can access such information so that they can be exploited. Demographic information about users such as their age and gender can be useful by integrating them with movie-related auxiliary information while analyzing the taste of the users.

While determining which items have been voted, the abovementioned auxiliary information has been used. Essentially, two main cases should be considered, constant β and β_{max}. The number of actually rated items for a user can be calculated as formulated in the previous section for the first case, where β is constant (m_t). After estimating the actual number of rated items, we need to find out which items are rated by users. At this point, auxiliary public information might be helpful to discover the exact items being voted. As mentioned before, number of votes from IMDB has been collected for each movie and they are sorted in descending order. Then, m_t number of movies with IMDB higher number of votes are selected as rated by the user. This approach is quite simple to reveal which items are voted by whom. The drawback of this approach is that movie with higher votes are always promoted while lower votes are neglected.

In the second case, β_u is selected by each users independently over the range (0, β_{max}]. In this case, determining number of rated items for each user is difficult because we need to guess β_u. Instead, number of users voting for an item can be estimated by using expected value of β_u, which is β_{ue} or $E[\beta_u] = \beta_{max}/2$. Since each user select a random β_u, β_{ue} can be utilized for each item. The calculation of estimated number of users voted for an item n_{users} can be performed using β_{ue}, $n_{users} = (n_f - n_{items}\beta_{ue})/(1-\beta_{ue})$, where n_f is number of filled row after random filling and n_{items} is the number of total items. For this case, movie related public information such as IMDB vote numbers cannot be utilized as in constant β case. That method of discovering rated items tries to find out which items are voted a particular user. Such a case requires more movie-user related public information because each user is evaluated independently and we need to make a decision among movies for a particular user. On the other hand, β_{max} case is more interested in finding which users vote a particular item. At this time, we are handling the problem in items' perspective. The second case handles each movie independently and a decision has to be made among users. Hence, user related auxiliary information about movies would be more helpful to select which users rate for the item. Therefore, we have integrated demographic user data along with movie genre data believing that people's taste of movie-genre differ by age group. We think that some age groups are more willing to watch some movie genres. Our algorithm to discover rated items is a three-phase algorithm. First, each item is checked based on its genre (comedy, action, adventure, drama, or thriller). Then, users' age groups are determined and we leave users cell rated if age group and movie genre relation we have defined holds. For example, if a user is young, we assume that he likes comedy. We have couple of rules for each age group inspired by a report conducted for British Film Institute[1]. After this process, there still might be some user cells to be marked rated. In order to decide the remaining users who rate the item, we list the number of votes of each user in the disguised matrix. Higher number of votes for a user in the disguise matrix can be an indication of that particular user has also voted for the related movie. Then, the third step of the algorithm is to mark remaining users as rated based the number of votes they have in the disguised data.

6. EXPERIMENTS

6.1 Data Set and Evaluation Criteria

In order to evaluate the overall success of our proposed schemes, we conducted various experiments using a real data set called ML. We performed trials by varying the values of different controlling parameters. The ML data set includes ratings of 943 users on 1,682 movies. It contains 100.000 ratings made by such users. The ratings are discrete and vary from 1 to 5. Each user rated at least 20 movies. The data set is a sparse data set.

To evaluate the result from the proposed approaches, classification accuracy (CA) is preferred as a metric. Two types of CA are considered for this study. Classification accuracy for actually rated items (CAR) is the ratio of correctly determined rated items made in original rating matrix. Classification accuracy for filled items (CAF) is the ratio of correctly determined items as empty of filled items in the disguised matrix. Such a ratio can be computed by comparing the estimated matrix with the original one.

6.2 Methodology

ML data set is on a 5-star scale. Its ratings should be transformed into binary ratings. It is converted into a binary format by marking items with rating less than or equal to 3 as dislike and greater than 3 as like [18]. We have repeated our test 10 times due to randomness and displayed the overall averages. There might be different factors that must be controlled in RRT scheme. As we discuss throughout the paper, these control parameters in terms of data disguising are θ and M. In random filling, the control parameters discussed in the paper is how β is selected either constant or over a range. Therefore, three parameters are going to be explored throughout the experiments. Since varying values of such parameters might affect the success of the proposed

[1] This report by Northern Allience and Ipsos MediaCT is available at http://old.bfi.org.uk/publications/openingoureyes/downloads/Appendix-2-Results-Tables-Cultural-Contribution-of-Film.pdf

approaches, we conducted different sets of trials by varying the values of such controlling parameters.

6.3 Experiments

6.3.1 Effects of data disguising parameters: θ & M

In the first set of experiments, the effects of varying data disguising parameters (θ and M) are tested. As discussed earlier, data disguising parameters have direct effects on privacy level of original data. These parameters control how much privacy is provided in terms of preserving original ratings of individuals. These parameters are related to the first aspect of privacy, which is to preserve actual values of ratings. Hence, we do not expect that these parameters will influence the success of our method of discovering fake ratings. In this experiment, β is set 3; and θ and M values vary. Fig. 4 displays the outcomes with varying values of controlling parameters.

Fig.4 confirms our intuition that data disguising parameters have no effect on discovering fake parameters when auxiliary public information is exploited. As one can see in the Fig. 4, the results for varying M and θ values are almost the same in the range less than 0.86 and greater than 0.84 for constant β. Outcomes have a similar trend for β_{max}. Especially, the outcomes for increasing M are so close to each other that it is very difficult to recognize them independently. As a result, Fig. 4 shows that masking parameters do not have any dominating effect on discovering fake items

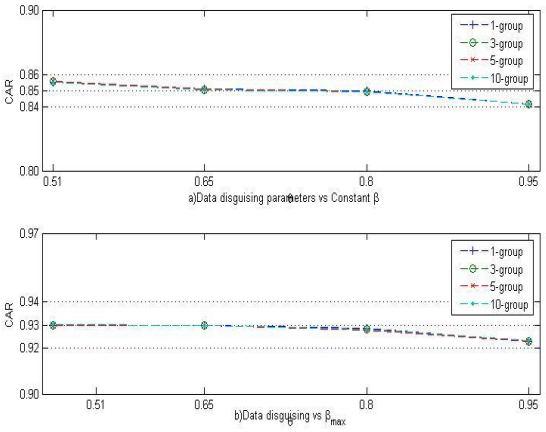

Figure 4. Effects of data disguising parameters

6.3.2 Effects of random filling parameter: β

After seeing how data disguising parameters affect our method of discovering fake items, we now consider random filling parameters. Remember that there are two options available while inserting fake items into a user vector. First one is to insert $\beta\%$ fake items for each user. The second one is that each user picks a β_u independently. Two output parameters are watched for this experiment, CAR and CAF values. We believe that the second option will be more resilient compared to the first option in terms of discovering fake ratings because of more randomness in inserting fake ones.

In constant β approach, we can at least almost accurately calculate number of true ratings (m_t). In general, we expect a decrease in CAR value as β values increase regardless of constant β or β_{max}. Greater β values insert more fake ratings and introduce more randomness to identify fake ratings; therefore, a decrease in CAR is possible. On the other hand, CAF is an interesting parameter to watch. Although growing β introduces randomness, interestingly CAF is also expected to increase. Greater β values increase the number of filled cells (m_f). Since the number of rated items (m_t) will not alter for varying β values, the remaining items, which are marked unrated will grow ($m_f - m_t$). m_f always grows while m_t remains the same. The effect of m_t is smaller when compared to $m_f - m_t$. Thus, as β increases, we expect CAF increases, as well. β is varied between 1, 3, 5, 6, and 10 percent and the test is repeated 100 times for both constant β and β_{max}. θ and M have limited effects as discussed in the previous experiments and they are set 0.65 and 3, respectively.

Fig. 5 depicts the result of our experiments. CAR gets worse for both cases as we intuitively believe. This is expected because larger β values cause more items to be inserted and our method has to choose m_t number of items from a larger set. On the other hand, constant β results are poorer than β_{max} results contrary to our assumption. This might be because auxiliary public information is exploited for our experiments and auxiliary information for β_{max} test is more helpful to discover fake ratings. Recall that we use different kinds of auxiliary information for both cases. CAF gets better for both cases as we intuitively believe. This is expected because number of filled items becomes too large when compared to number of rated items as β grows. As previously analyzed, increasing number of inserted items becomes too large as β increases and discovering these fake items becomes more possible.

7. CONCLUSION

Privacy-preserving collaborative filtering schemes have been receiving increasing attention to convince people to be part of recommendation systems. On the other hand, these schemes should be analyzed whether they offer the claimed privacy. In this study, we perform an analysis of randomization for binary data and exploit auxiliary and public information to discover fake ratings filled into user-item matrix. We perform different experiments with data disguising parameters. Our results show that public information is a great source to infer so-claimed private individual binary data perturbed using randomization. As a future goal, we are planning to investigate other schemes in terms of privacy and discuss what measures might be taken to enhance privacy.

8. ACKNOWLEDGMENTS

This work is supported by Grant 113E262 from TUBITAK.

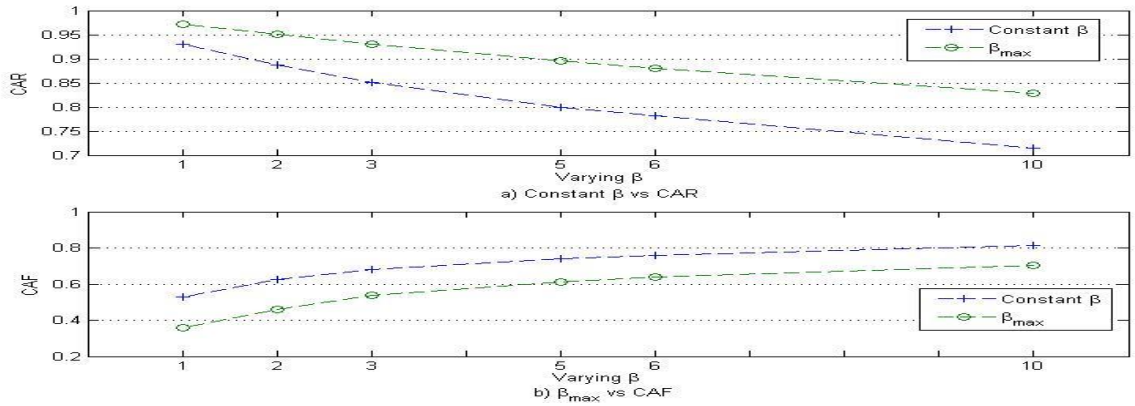

Figure 5. Effects of random filling parameters

9. REFERENCES

[1] Canny, J. Collaborative filtering with privacy. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002, 45-57.

[2] Canny, J. Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002, 238-245.

[3] Cranor, L. F. `I didn't buy it for myself' Privacy and ecommerce personalization, In *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, 2003, 111-117.

[4] Polat, H. and Du, W. Privacy-preserving collaborative filtering using randomized perturbation techniques, In *Proceedings of the 3rd IEEE International Conference on Data Mining*, 2003, 625-628.

[5] Polat, H. and Du, W. Privacy-preserving collaborative filtering. *International Journal of Electronic Commerce*, *9* (*4*), 2005, 9-35.

[6] Polat, H. and Du, W. Achieving private recommendations using randomized response techniques. *Lecture Notes in Computer Science*, *3918*, 2006, 637-646.

[7] Bilge, A., Kaleli, C., Yakut, I., Gunes, I., and Polat, H. A survey of privacy-preserving collaborative filtering schemes. *International Journal of Software Engineering and Knowledge Engineering*, *23* (*8*), 2013, 1085-1108.

[8] Warner, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *J. Am. Stat. Assoc.*, 60 (309), 1965, 63-69.

[9] Okkalioglu, M., Koc, M., and Polat, H. Deriving binary ratings from masked data. *Submitted for review*.

[10] Agrawal, R. and Srikant, R. Privacy-preserving data mining. In *Proceedings of the 19th ACM SIGMOD International Conference on Management of Data*, 2000, 439-450.

[11] Agrawal, D. and Aggarwal, C. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium of Principles of Database Systems*, 2001, 247-255.

[12] Rizvi, S. J. and Haritsa, J. R. Maintaining data privacy in association rule mining. In *Proceedings of the 28th International Conference on Very Large Data Bases*, 2002, 682-693.

[13] Evfimievski, A., Gehrke, J., and Srikant, R. Limiting Privacy breaches in privacy preserving data mining. In *Proceedings of the Twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2003, 211-222.

[14] Huang, Z. and Du, W. OptRR: Optimizing randomized response schemes for privacy-preserving data mining. In *Proceedings of the 24th International Conference on Data Engineering*, 2008, 705-714.

[15] Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W., and Shmatikov, V. "You might also like:" Privacy risks of collaborative filtering. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2011, 231-246.

[16] Zhang, S., Ford, J., and Makedon, F. Deriving private information from randomly perturbed ratings. In *Proceedings of the 6th SIAM International Conference on Data Mining*, 2006, 59-69.

[17] Yakut, I. and Polat, H. Estimating NBC-based recommendations on arbitrarily partitioned data with privacy. *Knowledge-Based Systems*, *36*, 2012, 353-362.

[18] Miyahara, K. and Pazzani, M. Collaborative filtering with the simple Bayesian classifier. *Lecture Notes in Computer Science*, *1886*, 2000, 679-689.