Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 33 – 42

Procedia
Computer Science

20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

Randomization-based Privacy-preserving Frameworks for Collaborative Filtering

Zeynep Batmaz[a], Huseyin Polat[a,*]

[a] *Computer Engineering Department, Anadolu University, 26470 Eskisehir, Turkey*
{zozdemir, polath@anadolu.edu.tr}

Abstract

Randomization-based privacy protection methods are widely used in collaborative filtering systems to achieve individual privacy. The basic idea behind randomization utilized in collaborative filtering schemes is to add randomness to original data in such a way so that required levels of accuracy and privacy can be achieved. Although there are various studies on privacy-preserving collaborative filtering using randomization, there are no well-defined privacy-preserving frameworks for collaborative filtering algorithms based on randomization. In this paper, we present eight randomization-based privacy-preserving frameworks for privacy protection in collaborative filtering schemes. We first group privacy-preserving methods into two broad categories. We then classify them based on private data. Final grouping is done while considering varying privacy concerns of individual users. The frameworks can be chosen according to individual users' expectations and be utilized for privacy protection. The well-defined privacy-preserving frameworks form a basis for privacy protection based on randomized perturbation and randomized response techniques in collaborative filtering studies.

Keywords: Framework, privacy, randomization, collaborative filtering, data masking

1. Introduction

E-commerce helps online vendors collect user preferences about various products traded over the Internet. Such preferences are considered valuable and private asset because they can be used for recommendation purposes and help online vendors increase their sales/profits. Moreover, revealing such data might cause various privacy risks. To transform such data into knowledge, e-commerce sites employ recommender systems. Collaborative filtering (CF) is one of the most common recommender systems. CF schemes employ other people's data for generating predictions for single items and top-*N* recommendation lists. A traditional CF algorithm consists of three steps such as similarity computation, neighbor selection, and recommendation estimation[1].

Users provide their preferences about products they bought or showed interest to online vendors. An $n \times m$ user-item matrix is created to store such data, where n and m represent number of users and items, respectively.

[*] Corresponding author. Tel.: +90-222-321-3550 ; fax: +90-222-323-9501.
E-mail address: polath@anadolu.edu.tr

doi:10.1016/j.procs.2016.08.091

Without privacy protection, user might not feel comfortable to share their preferences with e-commerce sites. They either refuse to give data at all or tend to provide false data. High quality results can be derived from high quality data. To collect high quality enough data for recommendation purposes, privacy-preserving collaborative filtering (PPCF) schemes are used[2,3]. Users' preferences represented by numeric or binary ratings can be considered private. Furthermore, it might be more damaging for revealing whether a user bought an item or not. Hence, rated/unrated items can also be considered private data. Privacy protection methods aim to mask such private data (ratings and rated/unrated items). To achieve privacy, different privacy protection methods are used in PPCF systems[2]. The most common method is known as randomization in general. Randomization adds some randomness to original data. Since CF algorithms usually depend on aggregate data, it is still possible to estimate accurate recommendations from perturbed data. Level of randomness should be chosen in such a way so that accurate predictions can be estimated while preserving privacy. However, accuracy and privacy are conflicting goals. Moreover, users' privacy concerns might be different. Likewise, user preferences can be represented using numeric or binary ratings. These constraints require different privacy protection measures.

As presented in[2,3], there are numerous PPCF studies based on randomization. Different researchers propose to use randomization-based privacy-preserving techniques for privacy in CF systems. The methods can be grouped as randomized perturbation techniques (RPTs) and randomized response techniques (RRTs) for numeric and binary ratings-based PPCF schemes, respectively. There are also different privacy control parameters. Their values should be carefully chosen. Varying users' privacy concerns should be considered as well. The studies on randomization presented in[2,3] do not use steady privacy-preserving measures. It is more appropriate to employ common privacy-preserving frameworks for fair comparisons in CF schemes. Well-defined and structured privacy-preserving frameworks should be presented for CF systems.

Following the above-mentioned motivation, we design eight privacy-preserving frameworks for CF. The frameworks are structured for numeric and binary ratings-based schemes because they use RPTs and RRTs, respectively. Some users might consider true ratings only as confidential while some might consider true ratings and rated/unrated items as private data. Hence, different frameworks should be designed for two different confidential data considerations. Finally, due to varying privacy concerns, users might perturb their confidential data variable; or they might decide to use invariable data masking. Thus, two different approaches are proposed for variable and invariable data perturbation. Considering these cases, we design eight frameworks. These frameworks will form a common base for researchers in the CF research field.

2. Related Work

Agrawal and Srikant[4] propose to use value distortion on randomization as a privacy-preserving method. The true value x_i is masked with a random value r_i; and $x_i + r_i$ is shared. Note that r_i is drawn from a distribution with zero mean. The authors also discuss privacy levels provided by randomization based on uniform and Gaussian random number distributions. Polat and Du[5,6,7] propose RPTs in order to achieve confidentiality in CF systems. The authors consider invariable data disguising. In addition to invariable data perturbation, variable data disguising-based RPT are used in PPCF schemes[8,9]. Due to varying privacy concerns, users tend to disguise their private data in such a way so that required levels of privacy is achieved. This results inconsistent data masking. Compared to invariable data perturbation, variable data disguising provides higher privacy level. Users randomly decide random number distribution and the related standard deviation. Uniform or Gaussian random number distribution can be used to generate random noise. To disguise rated/unrated items, the authors propose to fill in some randomly chosen unrated item cells. Number of the filled cells can be selected uniformly over a range. Similarly, standard deviations of the random number distributions can be uniformly randomly selected over a specified range. Gong[10] utilizes RPTs to achieve privacy in both centralized and decentralized PPCF schemes. Basu et al.[11] also use RPTs to protect individual user ratings in Slope One predictors for CF. The authors focus on masking real ratings only. They also propose to disguise deviations computed by subtracting the related ratings off-line. Zhu et al.[12] utilize data perturbation to hide the true ratings of the neighbors.

Numeric and binary ratings hold different properties. RPTs are suitable to perturb numeric ratings while RRTs are appropriate for masking binary ratings. Warner[13] proposes a surveying technique that forms the basis of RRTs. The method allows respondents to answer sensitive questions while preserving privacy. Instead of asking sensitive questions directly, two related questions are asked. The respondents use a randomized device to decide which question to answer. Polat and Du[14] propose a data masking method based on RRTs. They consider using group-based schemes for disguising binary ratings. The authors aim to hide both true ratings and rated/unrated items. Data in each group is masked independently. A random number is chosen and it is compared with a predefined value to decide whether to use true data or reversed data (1s are reversed to 0s and 0s are reversed to 1s). Kaleli and Polat[15] mask binary ratings using RRTs and show how to estimate naïve Bayesian classifier-based recommendations. They scrutinize how accuracy changes with varying number of groups. As expected, increasing number of groups makes accuracy worse due augmented randomness. Kikuchi and Mochizuki[16] perform a perturbation in a randomized response scheme.

3. Classification of Privacy-Preserving Frameworks

In this section, we explain the privacy-preserving frameworks and present their pseudo-codes. The frameworks are designed in such a way so that the appropriate one can be chosen based on the user requirements. The frameworks are structured based on the following three dimensions:

1. **Randomization type:** User preferences can be represented using numeric or binary ratings. In numeric ratings-based CF schemes, data is masked using RPTs. On the other hand, data is perturbed using RRTs in binary ratings-based CF systems. Thus, the frameworks can be grouped into two classes according to the randomization type as follows:

   (a) RPTs
   (b) RRTs

2. **Confidential data:** Users concern about their private data. Public data can be shared with e-commerce sites. However, users do not want to provide their private data in plain form to online vendors. Generally speaking, confidential data can be grouped into two broad categories as follows:

   (a) Ratings: Users usually do not want other to learn their true ratings about the products they bought or showed interest. Such ratings can be exploited for profiling, unsolicited marketing, price discrimination, government surveillance, and so on[17]. Due to these privacy risks, users mask their ratings and send perturbed ratings to the CF system.
   (b) Rating and rated/unrated items: In addition to ratings, it might be more damaging to reveal rated/unrated items. Rated items disclose information about users' tendencies and interests. Nobody wants others to learn that she bought specific magazines or products. Likewise, unrated items mean that users have not bought them yet; and thus, such products are prospective items to be purchased and the users are potential customers. This might lead to unsolicited marketing and price discrimination. Due to these reasons, users mask their ratings and rated/unrated items.

3. **Privacy concerns:** Privacy concerns differ from one user to another. Varying privacy concerns cause variable data masking. On the other hand, users can perturb their confidential data in the same way, known as invariable data masking. The frameworks can be grouped into two classes based privacy concerns dimension as follows:

   (a) Invariable: In this case, all users disguise their private data consistently. The CF system forces each user to mask their private data following the same procedure. Invariable data disguising is less complex and provides smaller privacy level.
   (b) Variable: Some users are privacy fundamentalist while some are marginally concerned about their privacy. The majority of the users are less concerned about their privacy than the fundamentals. Due to varying privacy concerns and expectations, users might decide to mask their data inconsistently.

As presented above, there are three major dimensions for classifying the privacy-preserving frameworks. Each dimension consists of two classes; and thus, that leads to eight privacy-preserving frameworks as follows:

1. Framework-1: Masking ratings only invariably using RPTs-RPTRI
2. Framework-2: Masking ratings only variably using RPTs-RPTRV
3. Framework-3: Masking ratings and rated/unrated items invariably using RPTs-RPTR^{2}I
4. Framework-4: Masking ratings and rated/unrated items variably using RPTs-RPTR^{2}V
5. Framework-5: Masking ratings only invariably using RRTs-RRTRI
6. Framework-6: Masking ratings only variably using RRTs-RRTRV
7. Framework-7: Masking ratings and rated/unrated items invariably using RRTs-RRTR^{2}I
8. Framework-8: Masking ratings and rated/unrated items variably using RRTs-RRTR^{2}V

4. Privacy-Preserving Frameworks

Our frameworks are based on RPTs and RRTs. We first basically describe these methods. RPTs select a random number distribution with zero mean and a standard deviation. Uniform or Gaussian random number distributions can be employed to generate random noise. Using the distribution with the chosen parameter values, required number of random numbers are generated. These random values are finally added to the private data items that are supposed to be disguised. In order to mask rated/unrated items, randomly selected some of the unrated item cells are filled in with random numbers, too. All users can use the same random number distribution with the fixed values of privacy control parameters. For variable disguising, each user can randomly select random number distribution and the values of the parameters over a specified range.

RRTs first determine number of groups and a threshold. If the ratings are grouped into a single group, this scheme is known as one-group method. If the number of groups two or more, then the corresponding method is called multi-group scheme. To mask true binary ratings, the ratings are grouped according to the number of groups. Data in each group is then independently masked. A random number is chosen for each group and the random numbers are compared with the chosen threshold. If the random number is larger than the threshold, binary ratings are reversed (1s are transformed into 0s and 0s are transformed into 1s) and sent to the CF system. Otherwise, the true ratings are sent. To perturb rated/unrated items, randomly selected some of the unrated item cells are filled in with binary ratings. For both of RRTs and RPTs, number of randomly selected unrated item cells depends on the amount of the unrated item cells. However, number of unrated cells to be filled in should be carefully determined. Filling too many unrated item cells might significantly affect originality of the collected true data. Hence, it is reasonable to determine the number of unrated item cells to be filled in based on the density of the user ratings vector. Note that users must divide their ratings vectors into the same number of groups for estimating recommendations with decent accuracy in RRTs.

4.1. Framework-1: Masking ratings only invariably using RPTs-RPTRI

The framework's pseudo-code is given in Algorithm 1. It uses RPTs and masks ratings only. All users perturb their data in the same way.

Algorithm 1 Masking ratings only invariably using RPTs-RPTRI

```
CF System:
    decides the random number distribution                              ▷ uniform or Gaussian
    chooses the random number distribution's standard deviation         ▷ determine $\sigma$
    lets users know them                  ▷ publish random number distribution and $\sigma$
Each user $u$:
    computes number of her ratings                                      ▷ find $m_u$
    if the random number distribution is uniform then
        computes $\alpha = 3^{1/2}\sigma$       ▷ random numbers are chosen over the range [-$\alpha$, +$\alpha$]
    end if
    generates random numbers $r_{uj}$ for $j = 1, 2, \ldots, m_u$   ▷ $m_u$ random numbers are needed to mask $m_u$ ratings
    adds random numbers to the corresponding ratings          ▷ ratings are masked with random noise
```

4.2. Framework-2: Masking ratings only variably using RPTs-RPTRV

The framework's pseudo-code is given in Algorithm 2. Unlike the first framework, users inconsistently perturb their ratings only. Each user determines the random number distribution from given choices independently. Furthermore, each user chooses the standard deviation of the random number distribution over the given range. Note that the range and the set of the random number distributions are determined by CF system.

Algorithm 2 Masking ratings only variably using RPTs-RPTRV

```
CF System:
    decides the set of random number distributions    ▹ the set includes uniform and Gaussian distributions
    decides the upper bound of standard deviations of the random number distribution    ▹ set $\sigma_{max}$
    lets users know them    ▹ publish random number distributions set and $\sigma_{max}$
Each user $u$:
    computes number of her ratings    ▹ find $m_u$
    uniformly randomly selects random number distribution    ▹ choose either uniform or Gaussian using flip coin
    uniformly randomly chooses standard deviation of the distribution from the range $(0, \sigma_{max}]$    ▹ $\sigma_u \in (0, \sigma_{max}]$
    if the random number distribution is uniform then
        computes $\alpha_u = 3^{1/2}\sigma_u$    ▹ random numbers are chosen over the range $[-\alpha_u, +\alpha_u]$
    end if
    generates random numbers $r_{uj}$ for $j = 1, 2, \ldots, m_u$    ▹ $m_u$ random numbers are needed to mask $m_u$ ratings
    adds random numbers to the corresponding ratings    ▹ ratings are masked with random noise
```

4.3. Framework-3: Masking ratings and rated/unrated items invariably using RPTs-RPTR^{2}I

The framework's pseudo-code is given in Algorithm 3. Since confidential data includes ratings and rated/unrated items, users invariably disguise both their true ratings and rated/unrated items using RPTs. Due to invariable data perturbation, all users disguise their private data in the same way. The CF system forces each user to mask their private using the given choices.

Algorithm 3 Masking ratings and rated/unrated items invariably using RPTs-RPTR^{2}I

```
CF System:
    decides the random number distribution    ▹ uniform or Gaussian
    chooses the random number distribution's standard deviation    ▹ determine $\sigma$
    decides the value of filling parameter    ▹ choose $\beta$
    lets users know them    ▹ publish random number distribution, $\sigma$, and $\beta$
Each user $u$:
    computes number of her ratings    ▹ find $m_u$
    computes number of her unrated items cells    ▹ $m_{un} = m - m_u$
    if the random number distribution is uniform then
        computes $\alpha = 3^{1/2}\sigma$    ▹ random numbers are chosen over the range $[-\alpha, +\alpha]$
    end if
    computes number of the unrated items cells to be filled in    ▹ calculate $m_{uf}$
        $m_{uf} = \beta \times m_u/100$    ▹ value of $\beta$ depends on how many unrated items cells to be filled in
    generates random numbers $r_{uj}$ for $j = 1, 2, \ldots, m_u + m_{uf}$    ▹ $m_u + m_{uf}$ random numbers are needed for masking
    uniformly randomly selects $m_{uf}$ number of the $m_{un}$ unrated items cells    ▹ $m_{uf}$ unrated items cells to be filled in
    adds random numbers to the corresponding ratings and the chosen unrated cells    ▹ ratings and rated/unrated
items are masked with noise data
```

4.4. Framework-4: Masking ratings and rated/unrated items variably using RPTs-RPTR^{2}V

The framework's pseudo-code is given in Algorithm 4. Inconsistent data disguising is performed by each user using RPTs. All users perturb their ratings and the rated/unrated items.

Algorithm 4 Masking ratings and rated/unrated items variably using RPTs-RPTR^{2}V

```
CF System:
    decides the set of random number distributions    ▷ the set includes uniform and Gaussian distributions
    decides the upper bound of standard deviations of the random number distribution    ▷ set $\sigma_{max}$
    decides the upper bound of filling parameter    ▷ set $\beta_{max}$
    lets users know them    ▷ publish random number distribution, $\sigma_{max}$, and $\beta_{max}$
Each user u:
    computes number of her ratings    ▷ find $m_u$
    computes number of her unrated items cells    ▷ $m_{un} = m - m_u$
    uniformly randomly selects random number distribution    ▷ choose either uniform or Gaussian using flip coin
    uniformly randomly chooses standard deviation of the distribution from the range (0, $\sigma_{max}$]    ▷ $\sigma_u \in (0, \sigma_{max}]$
    if the random number distribution is uniform then
        computes $\alpha_u = 3^{1/2}\sigma_u$    ▷ random numbers are chosen over the range [-$\alpha_u$, +$\alpha_u$]
    end if
    uniformly randomly chooses value of filling parameter from the range (0, $\beta_{max}$]    ▷ $\beta_u \in (0, \beta_{max}]$ and $\beta_{max}$ depends on density of user ratings vector ($d_u$)
    computes number of the unrated item cells to be filled in    ▷ calculate $m_{uf}$
        $m_{uf} = \beta_u \times m_u / 100$    ▷ $\beta_u$ percent of the unrated items cells are computed
    generates random numbers $r_{uj}$ for $j = 1, 2, \ldots, m_u + m_{uf}$    ▷ $m_u + m_{uf}$ random numbers are needed for masking
    uniformly randomly selects $m_{uf}$ number of the $m_{un}$ unrated items cells    ▷ $m_{uf}$ unrated item cells to be filled in
    adds random numbers to the corresponding ratings and the chosen unrated cells    ▷ ratings and rated/unrated items are masked with noise data
```

4.5. Framework-5: Masking ratings only invariably using RRTs-RRTRI

The framework's pseudo-code is given in Algorithm 5. In binary ratings-based filtering schemes, RRTs can be used for data masking. All users perturb their ratings only in the same way.

Algorithm 5 Masking ratings only invariably using RRTs-RRTRI

```
CF System:
    decides the number of groups    ▷ determine $M$, $M$ is a small positive integer
    decides the threshold    ▷ choose $\theta$
    lets users know them    ▷ publish $M$ and $\theta$
Each user u:
    groups her ratings into $M$ groups    ▷ divide the ratings vector into $M$ sub-vectors
    uniformly randomly creates random numbers $r_{uj}$ over the range [0, 1] for $j = 1, 2, \ldots, M$    ▷ $M$ random numbers are needed for $M$ groups
    compares $r_{uj}$ and $\theta$ for $j = 1, 2, \ldots, M$    ▷ to decide wether to reverse the ratings or not in the $j^{th}$ group
        if $r_{uj} < \theta$ then
            keeps true ratings in the $j^{th}$ group    ▷ do not reverse any ratings
        else
            reverses her ratings in the $j^{th}$ group    ▷ transform 1s into 0s and 0s into 1s
        end if
    sends perturbed data    ▷ send disguised ratings vector
```

4.6. Framework-6: Masking ratings only variably using RRTs-RRTRV

The framework's pseudo-code is given in Algorithm 6. Users might decide to inconsistently mask their ratings only using RRTs.

Algorithm 6 Masking ratings only variably using RRTs-RRTRV

```
CF System:
    decides the number of groups                                  ▷ determine $M$
    decides the upper bound of the threshold                      ▷ choose $\theta_{max}$
    lets users know them                                          ▷ publish $M$ and $\theta_{max}$
Each user u:
    groups her ratings into $M$ groups                            ▷ divide the ratings vector into $M$ sub-vectors
    uniformly randomly creates $\theta_{uj}$ over the range $(0, \theta_{max}]$ for $j = 1, 2, \ldots, M$    ▷ $\theta_{uj} \in (0, \theta_{max}]$
    uniformly randomly creates random numbers $r_{uj}$ over the range [0, 1] for $j = 1, 2, \ldots, M$    ▷ $M$ random numbers are needed for $M$ groups
    compares $r_{uj}$ and $\theta_{uj}$ for $j = 1, 2, \ldots, M$    ▷ to decide wether to reverse the ratings or not in the $j^{th}$ group
        if if $r_{uj} < \theta_{uj}$ then
            keeps true ratings in the $j^{th}$ group              ▷ do not reverse any ratings
        else
            reverses her ratings reverses her ratings in the $j^{th}$ group    ▷ transform 1s into 0s and 0s into 1s
        end if
    sends perturbed data                                          ▷ send disguised ratings vector
```

4.7. Framework-7: Masking ratings and rated/unrated items invariably using RRTs-RRTR^{2}I

The framework's pseudo-code is given in Algorithm 7, where ratings and rated/unrated items are masked.

Algorithm 7 Masking ratings and rated/unrated items invariably using RRTs-RRTR^{2}I

```
CF System:
    decides the number of groups                                  ▷ determine $M$
    decides the threshold                                         ▷ choose $\theta$
    decides the value of filling parameter                        ▷ select $\beta$
    lets users know them                                          ▷ publish $M$, $\theta$, and $\beta$
Each user u:
    computes number of her ratings                                ▷ find $m_u$
    computes number of her unrated items cells                    ▷ $m_{un} = m - m_u$
    computes number of the unrated items cells to be filled in    ▷ calculate $m_{uf}$
        $m_{uf} = \beta \times m_u / 100$    ▷ value of $\beta$ depends on how many unrated items cells to be filled in
    uniformly randomly selects $m_{uf}$ number of the $m_{un}$ unrated items cells ▷ $m_{uf}$ unrated items cells to be filled in
    randomly fills in the chosen unrated item cells with binary ratings    ▷ fill in them with 1s or 0s
    groups her filled ratings vector into $M$ groups              ▷ divide the filled ratings vector into $M$ sub-vectors
    creates random numbers $r_{uj}$ over [0, 1] for $j = 1, 2, \ldots, M$    ▷ $M$ random numbers are needed for $M$ groups
    compares $r_{uj}$ and $\theta$ for $j = 1, 2, \ldots, M$    ▷ to decide wether to reverse the ratings or not in the $j^{th}$ group
        if $r_{uj} < \theta$ then
            keeps the same ratings in the $j^{th}$ group including the filled ones    ▷ do not reverse any ratings
        else
            reverses ratings in the $j^{th}$ group including the filled ones    ▷ transform 1s into 0s and 0s into 1s
        end if
    sends perturbed data                                          ▷ send disguised ratings vector
```

4.8. Framework-8: Masking ratings and rated/unrated items variably using RRTs-RRTR^{2}V

The framework's pseudo-code is given in Algorithm 8. In binary ratings-based PPCF schemes, users inconsistently perturb their ratings and the rated/unrated items using RRTs in order to achieve required levels of privacy.

Algorithm 8 Masking ratings and rated/unrated items variably using RRTs-RRTR^{2}V

```
CF System:
    decides the number of groups                                ▷ determine $M$
    decides the upper bound of threshold                        ▷ choose $\theta_{max}$
    decides the upper bound of filling parameter                ▷ set $\beta_{max}$
    lets users know them                                        ▷ publish $M$, $\theta_{max}$, and $\beta_{max}$
Each user u:
    computes number of her ratings                              ▷ find $m_u$
    computes number of her unrated item cells                   ▷ $m_{un} = m - m_u$
    uniformly randomly chooses value of filling parameter from the range $(0, \beta_{max}]$   ▷ $\beta_u \in (0, \beta_{max}]$ and $\beta_{max}$
  depends on density of user ratings vector ($d_u$)
    computes number of the unrated item cells to be filled in   ▷ calculate $m_{uf}$
        $m_{uf} = \beta_u \times m_u / 100$                     ▷ $\beta_u$ percent of the unrated items cells are computed
    uniformly randomly selects $m_{uf}$ number of the $m_{un}$ unrated items cells ▷ $m_{uf}$ unrated items cells to be filled in
    randomly fills the chosen unrated item cells with binary ratings   ▷ fill in them with 1s or 0s
    groups her filled ratings vector into $M$ groups            ▷ divide the filled ratings vector into $M$ sub-vectors
    uniformly randomly creates $\theta_{uj}$ over the range $(0, \theta_{max}]$ for $j = 1, 2, \ldots, M$   ▷ $\theta_{uj} \in (0, \theta_{max}]$
    uniformly randomly creates random numbers $r_{uj}$ over the range $[0, 1]$ for $j = 1, 2, \ldots, M$   ▷ $M$ random
  numbers are needed for $M$ groups
    compares $r_{uj}$ and $\theta_{uj}$                         ▷ to decide wether to reverse the ratings or not in the $j^{th}$ group
        if $r_{uj} < \theta_{uj}$ then
            keeps the same ratings in the $j^{th}$ group including the filled ones   ▷ do not reverse any ratings
        else
            reverses ratings in the $j^{th}$ group including the filled ones         ▷ transform 1s into 0s and 0s into 1s
        end if
    sends perturbed data                                        ▷ send disguised ratings vector
```

5. Discussion and An Example

We have presented eight privacy-preserving frameworks for CF schemes on numeric and binary ratings to achieve privacy. Since privacy and accuracy are conflicting goals, privacy-preserving frameworks should be designed correspondingly. An additional criterion, called online performance, should also be considered. For both uniform and Gaussian random number distributions, larger σ values provide larger randomness; and thus, privacy level increases. However, accuracy becomes worse due to augmented randomness[5,7]. Disguising private data using variable privacy frameworks improves privacy level because values of privacy control parameters need to be first guessed. Generally speaking, the probability of guessing β_u is 1 out of β_{max}. Also note that the value of β_{max} depends on the density of the users' ratings vectors, which are not known by the server. After estimating true β_u values, the probability of guessing true rating vector is 1 out of ($\beta_{max} \times C_y^x$), where x and y represent the number of ratings in disguised and true rating vectors, respectively[18,19]. Note that C_a^b is the number of a-combinations given a set of b. Privacy level of RRTs improves with increasing group numbers and θ values close to 0.5[14,15]. Conversely, smaller number of groups or θ values close to 1 or 0 make privacy worse while making accuracy better. In case of both unrated items and ratings are masked, the probability of guessing the number of filled unrated items m_{uf} is 1 out of $m'/2$, where m' represents the minimum of the number of 1s ($m_1{}'$) or 0s ($m_0{}'$) in disguised rating vector[14]. After estimating m_{uf}, the chosen unrated items filled as 1s and 0s can be guessed as 1 out of $C^{m_1'}_{m_{uf}/2}$ and 1 out of $C^{m_0'}_{m_{uf}/2}$, respectively. Therefore, the probability of estimating the fake ratings is 1 out of $(m'/2 \times C^{m_1'}_{m_{uf}/2} C^{m_0'}_{m_{uf}/2})$[14,15].

Considering privacy, accuracy, and online performance, appropriate privacy-preserving framework for achieving privacy in CF systems can be chosen as follows: (*i*) If higher accuracy and lower computational costs are more important than privacy, then Framework 1 and Framework 5 can be used for numeric and binary ratings-oriented CF systems, respectively. (*ii*) For numeric ratings-oriented CF systems, if providing higher privacy level with admissible accuracy and computational costs by letting each user to determine their own parameter values is more important, then Framework 2 should be used. Correspondingly, Framework 6 should be preferred for binary ratings-oriented CF systems. (*iii*) When providing higher privacy level by masking ratings and rated/unrated items with admissible accuracy and computational costs is more important, Framework 3 and Framework 7 should be utilized for numeric and binary ratings-oriented CF systems, respectively. (*iv*) If providing full privacy is more important then accuracy and lower computational costs, then Framework 4 and Framework 8 should be preferred for numeric and binary ratings-oriented CF systems, respectively.

To show how the proposed eight frameworks can be used to mask data, we use a simple example. We generate two user ratings vectors. One of them includes numeric ratings while the other consists of binary ratings, as shown in Table 1. The vectors include four ratings and six unrated items, where "-" shows unrated items. Thus, there are 10 items like i_1, i_2, ..., i_{10}. For Framework 1, we choose Gaussian distribution and set σ to 1. Then, we generate four random numbers like [-0.71, 1.35, -0.22, -0.59]. We add these random numbers to the true ratings and obtain the masked vector, as shown in Table 1 for Framework 1. For Framework 2, we choose 1 as σ_{max} and σ_u is uniformly randomly selected over the range (0, 1] as 0.0965. We use Gaussian distribution and generate four random numbers like [0.11, -0.16, -0.15, -0.12]. We add them to the real ratings and obtain the perturbed vector for Framework 2. We select Gaussian distribution for Framework 3, where σ and β are set to 1 and 50, respectively. It means that two unrated items cells are supposed to be filled in random data. Hence, we generate six random numbers like [0.05, -0.83, 0.53, 0.47, -0.63, 0.18]. Suppose that the fifth and the tenth items are selected to be filled in. The perturbed vector is shown in Table 1 for Framework 3. For Framework 4, we choose Gaussian distribution. We set σ_{max} and β_{max} to 1 and 50, respectively. Then, σ_u and β_u are uniformly randomly selected over the range (0, 1] and (0, 50] as 0.74 and 28, respectively. Thus, one unrated item cell will be filled in. Assume that the sixth item is filled. We generate five random numbers like [0.62, -0.40, 0.76, 0.81, 0.92], mask the vector, and obtain the perturbed one for Framework 4. For Frameworks 5-8, we set M to 2. Suppose that θ is set to 0.8; and 0.25 and 0.85 are uniformly randomly chosen two random numbers for two groups for Framework 5. Similarly, θ_{max} is set to 0.8 and θ_u is selected as 0.29 for Framework 6. Also, 0.42 and 0.04 are chosen as random numbers for two groups. For Framework 7, θ and β are selected as 0.8 and 50, respectively. Thus, two unrated items cells will be filled in. Assume that the third and the tenth items are selected and filled in 1 and 0, respectively. The two random numbers for the two groups are 0.44 and 0.10. Finally, for Framework 8, we choose θ_{max} as 0.8 and θ_u is uniformly randomly selected as 0.24. We set β_{max} to 50 and β_u is randomly selected as 33. Hence, one unrated item will be filled in. Assume that the fifth item is filled with 0. The two random numbers are 0.45 and 0.08 for two groups. The disguised vectors are shown in Table 1.

6. Conclusions and Future Work

Protecting users' confidential data is extremely important for improving quality of a recommender system. Used data set in recommender systems may be binary or numeric. Hence, randomization type must be determined considering the data set's structure. Besides protecting ratings of users from third parties due to various privacy risks, users also want to keep their number of ratings and their rated items private in order to avoid from disclosing their tendencies and interests. Also, since privacy concerns may differ from one user to another, users may want to disguise their confidential data differently. We have presented eight privacy frameworks for randomization-based privacy protection collaborative filtering systems. Our frameworks help researchers select appropriate privacy preserving method. They also form a basis for fair comparison of different privacy-preserving collaborative filtering schemes on randomization.

Our future research direction includes providing evaluation frameworks for privacy-preserving collaborative filtering schemes. Also, frameworks should be designed by considering privacy, accuracy, performance, and robustness.

Table 1. Ratings vectors and their disguised versions using the proposed frameworks

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
Numeric ratings vector	1	5	-	4	-	-	-	-	3	-
Framework 1	0.29	4.35	-	3.78	-	-	-	-	2.41	-
Framework 2	1.11	4.84	-	3.85	-	-	-	-	2.88	-
Framework 3	1.05	4.17	-	4.53	0.47	-	-	-	2.37	0.18
Framework 4	1.62	4.60	-	4.76	-	0.81	-	-	3.92	-
Binary ratings vector	0	1	-	1	-	-	-	-	0	-
Framework 5	0	1	-	1	-	-	-	-	1	-
Framework 6	1	0	-	0	-	-	-	-	0	-
Framework 7	0	1	1	1	-	-	-	-	0	0
Framework 8	1	0	-	0	1	-	-	-	0	-

Acknowledgements

This study is supported by the Grant 114E571 from TUBITAK.

References

1. Bobadilla J, Ortega F, Hernando A, Gutiérrez A. Recommender systems survey. *Knowledge-Based Systems* 2013; **46**:109-132.
2. Bilge A, Kaleli C, Yakut I, Gunes I, Polat H. A survey of privacy-preserving collaborative filtering schemes. *International Journal of Software Engineering and Knowledge Engineering* 2013; **23(8)**:1085-1108.
3. Ozturk A, Polat H. From existing trends to future trends in privacy-preserving collaborative filtering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2015; **5(6)**:276-291.
4. Agrawal R, Srikant R. Privacy-preserving data mining. *ACM SIGMOD Record* 2000; **29(2)**:439-450.
5. Polat H, Du W. Privacy-preserving collaborative filtering using randomized perturbation techniques. *Proceedings of the 3rd IEEE International Conference on Data Mining* 2003; Melbourne, FL, USA, pp. 625-628.
6. Polat H, Du W. SVD-based collaborative filtering with privacy. *Proceedings of the ACM Symposium on Applied Computing* 2007; Santa Fe, NM, USA, pp. 791-795.
7. Polat H, Du W. Privacy-preserving collaborative filtering. *International Journal of Electronic Commerce* 2005; **9(4)**:9-35.
8. Polat H, Du W. Effects of inconsistently masked data using RPT on CF with privacy. *Proceedings of the ACM Symposium on Applied Computing* 2007; Seoul, Korea, pp. 649-653.
9. Yakut I, Polat H. Achieving private SVD-based recommendations on inconsistently masked data. *Proceedings of the 1st International Conference on Security of Information and Networks* 2007; Gazimagusa, Cyprus, pp. 172-176.
10. Gong S. Privacy-preserving collaborative filtering based on randomized perturbation techniques and secure multiparty computation. *International Journal of Advancements in Computing Technology* 2011; **3(4)**:89-99.
11. Basu A, Vaidya J, Kikuchi H. Perturbation-based privacy-preserving Slope One predictors for collaborative filtering. *IFIP Advances in Information and Communication Technology* 2012; **374**:17-35.
12. Zhu T, Ren Y, Zhou W, Rong J, Xiong P. An effective privacy preserving algorithm for neighborhood-based collaborative filtering. *Future Generation Computer Systems* 2014; **36**:142-155.
13. Warner SL. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association* 1965; **60(309)**:63-69.
14. Polat H, Du W. Achieving private recommendations using randomized response techniques. *Lecture Notes in Computer Science* 2006; **3918**:637-646.
15. Kaleli C, Polat H. Providing private recommendations using naïve Bayesian classifier. *Advances in Soft Computing* 2007; **43**:168-173.
16. Kikuchi H, Mochizuki A. Privacy-preserving collaborative filtering using randomized response. *Journal of Information Processing* 2013; **21(4)**:617-623.
17. Cranor, LF. "I didn't buy it for myself" privacy and e-commerce personalization. *Proceedings of the ACM Workshop on Privacy in Electronic Society* 2003; Washington, DC, USA, pp. 111-117.
18. Bilge A, Polat H. A comparison of clustering-based privacy-preserving collaborative filtering schemes. *Applied Soft Computing* 2013; **13(5)**:2478-2489.
19. Bilge A, Polat H. A scalable privacy-preserving recommendation scheme via bisecting *k*-means clustering. *Information Processing & Management* 2013; **49(4)**:912-927.