World Wide Web
https://doi.org/10.1007/s11280-018-0620-z

Towards privacy preserving social recommendation under personalized privacy settings

Xuying Meng[1] · Suhang Wang[2] · Kai Shu[2] · Jundong Li[2] · Bo Chen[3] · Huan Liu[2] · Yujun Zhang[1,4]

Received: 18 December 2017 / Revised: 19 June 2018 / Accepted: 27 June 2018

Abstract Privacy leakage is an important issue for social relationships-based recommender systems (i.e., social recommendation). Existing privacy preserving social recommendation approaches usually allow the recommender to fully control users' information. This may be problematic since the recommender itself may be untrusted, leading to serious privacy leakage. Besides, building social relationships requires sharing interests as well as other private information, which may lead to more privacy leakage. Although sometimes users are allowed to hide their sensitive private data using personalized privacy settings, the data being shared can still be abused by the adversaries to infer sensitive private information. Supporting social recommendation with least privacy leakage to untrusted recommender and other users (i.e., friends) is an important yet challenging problem. In this paper, we aim to achieve privacy-preserving social recommendation under personalized privacy settings. We propose *PrivSR*, a novel privacy-preserving social recommendation framework, in which user can model user feedbacks and social relationships privately. Meanwhile, by allocating different noise magnitudes to personalized sensitive and non-sensitive feedbacks, we can protect

A preliminary version of this article was published in AAAI '18.

This article belongs to the Topical Collection: *Special Issue on Social Computing and Big Data Applications*
Guest Editors: Xiaoming Fu, Hong Huang, Gareth Tyson, Lu Zheng, and Gang Wang

✉ Yujun Zhang
mxyenguing@qq.com

Xuying Meng
mengxuying@ict.ac.cn

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

[2] Department of Computer Science, Arizona State University, Tempe, AZ, USA

[3] Department of Computer Science, Michigan Technological University, Houghton, MI, USA

[4] University of Chinese Academy of Sciences, Beijing, China

Published online: 14 July 2018

users' privacy against untrusted recommender and friends. Theoretical analysis and experimental evaluation on real-world datasets demonstrate that our framework can protect users' privacy while being able to retain effectiveness of the underlying recommender system.

Keywords Differential privacy · Social recommendation · Ranking · Personalized privacy settings

1 Introduction

A recommender system has become an imperative component of myriad online commercial platforms. With increasing popularity of social networks, recommender systems now can take advantage of these rich social relationships to improve recommendation effectiveness [34, 37, 43]. This new type of social relationships-based recommender system (i.e., *social recommendation* for short), however, suffers from a new source of privacy leakage. For example, by observing a victim user's feedbacks on products such as adult or medical items, the adversary may infer the victim's private sex inclination or health condition [8], and may further abuse the private information for financial benefits [29].

In practice, a privacy-preserving social recommender system, which can utilize social relationships to produce more accurate recommendation results without sacrificing privacy of users being involved, is very necessary. There were a few mechanisms designed for this purpose. However, they are all problematic as analyzed in the following. First, a few existing efforts [13, 22] heavily rely on an assumption that the recommender is fully trusted. They neglect the fact that the recommender itself may be untrusted and may conduct malicious behaviors, causing serious privacy leakage. Second, a few works [11, 38] rely on cryptography to prevent users' exact inputs from being leaked to the untrusted recommender. Nonetheless, it has been shown that attackers can still infer sensitive information about the victim users based on their influence on the final results [25]. In addition, the cryptographic process is usually expensive and may bring large computational overhead. Third, a few works [12, 13, 24] rely on friends' history feedbacks to make recommendations, but do not differentiate sensitive and non-sensitive feedbacks and simply treat them equally. In practice, social media sites such as IMDB and Facebook (Figure 1[1]) allow users to specify the visibility of their feedbacks on products. Treating all the feedbacks as equally sensitive and not exposing non-sensitive feedbacks for security, will make it difficult to attract common-interest friends and make effective recommendations, sacrificing user experience in the long run.

Resolving all the aforementioned defects is necessary for building an effective privacy-preserving social recommender system, which however is a very challenging task due to the following reasons: First, to relax the assumption that a recommender is fully trustful, we need to change the recommender system from a fully centralized manner to a semi-centralized manner. In other words, instead of fully relying on the recommender, we now allow users and the recommender to collaborate with each other for recommendation. Specifically, users can have access to both the sensitive and the non-sensitive feedbacks,

[1]Facebook provides public pages for products, e.g., https://www.facebook.com/pages/Google-Earth/107745592582048

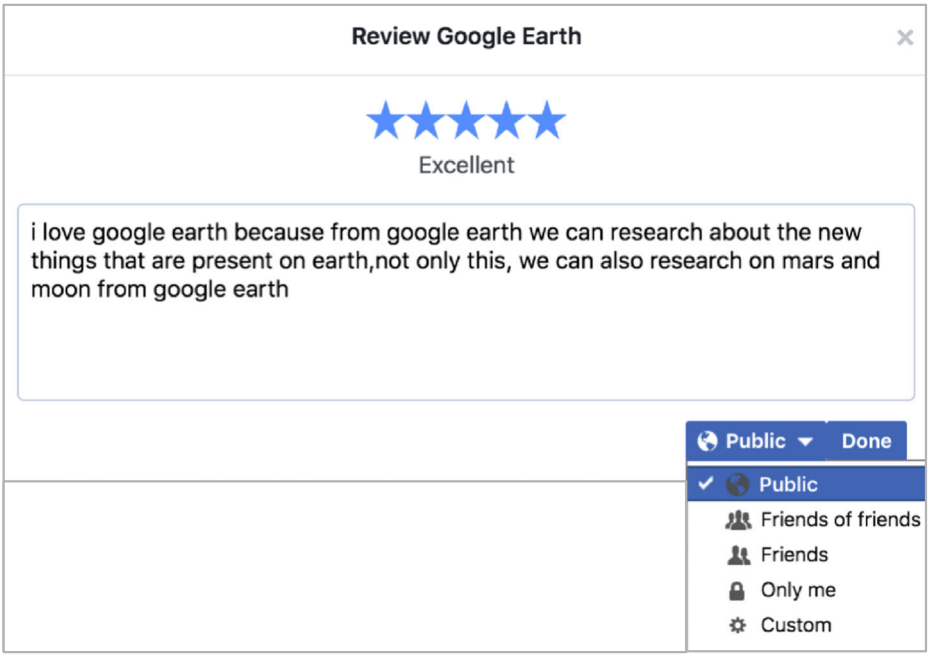


Figure 1 An example of personalized privacy settings from Facebook

while the recommender can only have access to the non-sensitive feedbacks, and they interact to make the final recommendation. In such a semi-centralized manner, private information may still be leaked during each interaction between the recommender and the user, and eliminating such leakage is necessary yet challenging. Second, to avoid using expensive cryptographic techniques, *differential privacy* [5] can be used to provide provable privacy guarantee with a small computational overhead. However, differential privacy requires adding noise which may degrade recommendation effectiveness. This will be exacerbated when the non-sensitive feedbacks are exposed and used as background knowledge to infer sensitive feedbacks. Third, users are usually allowed to configure their privacy settings in practice. Due to idiosyncrasy of different users, their personalized privacy settings could be quite diverse. Protecting sensitive feedbacks based on those personalized diversified privacy settings is not straightforward.

In this work, we initiate the study of privacy-preserving social recommendation based on personalized privacy settings. In particular, we propose a novel framework, *PrivSR*, that can protect sensitive feedbacks of users from being leaked to untrusted recommender and friends while retaining the effectiveness of recommendation. Our design is mainly based on matrix factorization-based social recommendation, a popular social recommendation approach. Our basic idea is three-fold: 1) We divide the learning process of user latent vectors into small components for each specific user, and utilize objective perturbation to provide privacy guarantee under differential privacy. 2) We categorize feedbacks into sensitive and non-sensitive feedbacks, and attach sensitive feedbacks with small privacy budgets,

i.e., big magnitude noises. In this way, the non-sensitive feedbacks' modeling will not be significantly affected, which can help retain recommendation effectiveness. 3) We decouple the components of noise perturbation into small pieces each of which can be independently processed by individual users. In this way, each user can decide his/her own noise magnitude locally.

There are not just explicit feedbacks like numerical ratings, but also implicit feedbacks like click, consuming and browse behaviors. We thus extend *PrivSR* to support implicit feedbacks with a flexible range of feedback values. In addition, there are not just exact preferences, i.e., ratings, but also relative preferences, i.e., rankings. Therefore, the extend *PrivSR* is designed based on ranking-based recommendation model, which can simultaneously protect sensitive feedbacks while maintaining social recommendation effectiveness.

Contributions We summarize the contributions in the following:

- We are the first to study the problem of privacy-preserving social recommendation under personalized privacy settings.
- We propose a novel social recommendation framework *PrivSR*. *PrivSR* works in a semi-centralized manner, and relies on differential privacy with well-balanced privacy budgets to handle untrusted recommender and friends while retaining recommendation effectiveness.
- We extend *PrivSR* to support ranking-based model with implicit feedbacks.
- We theoretically prove that both the *PrivSR* and the extended *PrivSR* can satisfy ϵ-differential privacy, and empirically validate their effectiveness using real-world datasets. The results are encouraging: both *PrivSR* and its extension provide a good balance between privacy protection and recommendation accuracy.

The rest of the paper is organized as follows. We outline the related work in Section 2 and introduce the background knowledge in Section 3. In Section 4, we formally define the problem. We describe the technical details of *PrivSR* and its extension in Section 5, and formally prove that the framework can achieve privacy guarantee in Section 6, respectively. In Section 7, we experimentally demonstrate the effectiveness of our framework on real-world datasets. We conclude in Section 8.

2 Related work

In this section, we summarize the related work, which includes privacy-preserving social recommendation and personalized privacy settings.

2.1 Privacy-preserving social recommendation

Most of the current privacy protection mechanisms for social recommender systems fully trust the recommender and hence all the data are disclosed to the recommender [13, 22, 24]. Their common goal is to ensure that the recommendation results generated by the recommender is non-sensitive to any single feedbacks or single social relationship such that any other users cannot infer whether the target feedbacks or social relationships are in the input or not.

Some attempts have been made to protect privacy leakage to untrusted recommender [12, 45], but they cannot fit in social recommendation as we require more exposure for social

regulation that may leak user privacy. For example, Xin et al. [45] were the first to deal with untrusted recommender with noise perturbation. However, their approach requires a group of public users to share all their feedbacks, which is not fulfilled in our scenario that offers all users privacy protection on their sensitive feedbacks. In addition, their approach is not provably secure. Hua et al. [12] proposed to protect users' feedbacks in the recommender systems. They require users to keep all their feedbacks locally, which is acceptable for the traditional recommender systems. However, it fails to handle new challenges brought by social recommendation: 1) how to locally utilize sensitive feedbacks to improve recommendation performance; 2) how to get users' similarity without privacy leakage; and 3) how to protect users' non-sensitive feedbacks from the exposure of sensitive feedbacks. To tackled these challenges, it requires social relationship involvement and well-balanced privacy design to make improvement on both recommendation effectiveness and privacy protection.

To utilize social relationships without privacy leakage under untrusted recommender, some works try to utilize cryptographic techniques [11, 38]. They prevent users' exact inputs from being leaked to the untrusted recommender, which provide no help for attacks we faced: 1) With inference attack, attackers can still infer users' exact feedbacks based on their influence on the final results [25]. 2) With reconstruction attack, the information the reconstruction attack required is the public feedbacks and final results, thus no matter how these approaches hide feedbacks, once the final results are accurate and they do not attach noise, the final privacy protection results will not be satisfactory. Thus, besides expensive cryptographic overheads, those works may still leak users' privacy since the untrusted recommender can still obtain the final accurate calculation results.

2.2 Personalized privacy settings

Personalized privacy preferences have received increasingly attention [10, 21, 22, 28]. Li et al. [21] proposed a semantics-based privacy configuration system to automatically recommend personalized privacy settings. Liu et al. [22] computed users' privacy score to indicate potential privacy risk caused by personalized privacy settings.

The majority of existing works mainly focus on recommending personalized privacy settings and evaluating privacy risks caused by inappropriate personalized privacy settings. However, little attention has been paid to handle the hidden potential privacy leakage from those personalized privacy settings [1]. Zhang et al. [1] attempted to implement an information-theoretic privacy-utility framework preventing inference attack on users' private attribute from released feedbacks, which can only protect users' private profile but ignore that some released feedbacks are sensitive. Moreover, attackers can conduct reconstruction attack by exploiting the released model to predict the sensitive input feedbacks of a target victim based on some background input (e.g. non-sensitive feedbacks) and the corresponding output from the model [8, 15]. In order to protect users' privacy, Fredrikson et al. showed that differential privacy mechanisms can prevent model inversion attacks when the privacy budget is very small [8]. However, the small privacy budget will significantly degrade the model's utility. Therefore, it is a great challenge to protect personalized privacy while retaining the effectiveness of social recommendation.

3 Background knowledge

This section outlines preliminary knowledge about social recommendation, differential privacy and potential privacy attacks.

3.1 Social recommendation

The matrix factorization model represents a matrix as the product of two lower-rank matrices [18, 32], which is widely used for recommendation. Let $\mathcal{U} = \{u_1, \ldots, u_n\}$ be a set of n users and $\mathcal{V} = \{v_1, \ldots, v_m\}$ be a set of m items. We denote the feedback from user u_i to item v_j as $\mathbf{R}_{ij}$. There are different kinds of feedbacks, such as explicit feedback (e.g., ratings) and implicit feedback (e.g., consume, click and browse etc.), which can be used to determine the range of $\mathbf{R}_{ij}$. If we observed a feedback from u_i to v_j, we set $\mathbf{I}_{ij} = 1$; otherwise $\mathbf{I}_{ij} = 0$. The matrix factorization objective function over the observed feedbacks can be written as

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{I}_{ij} (\mathbf{R}_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 + \lambda \Omega(\mathbf{U}, \mathbf{V}) \tag{1}$$

where $\mathbf{U} = [\mathbf{u}_i]_{i \in [n]} \in \mathbb{R}^{K \times n}$ and $\mathbf{V} = [\mathbf{v}_j]_{j \in [m]} \in \mathbb{R}^{K \times m}$ are latent matrix for users and items, K is the number of latent factors, λ is a scalar to control relative contribution, and $\Omega(\mathbf{U}, \mathbf{V})$ is the regularization term to avoid overfitting.

Social regularization represents the social constraints on recommender systems based on the assumption that every user's taste is close to the average taste of this user's friends [23]. Considering users' preferences are similar or influenced by their socially connected friends, social relationships are widely employed in designing social regularization term $\sum_{i=1}^{n} \sum_{f \in \mathcal{F}_i} S_{if} ||\mathbf{u}_i - \mathbf{u}_f||_F^2$ to constrain the matrix factorization objective function of recommender systems. We use $\mathcal{F}_i$ to represent the set of user u_i's friends. Then we can mathematically define the social recommendation algorithm over the observed social relationships and the observed feedbacks as:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{I}_{ij} (\mathbf{R}_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 + \alpha \sum_{i=1}^{n} \sum_{f \in \mathcal{F}_i} S_{if} ||\mathbf{u}_i - \mathbf{u}_f||_F^2 + \lambda \Omega(\mathbf{U}, \mathbf{V}) \tag{2}$$

where $|| \cdot ||_F^2$ denotes the Frobenius norm, S_{if} is the cosine similarity between feedbacks of u_i and u_f on the same items, and α is the scalar to control the contribution of social regularization.

3.2 Differential privacy

Differential privacy [5] is a popular privacy-preserving technique, which effectively perturbs the raw datasets by injecting noise and ensures that the output is not significantly affected by removal/addition of a single rating [2, 36]. Considering its provable privacy guarantee with light computational overhead, we will use differential privacy in our proposed framework.

Definition 1 ϵ-Differential Privacy [5]: A randomized algorithm f satisfies *ϵ-differential privacy*, if for any two datasets D_1 and D_2 which differ at most one rating, and for any possible anonymized output dataset $\tilde{D} \in Range(f)$,

$$Pr[f(D_1) = \tilde{D}] \leq e^{\epsilon} \times Pr[f(D_2) = \tilde{D}] \tag{3}$$

where $Range(f)$ denotes the output range of algorithm f.

The probability is taken over the randomness of f, and the privacy budget ϵ defines the magnitude of privacy being achieved, where ϵ is a positive real number and the smaller the ϵ, the harder to infer users' privacy.

Laplace mechanism [5] is commonly used to satisfy ϵ-differential privacy by adding i.i.d. noise from $Lap(GS(D)/\epsilon)$ to each output, where the global sensitivity $GS(D)$ is the maximal change to which any single rating in the input D can affect the output.

Considering the rare characteristics of Laplace distribution compared with normal distribution, researchers proposed an effective way [6, 19] to transfer it into the combination of exponential and normal distribution:

Lemma 1 *If a random number* $h \sim Exp(1)$*, a random number* $c \sim N(0, 1)$*, then for any real number* $b > 0$*, there is* $b\sqrt{2h}c \sim Lap(b)$.

3.3 Inference and reconstruction attack

Inference attack is always conducted to infer whether an individual rating is included in the training set [33], while differential privacy is widely used to defend against inference attack [25, 38, 46] by adding noise to perturb and reduce each individual's impact on the trained model.

Reconstruction attack is conducted to predict exact value of some sensitive features about a target victim based on some background information. A few existing works explored how to reconstruct model to predict users' sensitive information [9, 14, 15]. For example, Komarova et al. [15] attempted to infer the sensitive features of an individual given fixed statistical estimate from combined public and private sources. Fredrikson et al. [8] demonstrated that differential privacy mechanisms can mitigate reconstruction attacks only when the privacy budget is very small, which unfortunately will significantly degrade the effectiveness of the model. Wang et al. [41] were the first to propose to balance the utility and privacy from regression model based on functional mechanism [47].

However, the existing proposed mechanisms can not be applied to handle the reconstruction attack in social recommendation since the way to reconstruct the recommendation model is completely different, where the attackers can utilize non-sensitive ratings to inversely predict a victim user's latent features, reconstructing the user's sensitive ratings by matrix factorization [18].

4 Problem statement

In social recommendation, there are three types of actors, namely, users, friends and recommender. Among them, the friends and the recommender may be not fully trusted (more strictly speaking, semi-trusted, since they are curious about users' sensitive feedbacks but still honestly follow the protocol to provide recommendation).

We use a concrete example (as shown in Figure 2) to show some potential privacy leakage, where we represent user feedbacks as ratings although it's the same case for other kinds of feedbacks. To model history ratings in matrix factorization-based social recommendation, the victim user u_1 is required to share some processed outputs with the recommender

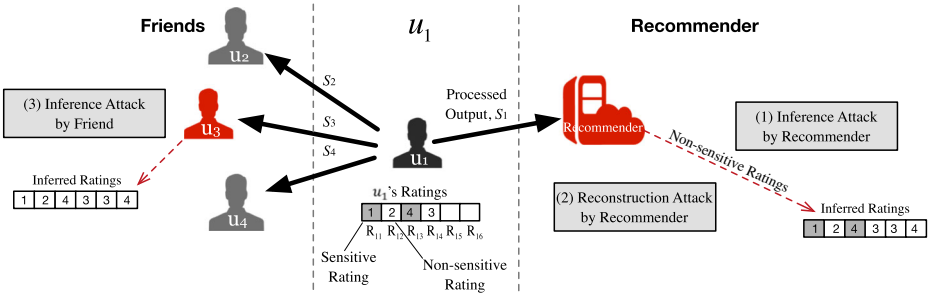


Figure 2 The figure gives an example of user's privacy attacks in social recommendation from the perspective of victim user u_1. Assume there's six items, u_1 has rated four with personalized sensitivity. In order to learn social recommendation model, u_1 exposes processed outputs S_1, S_2, S_3 and S_4 to the recommender and to his friends u_2, u_3 and u_4. The black arrows show the exposure directions. However, attackers, who are colored red, utilize the exposed information to conduct attacks to obtain u_1's sensitive ratings. The attacks are shown in gray boxes with utilized information. For example, u_3 utilize $\mathbf{u}_1$ and $\mathbf{V}$ to conduct inference attack. We also color the process of attacks with red dashed arrows

and friends u_2, u_3, u_4. However, the attackers can manage to learn sensitive information from the exposed outputs in the learning process: (1) To update $\mathbf{v}_j$, the recommender requires user u_1 who has rated item v_j to share S_1, which is calculated by a rating $\mathbf{R}_{1j}$, user latent vector $\mathbf{u}_1$ and item latent matrix $\mathbf{V}$. However, with a non-sensitive rating $\mathbf{R}_{1j}$, the recommender can directly compute an accurate $\mathbf{u}_1$, which then further leaks sensitive ratings $\mathbf{R}_{11}$ and $\mathbf{R}_{13}$ by $\mathbf{u}_i^T\mathbf{V}$; (2) With the exposed non-sensitive ratings, the recommender can conduct reconstruction attack to infer a proximate latent vector $\tilde{\mathbf{u}}_1$, by which u_1's all ratings may be disclosed by computing $\tilde{\mathbf{u}}_i^T\mathbf{V}$; and (3) The malicious friend u_3 requires user latent vector $\mathbf{u}_1$ for social regularization, by which u_3 may learn u_1's ratings by $\mathbf{u}_1$ and $\mathbf{V}$.

To formally define our problem, we first describe the notations used in this paper. When the user u_i give feedbacks to item v_j (i.e., $\mathbf{R}_{ij}$), u_i will specify his/her privacy setting on $\mathbf{R}_{ij}$ as private, sharing within friends, or public. We use $\mathbf{F}_{ij} = 1$ to indicate that $\mathbf{R}_{ij}$ is a *sensitive feedback*, and only visible to user u_i due to privacy concerns; otherwise $\mathbf{F}_{ij} = 0$. Similarly, $\mathbf{G}_{ij} = 1$ indicates that $\mathbf{R}_{ij}$ is a *non-sensitive feedback*, and visible to friends/public; otherwise $\mathbf{G}_{ij} = 0$. As $\mathbf{F}_{ij} = 1$ and $\mathbf{G}_{ij} = 1$ are mutually exclusive, we have $\mathbf{I}_{ij} = \mathbf{F}_{ij} + \mathbf{G}_{ij}$ for all observed feedbacks. Then we define the set of sensitive feedbacks as $\mathcal{R}_s = \{\mathbf{R}_{ij} | \forall(i, j)\ s.t.\ \mathbf{F}_{ij} = 1\}$, and the set of non-sensitive feedbacks as $\mathcal{R}_n = \{\mathbf{R}_{ij} | \forall(i, j)\ s.t.\ \mathbf{G}_{ij} = 1\}$. With these definitions, our privacy-preserving social recommendation problem can be formally defined as:

> Given the observed values in $\mathbf{R}$, the set of friends $\mathcal{F}$, a set of sensitive feedbacks $\mathcal{R}_s$, as well as a set of non-sensitive feedbacks $\mathcal{R}_n$, we want to infer the missing values in $\mathbf{R}$ without privacy leakage of $\mathcal{R}_s$.

5 Privacy-preserving social recommendation

Our proposed framework, *PrivSR*, aims to allow recommender systems to incorporate social relationships without leaking sensitive feedbacks to untrusted recommender and friends.

To achieve this goal, we perform the following: First, we incorporate social relationships into traditional recommender systems with consideration of both non-sensitive and sensitive feedbacks. We divide the entire framework into *users' feedbacks* component and *social relationships* component (Figure 3), and keep balanced noise perturbation on sensitive and non-sensitive feedbacks in users' feedbacks component, and meanwhile, only utilize non-sensitive feedbacks to model social similarity with untrusted friends in the social relationships component. Second, to remove the centralized control of the untrusted recommender or any third parties, we involve users during the recommendation, making the recommender and users collaborate to perform recommendation, such that processing of sensitive data can be performed by users' side, and remaining processing are left to be performed by the recommender's side. We allocate different resources to the recommender and individual users as shown in the green part of Figure 3, in which the recommender can only have access to non-sensitive feedbacks $\mathcal{R}_n$ and share the updated item latent matrix $\mathbf{V}$ with everyone for recommendation purpose. Except public information, every user holds his/her private information, including all his/her feedbacks $\mathbf{R}_i$ and friends set $\mathcal{F}_i$, in local machines (which can be mobile phones or personal computers). In particular, since the user latent vector $\mathbf{u}_i$ can be used to obtain sensitive feedbacks (e.g., by computing $\mathbf{u}_i^T\mathbf{V}$), $\mathbf{u}_i$ should be also kept locally.

5.1 Modeling sensitive and non-sensitive feedbacks

A non-trivial task for our *PrivSR* design is to model feedbacks without leakage of sensitive feedbacks, especially in face of personalized privacy settings and public non-sensitive feedbacks, which may be used by the adversary as the background information to infer the sensitive feedbacks. We present the basic model based on matrix factorization model [18, 27, 40]. Based on (1) and $\mathbf{I}_{ij} = \mathbf{F}_{ij} + \mathbf{G}_{ij}$, the objective function can be written as follows:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n}\sum_{j=1}^{m}(\mathbf{F}_{ij}+\mathbf{G}_{ij})(\mathbf{R}_{ij}-\mathbf{u}_i^T\mathbf{v}_j)^2 \tag{4}$$

To conduct recommendation in a semi-central manner and protect privacy from untrusted recommender, we utilize gradient descent to decouple and update each latent vector $\mathbf{u}_i$ of each user. Because the gradient of (4) w.r.t. $\mathbf{u}_i$ is $\sum_{j=1}^{m} 2(\mathbf{F}_{ij}+\mathbf{G}_{ij})(\mathbf{u}_i^T\mathbf{v}_j-\mathbf{R}_{ij})\mathbf{v}_j$, which only involves $\mathbf{u}_i$ and $\mathbf{V}$, then each $\mathbf{u}_i$ can be updated locally with the shared $\mathbf{V}$, and can be kept private.

On the other hand, to update $\mathbf{v_j}$, the gradient of (4) w.r.t. $\mathbf{v}_j$ is $\sum_{i=1}^{n} 2(\mathbf{F}_{ij}+\mathbf{G}_{ij})(\mathbf{u}_i^T\mathbf{v}_j-\mathbf{R}_{ij})\mathbf{u}_i$, which requires each user (e.g., u_i) who has rated v_j to submit a copy of $\sigma_j^i = 2(\mathbf{F}_{ij}+\mathbf{G}_{ij})(\mathbf{u}_i^T\mathbf{v}_j-\mathbf{R}_{ij})\mathbf{u}_i$ to the recommender, whereas the individual submission may raise great privacy concerns: Attackers can directly obtain the accurate $\mathbf{u}_i$ when $\mathbf{G}_{ij} = 1$, then all sensitive feedbacks are exposed by $\mathbf{u}_i^T\mathbf{V}$. Although encryption techniques may solve this problem and ensure the recommender only knows the final summation but not the exact value from each user, the untrusted recommender can still conduct inference attack from the contribution of a particular user u_i, and obtain the accurate $\mathbf{u}_i$ [25]. To tackle all these problems, we apply the objective perturbation method [3] with ϵ-differential privacy, and perturb individual's involvement by adding noise into the objective function. We then introduce noise to (4) as:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n}\sum_{j=1}^{m}\left((\mathbf{F}_{ij}+\mathbf{G}_{ij})(\mathbf{R}_{ij}-\mathbf{u}_i^T\mathbf{v}_j)^2+\mathbf{v}_j^T\mathbf{o}_j^i\right) \tag{5}$$

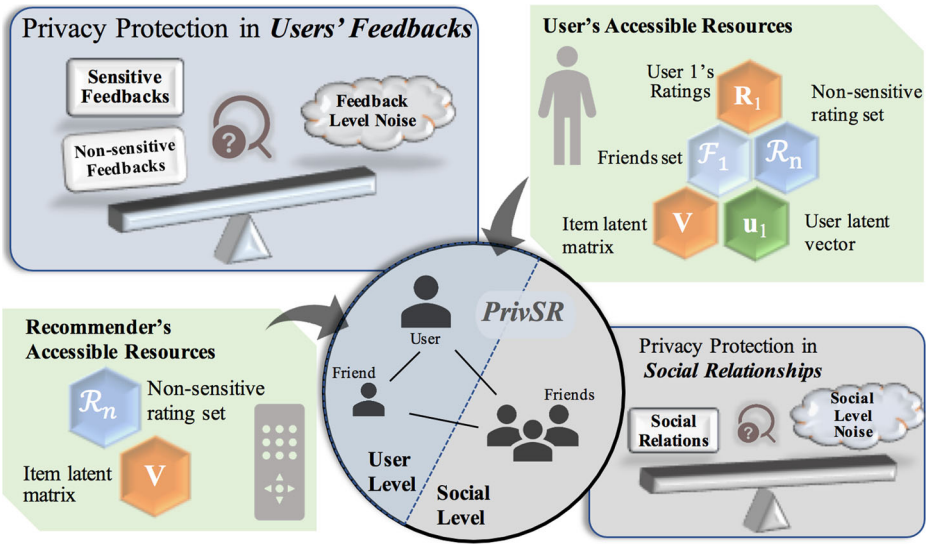


Figure 3 The proposed framework – *PrivSR*

where $\mathbf{o}_j = \sum_i \mathbf{o}_j^i \in \mathbb{R}^{K\times 1}$ is a noise vector, and each user u_i protects σ_j^i by adding $\mathbf{o}_j^i$ in the derivative w.r.t. $\mathbf{v}_j$.

Then there comes the second privacy concern that attackers can still obtain proximate users' sensitive feedbacks with the exposed non-sensitive feedbacks by performing reconstruction attack. This can be prevented only when privacy budget ϵ for noise sampling is extremely small [8], whereas, small privacy budgets will lead to large noise magnitude and the recommendation effectiveness will degrade. Thus the unified noise $\mathbf{o}_j$ without considering personalized privacy settings, will definitely reduce the effectiveness of recommendation. To protect users' privacy while retaining recommendation effectiveness, we allocate balanced privacy budgets for sensitive and non-sensitive feedbacks as:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} & \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{F}_{ij}\left((\mathbf{R}_{ij}-\mathbf{u}_i^T\mathbf{v}_j)^2+\mathbf{v}_j^T\mathbf{x}_j^i\right)\\ + & \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{G}_{ij}\left((\mathbf{R}_{ij}-\mathbf{u}_i^T\mathbf{v}_j)^2+\mathbf{v}_j^T\mathbf{y}_j^i\right)\end{aligned} \tag{6}$$

where $\mathbf{x}_j = \sum_i \mathbf{x}_j^i \in \mathbb{R}^{K\times 1}$, $\mathbf{y}_j = \sum_i \mathbf{y}_j^i \in \mathbb{R}^{K\times 1}$ are noise vectors for $\sum_i \sigma_j^i$ with sensitive and non-sensitive feedbacks respectively. We allocate a much smaller privacy budget ϵ_s for sensitive feedbacks and a larger ϵ_n for non-sensitive ones where $\epsilon_s = \beta\epsilon_n$ and the domain of β is (0, 1] which is used to control the relative noise magnitude. Then sensitive feedbacks can receive better privacy protection with the small privacy budget ϵ_s. We set the privacy budget of the derived $\mathbf{V}$ as $\epsilon = \frac{\beta\epsilon_n}{1+\beta}$.

Since $\epsilon_n > \epsilon_s$, Theorem 1 shows that our model can effectively protect sensitive feedbacks while retaining recommendation effectiveness with balanced privacy budgets. However, it is difficult for users to independently select $\mathbf{y}_j^i$ and achieve $\sum_i \mathbf{y}_j^i \sim Lap(2\Delta\sqrt{K}/\epsilon_n)$. It is similar for $\mathbf{x}_j^i$, and we use $\mathbf{y}_j^i$ as an example. Although the sum of numbers from Laplace distribution does not follow Laplace distribution anymore, the summation of numbers from normal distribution can still follow normal distribution. According to Lemma 1, the recommender first construct $\mathbf{h}_j \in \mathbb{R}^K$, where each element of $\mathbf{h}_j$ is randomly and independently picked from $Exp(1)$. Then the recommender shares $\mathbf{h}_j$ to users in $\mathcal{R}_{n,j}$, where we define $\mathcal{R}_{n,j}$ (or $\mathcal{R}_{s,j}$) as the set of users who gave v_j non-sensitive (or sensitive) feedbacks. After that, each user selects $\mathbf{c}_{ij_n} \in \mathbb{R}^K$, where each element in $\mathbf{c}_{ij_n}$ is randomly and independently picked from $N(0, 1/|\mathcal{R}_{n,j}|)$. Then σ_j^i can be protected using noise $2\Delta\sqrt{2K\mathbf{h}_j}\mathbf{c}_{ij_n}/\epsilon_n$ based on $\mathbf{h}_j$ and $\mathbf{c}_{ij_n}$, and the summation of noise $\sum_{i\in\mathcal{R}_{n,j}}(2\Delta\sqrt{2K\mathbf{h}_j}\mathbf{c}_{ij_n}/\epsilon_n) \sim Lap(2\Delta\sqrt{K}/\epsilon_n)$.

Theorem 1 *Let Δ denotes the difference between the maximal rating and the minimum rating. If each element in $\mathbf{x}_j$ and $\mathbf{y}_j$ is independently and randomly selected from $Lap(\frac{2\Delta\sqrt{K}}{\epsilon_s})$ and $Lap(\frac{2\Delta\sqrt{K}}{\epsilon_n})$, the derived $\mathbf{V}$ satisfies ϵ-differential privacy.*

Proof See Section 6.1 for the detailed proof. □

User involvement Note that the user only needs to decide if a rating is sensitive or not (i.e., a simple 0 or 1 decision). After receiving the user' decision, the local machine will take care of calculating the corresponding privacy budget (ϵ_s or ϵ_n) based on the user's decision. The user interface should be simple rather than complicate. In addition, for a rational user, we believe the decision should be straightforward, rather than "chaotic".

5.2 Modeling social relationships

Social relationships can be formulated as $\sum_{i=1}^n \sum_{f\in\mathcal{F}_i} S_{if}||\mathbf{u}_i - \mathbf{u}_f||_F^2$ [23], which requires calculating similarity S_{if} for all the sensitive and non-sensitive feedbacks between user u_i and u_f, and exchanging friends' very sensitive information, i.e., latent vectors $\mathbf{u}_f$. Without a fully trusted recommender, this sensitive information may be leaked in the course of optimization.

To protect sensitive feedbacks from untrusted friends, we only utilize non-sensitive feedbacks for the calculation of S_{if}. Also, to protect each friend's $\mathbf{u}_f$ from the optimization of $\sum_{i=1}^n \sum_{f\in\mathcal{F}_i} S_{if}||\mathbf{u}_i - \mathbf{u}_f||_F^2$ with gradient descent, we first calculate the gradient w.r.t $\mathbf{u}_i$ as $2S_{if}\mathbf{u}_i - \sum_{f\in\mathcal{F}_i} 2S_{if}\mathbf{u}_f$, where we set $\sigma_i^f = 2S_{if}\mathbf{u}_f$. To protect friends from sharing $\mathbf{u}_f$ to user u_i, we also propose the perturbation terms to hide friends' user latent vector $\mathbf{u}_f$

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^n \sum_{f\in\mathcal{F}_i} \left(S_{if}||\mathbf{u}_i - \mathbf{u}_f||_F^2 + \mathbf{u}_i^T \mathbf{q}_i^f \right) \tag{7}$$

where $\mathbf{q}_i = \sum_f \mathbf{q}_i^f \in \mathbb{R}^{K\times 1}$ is the noise vector, and each $\mathbf{q}_i^f$ is from friend u_f for derived $\mathbf{u}_i$. In order to make u_f help his friend u_i locally to learn $\mathbf{u}_i$ while not leaking $\mathbf{u}_f$ from the submission of σ_i^f, we add noise in (7). In this way, each friend can send the perturbed value $\mathbf{q}_i^f - \sigma_i^f$ to user u_i.

Theorem 2 ensures $\sum_f \mathbf{q}_i^f \sim Lap(2\sqrt{K}/\epsilon)$, thus we demand each user constructs $\mathbf{h}_i$ from $Exp(1)$, and shares $\mathbf{h}_i$ with all his/her friends. All the friends will also randomly and independently select $\mathbf{c}_{if}$ from $N(0, 1/|\mathcal{F}_i|)$. Then σ_i^f can be protected by noise $2\sqrt{2K\mathbf{h}_i}\mathbf{c}_{if}/\epsilon$, and the summation of noise $\sum_{f\in\mathcal{F}_i}(2\sqrt{2K\mathbf{h}_i}\mathbf{c}_{if}/\epsilon) \sim Lap(2\sqrt{K}/\epsilon)$.

Theorem 2 *If each element in $\mathbf{q}_i$ is independently and randomly selected from $Lap(\frac{2\sqrt{K}}{\epsilon})$, the derived U satisfies ϵ-differential privacy.*

Proof See Section 6.2 for the detailed proof. □

5.3 The proposed framework–*PrivSR*

To protect users' privacy from untrusted recommender with sensitive and non-sensitive model component, and from untrusted friends with social relationships model component, the final objective function of *PrivSR* to protect sensitive feedbacks while retaining recommendation effectiveness is to solve the following optimization problem:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} \mathcal{J} = & \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{F}_{ij}\left((\mathbf{R}_{ij}-\mathbf{u}_i^T\mathbf{v}_j)^2+\mathbf{v}_j^T\mathbf{x}_j^i\right)\\ & +\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{G}_{ij}\left((\mathbf{R}_{ij}-\mathbf{u}_i^T\mathbf{v}_j)^2+\mathbf{v}_j^T\mathbf{y}_j^i\right)\\ & +\alpha\sum_{i=1}^{n}\sum_{f\in\mathcal{F}_i}\left(S_{if}||\mathbf{u}_i-\mathbf{u}_f||_F^2+\mathbf{u}_i^T\mathbf{q}_i^f\right)\\ & +\lambda(||\mathbf{U}||_F^2+||\mathbf{V}||_F^2)\end{aligned} \tag{8}$$

where α and λ are scalars to control relative contribution, and $||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2$ is the regularization term to prevent overfitting. We use gradient descent to minimize the objective function. The gradients of (8) w.r.t. $\mathbf{u}_i$ and $\mathbf{v}_j$ are given as follows:

$$\begin{aligned}\frac{\partial\mathcal{J}}{\partial\mathbf{v}_j} = & \sum_{i=1}^{n}\mathbf{F}_{ij}\left(2(\mathbf{u}_i^T\mathbf{v}_j-\mathbf{R}_{ij})\mathbf{u}_i+\mathbf{x}_j^i\right)\\ & +\sum_{i=1}^{n}\mathbf{G}_{ij}\left(2(\mathbf{u}_i^T\mathbf{v}_j-\mathbf{R}_{ij})\mathbf{u}_i+\mathbf{y}_j^i\right)+2\lambda\mathbf{v}_j\end{aligned} \tag{9}$$

$$\begin{aligned}\frac{\partial\mathcal{J}}{\partial\mathbf{u}_i} = & 2\sum_{j=1}^{m}\mathbf{I}_{ij}(\mathbf{u}_i^T\mathbf{v}_j-\mathbf{R}_{ij})\mathbf{v}_j+2\alpha\sum_{f\in\mathcal{F}_i}S_{if}(\mathbf{u}_i-\mathbf{u}_f)\\ & +\alpha\sum_{f\in\mathcal{F}_i}\mathbf{q}_i^f+2\lambda\mathbf{u}_i\end{aligned} \tag{10}$$

Algorithm 1 *PrivSR* Algorithm

Input: $\mathcal{J}$, ϵ, γ, β, λ, observed feedbacks $\mathcal{R}$, user u_i holds its $\mathbf{u}_i$ and $\mathcal{F}_i$
Output: $\mathbf{U}$, $\mathbf{V}$

1: Initialize $\mathbf{U}$ and $\mathbf{V}$
2: **while** not converge **do**
3: **for** $j = 1, ..., m$ **do**
4: // *Calculate* $\mathbf{v}_j$ *on recommender's side*
5: **for** i *in* $\mathcal{R}_{s,j}$ **do**
6: Obtain $2(\mathbf{u}_i^T \mathbf{v}_j - \mathbf{R}_{ij})\mathbf{u}_i + \mathbf{x}_j^i$ from user u_i
7: **end for**
8: **for** i *in* $\mathcal{R}_{n,j}$ **do**
9: Obtain $2(\mathbf{u}_i^T \mathbf{v}_j - \mathbf{R}_{ij})\mathbf{u}_i + \mathbf{y}_j^i$ from user u_i
10: **end for**
11: Update $\mathbf{v}_j$ as $\mathbf{v}_j = \mathbf{v}_j - \gamma \frac{\partial \mathcal{J}}{\partial \mathbf{v}_j}$
12: **end for**
13: **for** $i = 1, ..., n$ **do**
14: // *Calculate* $\mathbf{u}_i$ *on user* u_i*'s side*
15: **for** f in $\mathcal{F}_i$ **do**
16: Obtain $\mathbf{q}_i^f - 2S_{if}\mathbf{u}_f$ from user u_f
17: **end for**
18: Update $\mathbf{u}_i$ as $\mathbf{u}_i = \mathbf{u}_i - \gamma \frac{\partial \mathcal{J}}{\partial \mathbf{u}_i}$
19: **end for**
20: **end while**

To address the challenge of protecting sensitive feedbacks against untrusted recommender and friends, we conduct objective perturbation with balanced privacy budgets in a semi-centralized way, which is described in Algorithm 1. To preserve privacy, item latent matrix is updated in the recommender's side with perturbed information from users, and user latent vectors are updated in each user's side individually with shared $\mathbf{V}$ and perturbed friends' user latent vectors. Next, we briefly describe Algorithm 1. In order to help the recommender to update $\mathbf{v}_j$ in lines 4 through 11 with (9), users send the recommender the required information individually with different privacy budget ϵ_n or ϵ_s. To help user u_i update $\mathbf{u}_i$ in lines 14 through 18 with (10), each of u_i's friends sends perturbed results with independent and random noise $\mathbf{q}_i^f$. After the algorithm converges, we can obtain the predicted result $\widehat{\mathbf{R}}$ by the optimized $\mathbf{U}$ and $\mathbf{V}$.

During the learning process, the time complexity for both the recommender and individual users are acceptable. In each iteration of the learning process: 1) on the recommender's side, it is $O(K|\mathcal{R}|)$ to learn the item latent matrix, where $|\mathcal{R}|$ is the number of observed feedbacks; 2) on each user's side, it is $O(K|\mathcal{R}_{i*}|)$ to learn the user latent vector, where $|\mathcal{R}_{i*}|$ denotes the number of feedbacks from user u_i, and it is $O(K|\mathcal{F}_i|)$ to help optimize the user latent vector, where $|\mathcal{F}_i|$ is the number of friends for user u_i. To decrease the complexity, we can use sub-gradient descent. Thus, in the whole learning process, the time complexity is $O(NK|\mathcal{R}|/M)$ for the recommender, and $O(NK(|\mathcal{R}_{i*}| + |\mathcal{F}_i|)/M)$ for each user, where N is the number of iterations, M is the number of batches.

Note that statistical information from users' submission in each iteration may be utilized by attackers. For example, to obtain a targeted sensitive rating $\mathbf{R}_{ij}$, the untrusted recommender can collect $\tilde{\sigma}_j^i(t) = \sigma_j^i(t) + \mathbf{x}_j^i$ in t-th iteration, where $\sigma_j^i(t) = 2\mathbf{F}_{ij}(\mathbf{u}_i^T(t)\mathbf{v}_j(t) -$

$\mathbf{R}_{ij})\mathbf{u}_i(t)$. Based on $\tilde{\sigma}^i_j(t) - \tilde{\sigma}^i_j(t-1) = \sigma^i_j(t) - \sigma^i_j(t-1)$, the impact of noise is eliminated. Therefore, we need to ensure $\mathbf{x}^i_j$ is randomly sampled in each iteration to eliminate the influence of statistics [30]. Similarly, $\mathbf{y}^i_j$ and $\mathbf{q}^f_i$ will also be updated in each iteration. Theorem 3 confirms us that *PrivSR* can achieve the desired security. After Algorithm 1 converges, our model can satisfy ϵ-differential privacy against untrusted recommender and friends.

Theorem 3 *PrivSR can satisfy ϵ-differential privacy.*

Proof See Section 6.3 for the detailed proof. □

Discussion *PrivSR* is a static recommender system which does not involve time dimension. We slightly discuss the dynamic recommender system which involves time dimension here. For such a dynamic recommender system, we have two cases: (1) There are temporal correlations of ratings among different time slots. We do not discuss this case further due to its high complication. (2) There are no temporal correlations of ratings among different time slots. We further analyze this case as follows. To update latent vectors based on ϵ_t-differential privacy, we can either utilize the new ratings in the t-th time slot or repeatedly utilize the entire dataset (i.e., from time slot 0 to t). If we only utilize the new ratings, the computation results on the disjoint inputs can satisfy $max\epsilon_t$-differential privacy based on parallel composition [25]. If we repeatedly utilize the entire dataset to update latent vectors, the computation results on non-disjoint inputs can satisfy $(\sum_t \epsilon_t)$-differential privacy based on sequential composition [25]. This may have better recommendation effectiveness but higher computation complexity and privacy leakage.

5.4 Extend *PrivSR* to support ranking-based recommendation

User preferences on items usually include exact preferences, i.e., ratings, and relative preferences, i.e., rankings [20, 48]. Also, since feedbacks include explicit feedbacks (e.g., ratings) and implicit feedbacks (e.g., consume, click, browse, etc.), the ranking-based recommendation model only requires relative preferences which makes it naturally support both types of feedbacks. As the *PrivSR* is designed for rating-based recommendation model, rather than ranking-based model, therefore, we further extend *PrivSR* to support the ranking-based model.

It is challenging to extend *PrivSR* to the ranking-based model, because: in the ranking-based model, items can be classified into different types in order to express users' relative preferences on these items. For example, there are 4 items v_1, v_2, v_3 and v_4, in which u_1 has consumed v_1 (or gave feedbacks to v_1) and u_1's friend u_2 has consumed v_2, thus the relative preferences of u_1 on these items should be $v_1 > v_2 > v_3$ (or v_4). To summarize, we have three types of items in a ranking-based recommendation model: items with feedbacks from users themselves (e.g., v_1), items with feedbacks from friends only (e.g., v_2), and items without any feedbacks from both friends and users themselves (e.g., v_3 and v_4). Therefore, unlike the rating-based recommendation model which only contains items from users themselves, we can simply classified the item as sensitive and non-sensitive based on the feedbacks; in the rankings-based model however, besides the items with feedbacks from users themselves, we also need to consider the other two types of items, which requires more complicate noise perturbation to simultaneously satisfy differential privacy and retain recommendation effectiveness.

For items with feedbacks from users themselves, the feedbacks can be classified as sensitive and non-sensitive. For other two types of items, the feedbacks are all non-sensitive. The sensitive feedbacks can be easily handled with privacy consideration following the idea in *PrivSR*. However, the non-sensitive feedbacks in the three types of items are from different preference ranges which will affect privacy protection during the ranking modeling process. Our idea for resolving the aforementioned issue is to normalize the relative preferences on the three types of non-sensitive feedbacks into the same range. In this way, we can attach them with noises of a unified global sensitivity to simplify the entire model.

Our extended *PrivSR* is built on top of SBPR [48], one of the most popular ranking-based social recommender system.[2] Based on the assumption that, for a given user u_i, items u_i has consumed (denoted as set P_i) should be ranked higher than items u_i's friends have consumed (denoted as set F_i), and items u_i's friends have consumed should be ranked higher than items neither u_i nor u_i's friends have consumed (denoted as set N_i), we then define the set of triples $D_{PF} = \{(i, j, k)|j \in P_i \wedge k \in F_i\}$, and $D_{FN} = \{(i, j, k)|j \in F_i \wedge k \in N_i\}$. The objective function $\mathcal{J}_1$ of ranking for social recommendation can be formalized as

$$\max_{\mathbf{U},\mathbf{V}} \sum_{(i,j,k)\in D_{PF}} \ln \phi(\mathbf{u}_i^T \mathbf{v}_j - \mathbf{u}_i^T \mathbf{v}_k) + \sum_{(i,j,k)\in D_{FN}} \ln \phi(\mathbf{u}_i^T \mathbf{v}_j - \mathbf{u}_i^T \mathbf{v}_k) + \lambda \Omega(\mathbf{U}, \mathbf{V}) \quad (11)$$

which utilizes ranking form objective function to model relative preferences based on social relationships. Since the range of sigmoid function $\frac{1}{1+e^{-x}}$ and the range of sigmoid function's derivative is (0, 1) no matter what x is, then we define $\phi(x) = \frac{1}{1+e^{-x}}$ to simplify the later noise perturbation.

Since we will face the similar attacks as shown in Figure 2, to protect privacy in the learning process of $\mathbf{U}$ and $\mathbf{V}$, we will adopt the semi-centralized framework as Figure 3 for ranking in social recommendation, where the recommender can only have access to $\mathbf{V}$ and non-sensitive feedbacks $\mathcal{R}_n$, and users locally preserve their individual $\mathbf{u}_i$ and conduct recommendation with shared $\mathbf{V}$. We first obtain the gradient of (11) w.r.t. $\mathbf{u}_i$ as

$$\frac{\partial \mathcal{J}_1}{\partial \mathbf{u}_i} = \sum_{(i,j,k)\in D_{PF}} \psi_{ijk}(\mathbf{v}_j - \mathbf{v}_k) + \sum_{(i,j,k)\in D_{FN}} \psi_{ijk}(\mathbf{v}_j - \mathbf{v}_k) \quad (12)$$

where $\psi_{ijk} = \frac{1}{1+e^{(\mathbf{u}_i^T \mathbf{v}_j - \mathbf{u}_i^T \mathbf{v}_k)}}$, and (12) only involves $\mathbf{u}_i$ and $\mathbf{V}$, then each $\mathbf{u}_i$ can be updated locally and privately, just like what we have done for (4). Similarly, the gradient of (11) w.r.t. $\mathbf{v}_j$ is

$$\begin{array}{rl} \frac{\partial \mathcal{J}_1}{\partial \mathbf{v}_j} = & \sum_{(i,j,k)\in D_{PF}} \psi_{ijk}\mathbf{u}_i - \sum_{(i,k,j)\in D_{PF}} \psi_{ikj}\mathbf{u}_i \\ + & \sum_{(i,j,k)\in D_{FN}} \psi_{ijk}\mathbf{u}_i - \sum_{(i,k,j)\in D_{FN}} \psi_{ikj}\mathbf{u}_i \end{array} \quad (13)$$

where each user will choose the corresponding terms according to how they treat item v_j based on P_i, F_i and N_i, i.e., for users who have consumed v_j, they would submit $\sum_{i:(i,j)\in P_i} \sigma_j^{iP} = \sum_{(i,j,k)\in D_{PF}} \psi_{ijk}\mathbf{u}_i$; for users who have friends have consumed v_j, they would submit $\sum_{i:(i,j)\in F_i} \sigma_j^{iF} = -\sum_{(i,k,j)\in D_{PF}} \psi_{ikj}\mathbf{u}_i + \sum_{(i,j,k)\in D_{FN}} \psi_{ijk}\mathbf{u}_i$; and for users neither themselves nor their friends have consumed v_j, they would submit $\sum_{i:(i,j)\in N_i} \sigma_j^{iN} = -\sum_{(i,k,j)\in D_{FN}} \psi_{ikj}\mathbf{u}_i$. Since sensitive feedbacks are only visible to user themselves, F_i only contains items which u_i's friends regard as non-sensitive. Based on Sections 5.1 and 5.2, we can propose similar framework of the extended *PrivSR* to protect

[2]The idea presented in the extended *PrivSR* can also be applied to other ranking-based social recommender systems [42] after slight modifications.

privacy for SBPR. Thus for a user u_i, to protect his/her sensitive feedbacks in P_i, we will propose the objective function of the extended *PrivSR* with perturbed noises as,

$$\begin{aligned}\max_{\mathbf{U},\mathbf{V}} \mathcal{J}_2 = & \sum_{(i,j,k)\in D_{PF}} \mathbf{F}_{ij}(\ln\phi(\mathbf{u}_i^T\mathbf{v}_j - \mathbf{u}_i^T\mathbf{v}_k) + \mathbf{v}_j^T\mathbf{x'}^i{}_j) + \lambda\Omega(\mathbf{U},\mathbf{V}) \\ & + \sum_{(i,j,k)\in D_{PF}} \mathbf{G}_{ij}(\ln\phi(\mathbf{u}_i^T\mathbf{v}_j - \mathbf{u}_i^T\mathbf{v}_k)) \\ & + \sum_{(i,j,k)\in D_{FN}} (\ln\phi(\mathbf{u}_i^T\mathbf{v}_j - \mathbf{u}_i^T\mathbf{v}_k)) + \textstyle\sum_{i=1}^{n}\sum_{j=1}^{m} I(\mathbf{F}_{ij} \neq 1)\mathbf{v}_j^T\mathbf{y'}^i{}_j\end{aligned} \tag{14}$$

where $\mathbf{x'}_j = \sum_i \mathbf{x'}^i{}_j \in \mathbb{R}^{K\times 1}$ helps protect u_i's submission containing sensitive feedbacks on v_j with a relatively small privacy budget ϵ_s when calculating $\mathbf{F}_{ij}\sigma_j^{iP}$, and $\mathbf{y'}_j = \sum_i \mathbf{y'}^i{}_j \in \mathbb{R}^{K\times 1}$ provides protection on submission without sensitive feedbacks on v_j with a relatively large privacy budget ϵ_n when calculating $\mathbf{G}_{ij}\sigma_j^{iP}$, σ_j^{iF} and σ_j^{iN}. We then select noises based on the similar procedures of (6) and (7). According to Theorem 4, we can protect sensitive feedbacks with random noises from Lap($\frac{\sqrt{K}}{\epsilon_s}$) and Lap($\frac{\sqrt{K}}{\epsilon_n}$).

Theorem 4 *If each element in* $\mathbf{x'}_j$ *and* $\mathbf{y'}_j$ *is independently and randomly selected from Lap*($\frac{\sqrt{K}}{\epsilon_s}$) *and Lap*($\frac{\sqrt{K}}{\epsilon_n}$)*, the extended PrivSR can satisfy* ϵ*-differential privacy.*

Proof See Section 6.4 for the detailed proof. □

Algorithm 2 Extended *PrivSR* Algorithm

Input: $\mathcal{J}_2$, ϵ, γ, β, λ, observed feedbacks $\mathcal{R}$, user u_i holds its $\mathbf{u}_i$ and $\mathcal{F}_i$
Output: $\mathbf{U}$, $\mathbf{V}$

```
Initialize U and V
while not converge do
    for j = 1, ..., m do
        // Calculate v_j on recommender's side
        for i where (i, j) ∈ P_i do
            Obtain F_ij(σ_j^iP + x'^i_j) + G_ij(σ_j^iP + y'^i_j) from user u_i
        end for
        for i where (i, j) ∈ F_i do
            Obtain σ_j^iF + y'^i_j from user u_i
        end for
        for i where (i, j) ∈ N_i do
            Obtain σ_j^iN + y'^i_j from user u_i
        end for
        Update v_j as v_j = v_j + γ ∂J_2/∂v_j
    end for
    for i = 1, ..., n do
        // Calculate u_i on user u_i 's side
        Update u_i as u_i = u_i + γ ∂J_2/∂u_i
    end for
end while
```

We can perform gradient descent over $\mathbf{U}$ and $\mathbf{V}$ iteratively to learn the maximum $\mathcal{J}_2$. In each iteration of learning process: 1) on the recommender's side, it is $O(K(|D_{PF}| + |D_{FN}|))$ to optimize item latent matrix for items in sets D_{PF} and D_{FN}; 2) on each user's side, it is $O(K|D^i_{PF}| + |D^i_{FN}|)$ to learn the user latent vector. Calculating a full gradient will be too slow or even infeasible, therefore, we can utilize a stochastic learning scheme and randomly sample a subset of D_{PF} and D_{FN} [35], then in the whole learning process, the time complexity is around $O(NKa)$ for the recommender and $O(NKb)$ for each user, where $a \ll |D_{PF}| \ll |D_{FN}|$ and $b \approx |\mathcal{R}_{i*}| \ll |D^i_{PF}| \ll |D^i_{FN}|$. We describe our extended *PrivSR* in Algorithm 2, where the recommender updates $\mathbf{v}_j$ in line 4 through 14 referring to (13) with corresponding personalized noises, and users update $\mathbf{u}_i$ in line 17 through 18 with (12) since $\frac{\partial \mathcal{J}_2}{\partial \mathbf{u}_i} = \frac{\partial \mathcal{J}_1}{\partial \mathbf{u}_i}$.

6 Analysis of privacy guarantee

The detailed proofs for Theorems in Section 5 are shown in the following.

6.1 Proof of Theorem 1

Consider the characteristic of normal distribution that the sum of variable from normal distribution is also distributed as normal distribution [7], it is obvious that both of $\sum_{i\in\mathcal{R}_{s,j}} \mathbf{c}_{ij_s}$ and $\sum_{i\in\mathcal{R}_{n,j}} \mathbf{c}_{ij_n}$ are distributed as $N(0, 1)$. Then based on Lemma 1, we know $\sum_{i\in\mathcal{R}_{s,j}} \mathbf{x}^i_j$ and $\sum_{i\in\mathcal{R}_{n,j}} \mathbf{y}^i_j$ are distributed as $Lap(\frac{2\Delta\sqrt{K}}{\epsilon_s})$ and $Lap(\frac{2\Delta\sqrt{K}}{\epsilon_n})$.

Let $\epsilon_s = (1+\beta)\epsilon$, $\epsilon_n = (\frac{1}{\beta}+1)\epsilon$, and c is distributed as $N(0, 1)$. Since every user keeps the same $\mathbf{h}_j$ when updates $\mathbf{v}_j$ in each iteration, the summation of these random noise vector for sensitive and non-sensitive feedbacks can be calculated as

$$
\begin{aligned}
\mathbf{p}_j &= \sum_{i\in\mathcal{R}_{s,j}} \mathbf{x}^i_j + \sum_{i\in\mathcal{R}_{n,j}} \mathbf{y}^i_j \\
&= \frac{2\Delta\sqrt{2K\mathbf{h}_j}}{\epsilon_s} \sum_{i\in\mathcal{R}_{s,j}} \mathbf{c}_{ij_s} + \frac{2\Delta\sqrt{2K\mathbf{h}_j}}{\epsilon_n} \sum_{i\in\mathcal{R}_{n,j}} \mathbf{c}_{ij_n} \\
&= 2\Delta c\sqrt{2K\mathbf{h}_j}(\frac{1}{\epsilon_s} + \frac{1}{\epsilon_n}) \\
&= 2\Delta c\sqrt{2K\mathbf{h}_j}\left(\frac{1}{(1+\beta)\epsilon} + \frac{1}{(\frac{1}{\beta}+1)\epsilon}\right) \\
&= \frac{2\Delta\sqrt{K}}{\epsilon}\sqrt{2\mathbf{h}_j}c
\end{aligned}
$$

Then each element in $\mathbf{p}_j = \{p_{j1}, p_{j2}, ..., p_{jl}, ..., p_{jK}\}$ is distributed as $Lap(\frac{2\Delta\sqrt{K}}{\epsilon})$ based on Lemma 1, which is equal to that we randomly picked each p_{jl} from the $Lap(\frac{2\Delta\sqrt{K}}{\epsilon})$ distribution, whose probability density function is $Pr(p_{jl}) = \frac{\epsilon}{4\Delta\sqrt{K}} e^{-\frac{\epsilon|p_{jl}|}{2\Delta\sqrt{K}}}$.

Let D_1 and D_2 be two datasets only differ from one record $\mathbf{R}_{ab}$ and $\tilde{\mathbf{R}}_{ab}$, which can be sensitive or non-sensitive. From the different inputs D_1 and D_2, we obtain the same output, i.e., the same derived $\mathbf{V}$. Since the derived $\mathbf{V}$ are the optimized result after convergence, we

then have $\frac{\partial \mathcal{J}(D_1)}{\partial \mathbf{v}_j} = \frac{\partial \mathcal{J}(D_2)}{\partial \mathbf{v}_j} = 0$ as (9), which then can be formulated as,

$$2\sum_{i=1}^{n} \mathbf{I}_{ij}(\mathbf{u}_i^T \mathbf{v}_j - \mathbf{R}_{ij})\mathbf{u}_i + \mathbf{p}_j = 2\sum_{i=1}^{n} \mathbf{I}_{ij}(\mathbf{u}_i^T \mathbf{v}_j - \tilde{\mathbf{R}}_{ij})\mathbf{u}_i + \tilde{\mathbf{p}}_j \tag{15}$$

As feedbacks in D_1 and D_2 only differs from $\mathbf{R}_{ab}$ and $\tilde{\mathbf{R}}_{ab}$, then we can get

$$\mathbf{p}_j - \tilde{\mathbf{p}}_j = 2\mathbf{u}_i(\mathbf{R}_{ab} - \tilde{\mathbf{R}}_{ab}).$$

Considering $|\mathbf{R}_{ab} - \tilde{\mathbf{R}}_{ab}| \leq \Delta$ and $||\mathbf{u}_i|| \leq 1$, it's obvious $||\mathbf{p}_j - \tilde{\mathbf{p}}_j|| \leq 2\Delta$.

We then formulate the probability that we get the same derived $\mathbf{V}$ with the different datasets D_1 and D_2 after convergence. For each vector $\mathbf{v}_j$ of $\mathbf{V}$, we have

$$\begin{aligned} \frac{Pr[\mathbf{v}_j|D_1]}{Pr[\mathbf{v}_j|D_2]} &= \frac{\prod_{l\in\{1,2,\ldots,K\}} Pr(p_{jl})}{\prod_{l\in\{1,2,\ldots,K\}} Pr(\tilde{p}_{jl})} \\ &= e^{-\frac{\epsilon \sum_l |p_{jl}|}{2\Delta\sqrt{K}}} / e^{-\frac{\epsilon \sum_l |\tilde{p}_{jl}|}{2\Delta\sqrt{K}}} = e^{\frac{\epsilon \sum_l (|p_{jl}| - |\tilde{p}_{jl}|)}{2\Delta\sqrt{K}}} \\ &\leq e^{\frac{\epsilon\sqrt{K\sum_l (p_{jl}-\tilde{p}_{jl})^2}}{2\Delta\sqrt{K}}} = e^{\frac{\epsilon\sqrt{K}||\mathbf{p}_j - \tilde{\mathbf{p}}_j||}{2\Delta\sqrt{K}}} \leq e^{\epsilon} \end{aligned} \tag{16}$$

So, we obtain the conclusion.

6.2 Proof of Theorem 2

With the characteristic of normal distribution and Lemma 1, we know $2\sum_{f\in\mathcal{F}_i} \mathbf{q}_i^f \sim Lap(\frac{2\sqrt{K}}{\epsilon})$.

Let D_1 and D_2 be two datasets only differ from one record $\mathbf{u}_i^f$ and $\tilde{\mathbf{u}}_i^f$. From the different inputs D_1 and D_2, we obtain the same output, i.e., the same derived $\mathbf{U}$. Since the derived $\mathbf{U}$ is the optimized results after convergence, then we know $\frac{\partial \mathcal{J}(D_1)}{\partial \mathbf{u}_i} = \frac{\partial \mathcal{J}(D_2)}{\partial \mathbf{u}_i} = 0$ as (10), which can be formulated as,

$$\mathbf{q}_i^f + 2\sum_{f\in\mathcal{F}_i} S_{if}(\mathbf{u}_i - \mathbf{u}_f) = \tilde{\mathbf{q}}_i^f + 2\sum_{f\in\mathcal{F}_i} S_{if}(\mathbf{u}_i - \tilde{\mathbf{u}}_f) \tag{17}$$

As there's only one difference for D_1 and D_2, then we can get $\mathbf{q}_i^f - \tilde{\mathbf{q}}_i^f = 2\sum_{f\in\mathcal{F}_i} S_{if}(\mathbf{u}_f - \tilde{\mathbf{u}}_f)$. Considering $|S_{if} - S'_{if}| \leq 1$ and $||\mathbf{u}_f|| \leq 1$, it's obvious $||\mathbf{q}_i^f - \tilde{\mathbf{q}}_i^f|| \leq 2$.

We then formulate the probability that we get the same derived $\mathbf{U}$ with the different datasets D_1 and D_2. Similar to (16), for each $\mathbf{u}_i$ of $\mathbf{U}$, we have $\frac{P[\mathbf{u}_i|D_1]}{P[\mathbf{u}_i|D_2]} \leq e^{\epsilon}$. So, we obtain the conclusion.

6.3 Proof of Theorem 3

We combine the rating model and social relation model together in (8). Since we don't jointly optimize (8) w.r.t. $\mathbf{V}$ and $\mathbf{U}$, we then optimize (8) w.r.t. $\mathbf{V}$ and $\mathbf{U}$ separately with (9) and (10).

For $\mathbf{V}$, the only difference of derivative of (6) and (9) w.r.t. $\mathbf{v}_j$ is the regularization $2\lambda\mathbf{v}_j$. Then we should add $2\lambda\mathbf{v}_j$ on both sides of (15). Since both datasets get the same $\mathbf{v}_j$, then the results won't change. The derived V still satisfies ϵ-differential privacy.

For $\mathbf{U}$, because of the difference of derivative and (7) and (10) w.r.t. $\mathbf{u}_i$, we need to add $2\sum_{j=1}^{m} \mathbf{I}_{ij}(\mathbf{u}_i^T \mathbf{v}_j - \mathbf{R}_{ij})\mathbf{v}_j + 2\lambda\mathbf{u}_i$ on both sides of (17). Since D_1 and D_2 are only different

at $\mathbf{u}_f$ and $\tilde{\mathbf{u}}_f$, then $||\mathbf{q}_i^f - \tilde{\mathbf{q}}_i^f||$ won't change, thus the derived U still satisfies ϵ-differential privacy.

In conclusion, Algorithm 1 satisfies ϵ-differential privacy, which means attackers can't learn users' sensitive feedbacks or other user' latent profile in the whole process.

6.4 Proof of Theorem 4

Similar to Section 6.1, each element in $\mathbf{p}_j = \sum_i \mathbf{x'}^i_j + \sum_i \mathbf{y'}^i_j$ is distributed as $Lap(\frac{\sqrt{K}}{\epsilon})$. Then let D_1 and D_2 be two datasets only differ from one feedback $\mathbf{R}_{ij}$ and $\tilde{\mathbf{R}}_{ij}$, which can be sensitive or non-sensitive. From the different inputs D_1 and D_2, we obtain the same output, i.e., the same derived $\mathbf{V}$. Since the derived $\mathbf{V}$ are the optimized result after convergence, we then have $\frac{\partial \mathcal{J}_2(D_1)}{\partial \mathbf{v}_j} = \frac{\partial \mathcal{J}_2(D_2)}{\partial \mathbf{v}_j} = 0$. Suppose the different feedbacks are from P_i, which then can be formulated as,

$$\begin{aligned} &\sum_{i:(i,j)\in P_i} \sigma_j^{iP} + \sum_{i:(i,j)\in F_i} \sigma_j^{iF} + \sum_{i:(i,j)\in N_i} \sigma_j^{iN} + \mathbf{p}_j = \\ &\sum_{i:(i,j)\in P_i} \tilde{\sigma}_j^{iP} + \sum_{i:(i,j)\in F_i} \sigma_j^{iF} + \sum_{i:(i,j)\in N_i} \sigma_j^{iN} + \tilde{\mathbf{p}}_j \end{aligned} \tag{18}$$

As feedbacks in D_1 and D_2 only differs from σ_j^{iP} and $\tilde{\sigma}_j^{iP}$, then we can get

$$\mathbf{p}_j - \tilde{\mathbf{p}}_j = \tilde{\sigma}_j^{iP} - \sigma_j^{iP}.$$

It is the similar case if the different feedbacks are from F_i or N_i. As $\sum_{i:(i,j)\in P_i} \sigma_j^{iP} = \sum_{(i,j,k)\in D_{PF}} \psi_{ijk}\mathbf{u}_i$, $\sum_{i:(i,j)\in F_i} \sigma_j^{iF} = -\sum_{(i,k,j)\in D_{PF}} \psi_{ikj}\mathbf{u}_i + \sum_{(i,j,k)\in D_{FN}} \psi_{ijk}\mathbf{u}_i$ and $\sum_{i:(i,j)\in N_i} \sigma_j^{iN} = -\sum_{(i,k,j)\in D_{FN}} \psi_{ikj}\mathbf{u}_i$, considering $|\tilde{\psi}_{ijk} - \psi_{ijk}| < 1$ and $||\mathbf{u}_i|| \leq 1$, it's obvious $||\mathbf{p}_j - \tilde{\mathbf{p}}_j|| \leq 1$.

We then formulate the probability that we get the same derived $\mathbf{V}$ with the different datasets D_1 and D_2 after convergence. Similar to (16), for each vector $\mathbf{v}_j$ of $\mathbf{V}$, we have $\frac{Pr[\mathbf{v}_j|D_1]}{Pr[\mathbf{v}_j|D_2]} \leq e^{\epsilon}$. Then similar to Section 6.3, we obtain the conclusion.

7 Experimental evaluation

In this section, we conduct experimental evaluation to validate the effectiveness of *PrivSR*. We aim to answer two questions:

- Can *PrivSR* improve recommendation effectiveness by incorporating social relationships?
- Can it protect sensitive feedbacks under reconstruction attack while retaining recommendation effectiveness?

In the following, we first introduce our datasets and experimental settings, then we conduct experimental evaluation followed by parameter sensitivity analysis.

Table 1 Statistics of the datasets

Dataset	Ciao	Epinions
# of users	7,193	17,950
# of items	21,889	49,760
# of ratings	183,415	508,936
# of relationships	28,513	14,017

7.1 Datasets and experimental settings

Two publicly available datasets Ciao[3] and Epinions[4] are used for evaluation. For both datasets, users feedbacks are ratings although we can have the similar results with other kinds of feedbacks. Users' ratings are from 1 to 5 and users can establish social relations with others. Detailed statistics of these two datasets are shown in Table 1. These two datasets possess social relations of different sparsity which can help validate effectiveness and generality of *PrivSR*. For each dataset, to simulate the setting of personalized privacy preferences, we randomly select x percent of the ratings as sensitive ratings and the remaining $100 - x$ as non-sensitive ratings. We vary x as $\{0, 10, \ldots, 50\}$ and use five-fold cross validation for the following experiments.

To evaluate the performance of recommendation and reconstruction attack, we use the two popular metrics Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The MAE is defined as

$$MAE = \frac{\sum_{(i,j)\in\mathcal{R}} |\widehat{\mathbf{R}}_{ij} - \mathbf{R}_{ij}|}{|\mathcal{R}|} \tag{19}$$

where $\widehat{\mathbf{R}}_{ij}$ is the predicted rating from u_i to v_j. The RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in\mathcal{R}} (\widehat{\mathbf{R}}_{ij} - \mathbf{R}_{ij})^2}{|\mathcal{R}|}} \tag{20}$$

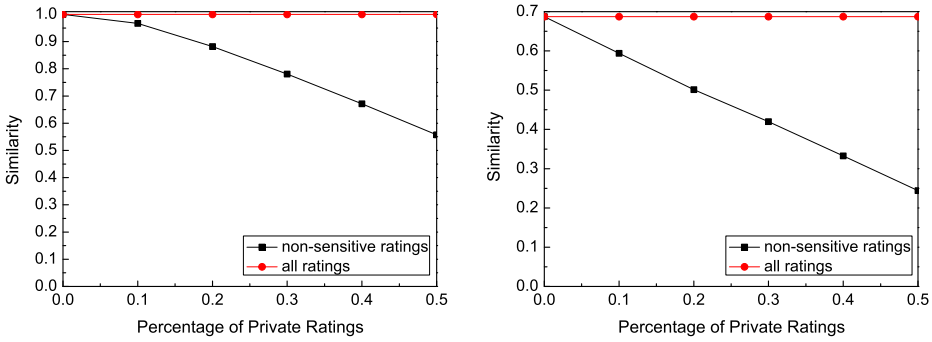
For recommendation, smaller MAE denotes better performance. For reconstruction attack on sensitive rating, larger MAE means better protection performance. Note that previous work demonstrated that small improvement in MAE/RMSE terms can have a significant impact on the quality of the top-few recommendation [16]. We compare among three representative state-of-the-art recommender systems:

- **MF**: matrix factorization tries to decompose the user-item rating matrix into two matrices for recommendation [18].
- **SoReg**: this method incorporates social regularization on matrix factorization to represent the social constrains on recommender systems [23].
- **DPMF**: differential private matrix factorization [12] treats all ratings private and uses equally perturbed noise for latent matrix learning.

For each mechanism, the parameters are tuned via cross-validation on training data, we then set $\gamma = 10^{-4}$, $\lambda = 10^{-3}$, $\alpha = 10^{-2}$ and the dimension $d = 10$. For convenience, we fix $\beta = 0.1$ for *PrivSR* in the first two experiments, and accordingly $\epsilon_s = 1.1\epsilon$ and $\epsilon_n = 11\epsilon$.

[3] http://www.ciao.co.uk/

[4] http://www.epinions.com/



(a) Similarity on Ciao (b) Similarity on Epinions

Figure 4 Friends similarity with non-sensitive data and the whole dataset

More details about parameter selection for the proposed framework will be discussed in the following subsections.

7.2 Feasibility analysis

Since nobody except users themselves can get the sensitive ratings, we calculate the similarity between friends with only non-sensitive datasets. Figure 4a and b show that within an appropriate range, the difference between similarity calculated by the whole datasets and by only non-sensitive data is acceptable, which also meets the statistic that users set around 10% of their posts private in the real-world social networks.[5]

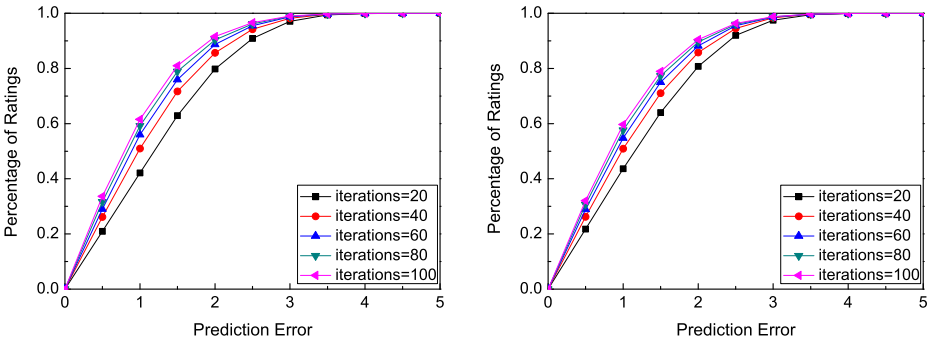
To better understand the proposed model, we also plot the CDF of each iteration in Figure 5a and b. From intuition, since we add noise in each iteration, it seems that the more iterations, the worse recommendation performance. However, Figure 5a and b indicate the iterations help the model converge and offer smaller prediction errors.

7.3 Recommendation effectiveness comparison

To answer the first question, we evaluate the recommendation effectiveness on the test datasets. To provide a fair comparison on these approaches, we provide the same input ratings to these baselines no matter whether or not they can protect users' privacy. The average MAE and RMSE results are shown in Figures 6 and 7, from which we observe:

- When $x = 0$, *PrivSR* with $\epsilon = 0.1$ can perform almost as good as SoReg, which confirms that noise perturbation on non-sensitive ratings will not significantly affect recommendation effectiveness.
- SoReg consistently provides the best recommendation effectiveness, showing that social relationships can help increase recommendation performance.
- In general, *PrivSR* with $\epsilon = 0.1$ slightly inferior than SoReg but outperforms other methods though we attach great noises, which shows the tradeoff between recommendation effectiveness and privacy protection, and also proves the effectiveness of the well-balanced privacy budgets for sensitive and non-sensitive ratings.

[5] https://techcrunch.com/2009/10/05/twitter-data-analysis-an-investors-perspective-2



(a) CDF on Ciao (b) CDF on Epinions

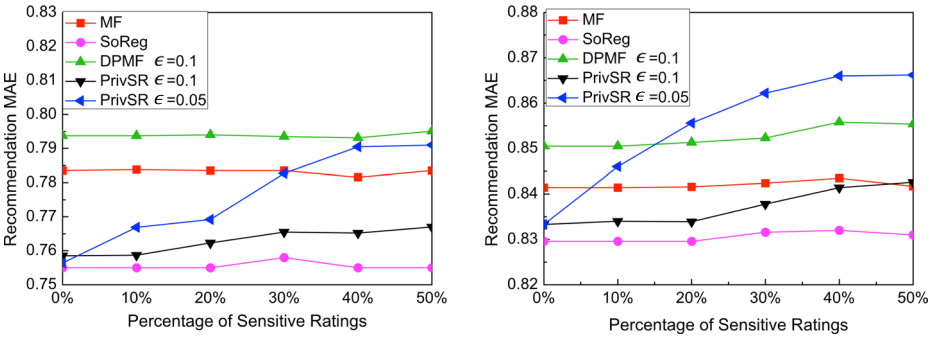
Figure 5 CDF of prediction error with varying iterations

– Although the privacy budget of *PrivSR* with $\epsilon = 0.05$ is much smaller than DPMF with $\epsilon = 0.1$, the corresponding recommendation effectiveness of *PrivSR* is better than DPMF when the percentage of private ratings is not too large. In real world, the percentage of sensitive ratings is usually around 10% [39], thus *PrivSR* with $\epsilon = 0.05$ can still achieve very good recommendation effectiveness in practical.

Based on the aforementioned observations, we conclude that *PrivSR* can improve recommendation effectiveness by utilizing rich social relationships and designing well-balanced privacy budgets for sensitive and non-sensitive ratings.

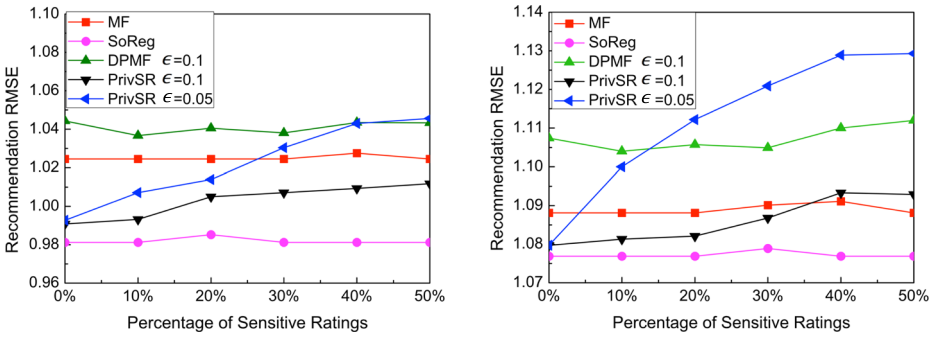
7.4 Privacy protection comparison

To answer the second question, we simulate the reconstruction attack. There are multiple options for conducting reconstruction attack [9]. We conduct it using the matrix factorization-based model. Since attackers can obtain both $\mathbf{V}$ and $\mathcal{R}_n$, they can infer a rough



(a) MAE on Ciao (b) MAE on Epinions

Figure 6 Recommendation performance comparison in terms of MAE



(a) RMSE on Ciao (b) RMSE on Epinions

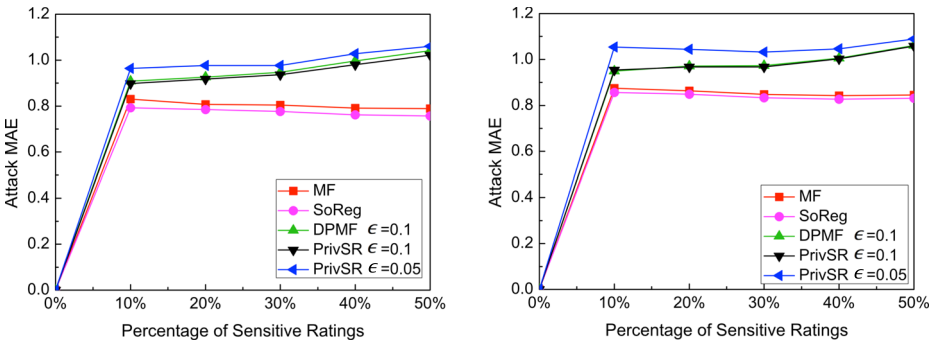
Figure 7 Recommendation performance comparison in terms of RMSE

user latent profile $\tilde{\mathbf{u}}_i$ of the victim u_i by solving the following equation:

$$\min_{\tilde{\mathbf{U}}} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{G}_{ij} (\mathbf{R}_{ij} - \tilde{\mathbf{u}}_i^T \mathbf{v}_j)^2 \tag{21}$$

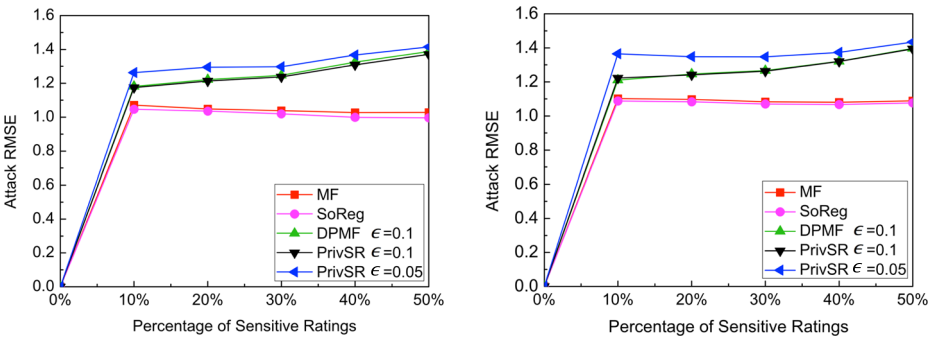
By using gradient descend, all the sensitive ratings can be obtained by $\tilde{\mathbf{U}}$ and $\mathbf{V}$. We want to protect sensitive ratings, such that prediction of sensitive ratings is inaccurate, and a larger MAE/RMSE value on sensitive ratings represents a better privacy protection. From Figures 8 and 9, we can obtain the following observations:

– Noise perturbation helps increase the level of privacy protection against reconstruction attacks.
– With the similar privacy budget, the level of privacy protection provided by *PrivSR* and DPMF are similar. However, *PrivSR* can achieve better recommendation effectiveness with different privacy budgets for sensitive and non-sensitive ratings. We perform t-test on recommendation effectiveness of *PrivSR* and DPMF with the same privacy budgets for sensitive ratings. The test results show that the improvement is statistically



(a) MAE on Ciao (b) MAE on Epinions

Figure 8 Privacy protection comparison in terms of MAE



(a) RMSE on Ciao (b) RMSE on Epinions

Figure 9 Privacy protection comparison in terms of RMSE

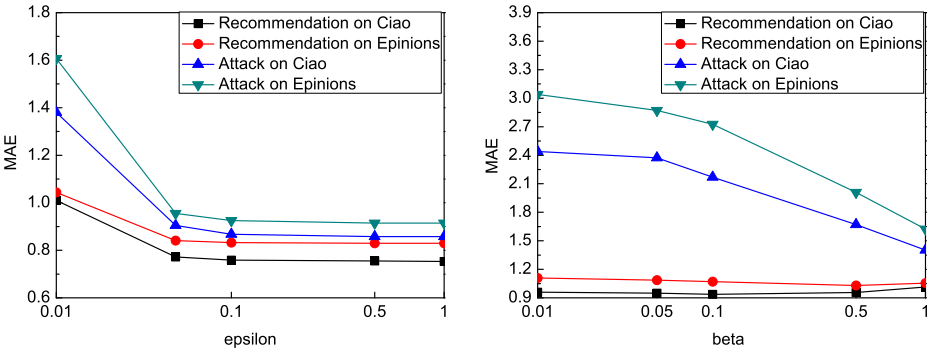
significant. These results indicate *PrivSR* can achieve a better balance between privacy protection and recommendation effectiveness in practical.

– *PrivSR* with a lower privacy budget can significantly increase the level of privacy protection while being able to retain a good recommendation effectiveness, especially when the percentage of private ratings x is smaller than 20% which can still meet the users' needs of privacy protection in real world [39].

Based on the aforementioned observations, we conclude that *PrivSR* outperforms the state-of-the-art recommender systems on privacy protection while retaining great recommendation effectiveness.

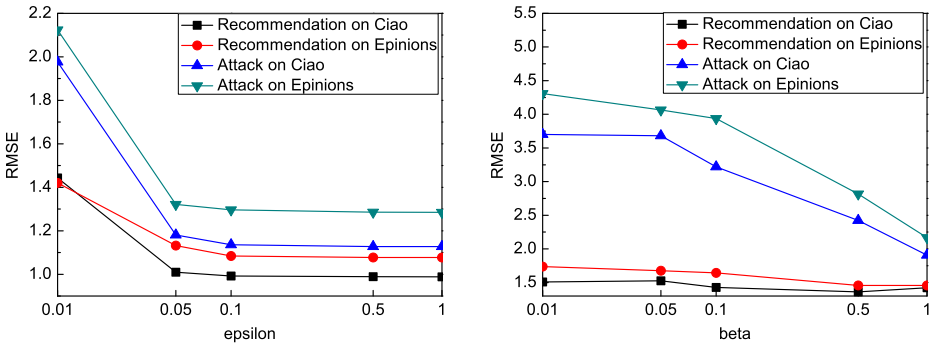
7.5 Impact of parameters ϵ and β

For simplification, we set $x = 10$ in the following experiments, based on the real-world statistical results [39]. We randomly select 10% ratings of the entire datasets as the sensitive rating set. To understand the impact of ϵ and β, we change ϵ from {0.01, 0.05, 0.1, 0.5, 1} with fixed $\beta = 1$. Also, we vary β from {0.01, 0.05, 0.1, 0.5, 1} with fixed $\epsilon = 0.01$. The results are shown in Figures 10 and 11, from which we observe that: 1) Larger privacy



(a) Performance with ϵ (b) Performance with β

Figure 10 Performance with varying parameters in terms of MAE



(a) Performance of ϵ (b) Performance of β

Figure 11 Performance with varying parameters in terms of RMSE

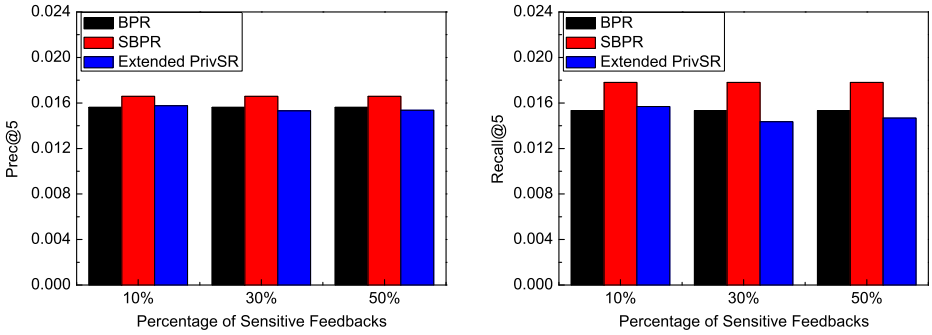
budget indicates less noise, resulting in better recommendation effectiveness and worse privacy protection. This is a common observation about the trade-off between privacy and utility [18, 26, 41]. 2) With fixed ϵ, the recommendation effectiveness stays the same, while larger β indicates larger privacy budget for sensitive data and smaller for the non-sensitive, which makes the privacy protection decrease on the sensitive ratings.

7.6 Evaluating the extended *PrivSR*

For ranking formed matrix factorization to model implicit feedbacks, similar to (21), attackers can obtain $\mathbf{V}$ and $\mathcal{R}_n$, and then they can also infer a rough user latent profile $\tilde{\mathbf{u}}_i$ of the victim u_i though by solving the following equation:

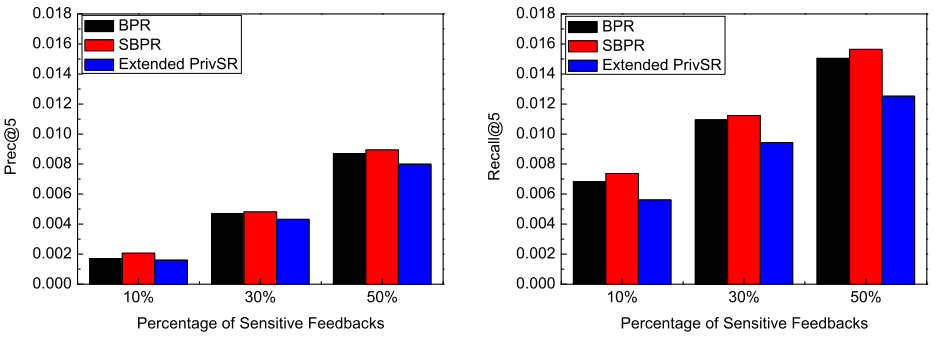
$$\max_{\tilde{\mathbf{U}}} \sum_{(i,j,k)\in D} \ln \phi(\tilde{\mathbf{u}}_i^T \mathbf{v}_j - \tilde{\mathbf{u}}_i^T \mathbf{v}_k) \tag{22}$$

where $D = \{(i, j, k)|\mathbf{G}_{ij} = 1 \wedge \mathbf{G}_{ik} \neq 1\}$. Following the common way [4, 31, 42, 44, 48], we remove all negative rating feedbacks (less than 4 stars) of Ciao dataset and treat remaining observed ratings as positive implicit feedbacks of "like" or "purchase". We conduct experiments, setting $\beta = 1$ and $\epsilon = 0.1$, with baseline schemes *BPR* [31] and



(a) $precision@5$ (b) $recall@5$

Figure 12 Recommendation performance comparison of the extended *PrivSR*



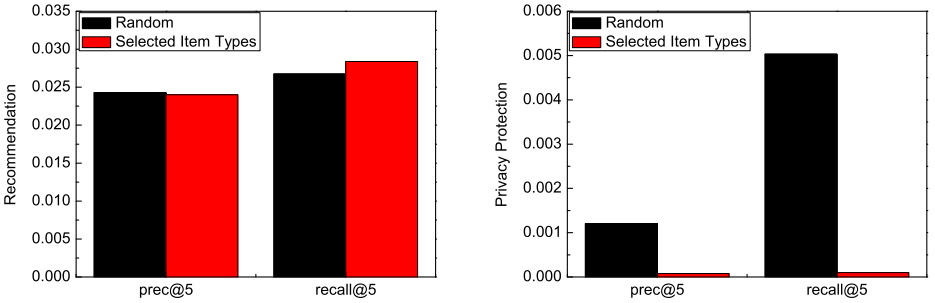
(a) $precision@5$ (b) $recall@5$

Figure 13 Privacy protection comparison of the extended *PrivSR*

SBPR [48]. To evaluate precision and recall of top k results from implicit feedbacks, we use two broadly used metrics $precision@k$ and $recall@k$, where we set k to be 5. Note that a higher $precision@k$ and $recall@k$ indicate a better recommendation effectiveness and worse privacy protection.

The experimental results are shown in Figures 12 and 13: 1) Based on the the recommendation effectiveness and privacy protection of SBPR and BPR, we can observe that social relationships help increase top k recommendation effectiveness but involve in more potential privacy leakage. 2) Compared to BPR, the extended *PrivSR* can effectively increase the level of privacy protection while can even achieve better recommendation effectiveness when percentage of sensitive feedbacks is less than 30%, which shows the extended *PrivSR* can achieve a good balance between privacy protection and recommendation effectiveness. 3) Compared to SBPR, the extended PrivSR can outperform on privacy protection but not on recommendation performance, which proves that there is tradeoff between recommendation effectiveness and privacy protection.

In real world, users are more likely to regards some types of items as sensitive, such as, items in "health", "adult products" and "family" categories. In dataset Ciao, we select sensitive feedbacks in two ways: 1) *selection based on selected item types*, i.e., set items in these three categories as sensitive; and 2) *randomly selection*, i.e., randomly select feedbacks and set them sensitive. We conduct experiments with the same number of sensitive feedbacks,



(a) Recommendation effectiveness (b) Privacy protection

Figure 14 Comparison of different sensitive feedbacks selection

and the results are shown in Figure 14. We can observe that: compared to randomly selection, selection based on selected item types can achieve similar recommendation effective but obtain much better privacy protection, which proves that our *PrivSR* can even perform better in practical.

8 Conclusion and future work

In this paper, we study the problem of privacy-preserving social recommendation with personalized privacy settings. We propose a novel differential privacy-preserving framework in a semi-centralized way which can protect users' sensitive ratings while being able to retain the effectiveness of recommendation. Theoretic analysis and experimental evaluation on real-world datasets demonstrate the effectiveness of the proposed framework for recommendation and privacy protection.

There are several directions can be further investigated. First, in this paper, we build our model based on traditional machine learning methods. We would like to study privacy preserving social recommendation with deep learning techniques. Second, in this paper, we only consider static recommender systems. We would like to investigate privacy preserving issue for temporal and dynamic data [17] as well.

Acknowledgments This work is supported by, or in part by, National Science Foundation of China (61672500, 61572474), and Program of International S&T Cooperation (2016YFE0121500). Suhang Wang and Huan Liu are supported by the National Science Foundation (NSF) under the grant #1614576 and Office of Naval Research (ONR) under the grant N00014-16-1-2257.

References

1. Bhamidipati, S., Fawaz, N., Kveton, B., Zhang, A.: Priview: Personalized media consumption meets privacy against inference attacks. IEEE Softw. **32**(4), 53–59 (2015)
2. Cao, Y., Yoshikawa, M., Xiao, Y., Xiong, L.: Quantifying differential privacy under temporal correlations. In: 33rd IEEE International Conference on Data Engineering, ICDE 2017. San Diego, CA, USA, April 19-22, 2017, pp. 821–832. https://doi.org/10.1109/ICDE.2017.132 (2017)
3. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. J. Mach. Learn. Res. **12**, 1069–1109 (2011)
4. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010. Barcelona, Spain, September 26-30, 2010, pp. 39–46. https://doi.org/10.1145/1864708.1864721 (2010)
5. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference, pp. 265–284. Springer (2006)
6. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Found. Trends. Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
7. Eisenberg, B., Sullivan, R.: Why is the sum of independent normal random variables normal? Math. Mag. **81**(5), 362–366 (2008)
8. Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., Ristenpart, T.: Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In: Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014., pp. 17–32 (2014)
9. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: ACM Sigsac Conference on Computer and Communications Security, pp. 1322–1333 (2015)

10. Gross, R., Acquisti, A.: Information revelation and privacy in online social networks. In: Proceedings of the Acm Workshop on Privacy in the Electronic Society, pp. 71–80 (2005)
11. Hoens, T.R., Blanton, M., Chawla, N.V.: A private and reliable recommendation system for social networks. In: IEEE Second International Conference on Social Computing, pp. 816–825 (2010)
12. Hua, J., Xia, C., Zhong, S.: Differentially private matrix factorization. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015. Buenos Aires, Argentina, July 25-31, 2015, pp. 1763–1770. http://ijcai.org/Abstract/15/251 (2015)
13. Jorgensen, Z., Yu, T.: A privacy-preserving framework for personalized, social recommendations. In: International Conference on Extending Database Technology, EDBT (2014)
14. Kasiviswanathan, S.P., Rudelson, M., Ullman, J.: The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In: ACM Symposium on Theory of Computing, pp. 775–784 (2010)
15. Komarova, T., Nekipelov, D., Yakovlev, E.: Estimation of treatment effects from combined data: Identification versus data security. In: Iccas-Sice, pp. 3066–3071 (2013)
16. Koren, Y.: Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434 (2008)
17. Koren, Y.: Collaborative filtering with temporal dynamics. Commun. ACM **53**(4), 89–97 (2010)
18. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Comput. **42**(8), 30–37 (2009). https://doi.org/10.1109/MC.2009.263
19. Kotz, S., Kozubowski, T., Podgorski, K.: The Laplace Distribution and Generalizations: A Revisit with Applications to Communications, Economics, Engineering, and Finance. Springer Science & Business Media (2012)
20. Krohngrimberghe, A., Drumond, L., Freudenthaler, C., Schmidtthieme, L.: Multi-relational matrix factorization using bayesian personalized ranking for social network data, 173–182 (2012)
21. Li, Q., Li, J., Wang, H., Ginjala, A.: Semantics-enhanced privacy recommendation for social networking sites. In: IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 226–233 (2011)
22. Liu, K., Terzi, E.: A framework for computing the privacy scores of users in online social networks. Acm Trans. Knowl. Discov. Data **5**(1), 1–30 (2010)
23. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011. Hong Kong, China, February 9-12, 2011, pp. 287–296. https://doi.org/10.1145/1935826.1935877 (2011)
24. Machanavajjhala, A., Korolova, A., Sarma, A.D.: Personalized social recommendations: Accurate or private. Proc. VLDB Endow. **4**(7), 440–450 (2011)
25. McSherry, F.: Privacy integrated queries: An extensible platform for privacy-preserving data analysis. Commun. ACM **53**(9), 89–97 (2010). https://doi.org/10.1145/1810891.1810916
26. Meng, X., Xu, Z., Chen, B., Zhang, Y.: Privacy-preserving query log sharing based on prior n-word aggregation. In: Trustcom, pp. 722–729 (2016)
27. Meng, X., Wang, S., Liu, H., Zhang, Y.: Exploiting emotion on reviews for recommender systems. In: AAAI (2018)
28. Minkus, T., Liu, K., Ross, K.W.: Children seen but not heard: When parents compromise children's online privacy. In: International Conference on World Wide Web, pp. 776–786 (2015)
29. Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., Boneh, D.: Privacy-preserving matrix factorization. In: 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13. Berlin, Germany, November 4-8, 2013, pp. 801–812. https://doi.org/10.1145/2508859.2516751 (2013)
30. Rajkumar, A., Agarwal, S.: A differentially private stochastic gradient descent algorithm for multiparty classification. Jmlr (2012)
31. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: UAI 2009, pp. 452–461 (2009)
32. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems. Vancouver, British Columbia, Canada, December 3-6, 2007, pp. 1257–1264. http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization (2007)
33. Shokri, R., Stronati, M., Shmatikov, V.: Membership inference attacks against machine learning models (2016)
34. Shu, K., Wang, S., Tang, J., Wang, Y., Liu, H.: Crossfire: Cross media joint friend and item recommendations. In: WSDM (2018)
35. Song, D., Meyer, D.A., Tao, D.: Top-k link recommendation in social networks. In: 2015 IEEE International Conference on Data Mining, ICDM 2015. Atlantic City, NJ, USA, November 14-17, 2015, pp. 389–398. https://doi.org/10.1109/ICDM.2015.136 (2015)

36. Song, S., Wang, Y., Chaudhuri, K.: Pufferfish privacy mechanisms for correlated data. In: Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017. Chicago, IL, USA, May 14-19, 2017, pp. 1291–1306. https://doi.org/10.1145/3035918.3064025 (2017)
37. Tang, J., Hu, X., Liu, H.: Social recommendation: A review. Social Netw. Analys. Mining **3**(4), 1113–1133 (2013). https://doi.org/10.1007/s13278-013-0141-9
38. Tang, Q., Wang, J.: Privacy-preserving friendship-based recommender systems. IEEE Trans. Depend. Secur. Comput. **PP**(99), 1–1 (2016)
39. Twitter data analysis: An investor's perspective. https://techcrunch.com/2009/10/05/twitter-data-analysis-an-investors-perspective-2
40. Wang, S., Tang, J., Liu, H.: Toward dual roles of users in recommender systems. In: CIKM (2015)
41. Wang, Y., Si, C., Wu, X.: Regression model fitting under differential privacy and model inversion attack. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, pp. 1003–1009. http://ijcai.org/Abstract/15/146 (2015)
42. Wang, X., Lu, W., Ester, M., Wang, C., Chen, C.: Social recommendation with strong and weak ties. In: ACM International on Conference on Information and Knowledge Management, pp. 5–14 (2016)
43. Wang, S., Wang, Y., Tang, J., Shu, K., Ranganath, S., Liu, H.: What your images reveal: Exploiting visual contents for point-of-interest recommendation. In: Proceedings of WWW, pp. 391–400 (2017)
44. Wang, S., Tang, J., Wang, Y., Liu, H.: Exploring hierarchical structures for recommender systems. IEEE Transactions on Knowledge and Data Engineering (2018)
45. Xin, Y., Jaakkola, T.: Controlling privacy in recommender systems. In: Advances in Neural Information Processing Systems, pp. 2618–2626 (2014)
46. Ying, X., Wu, X., Wang, Y.: On linear refinement of differential privacy-preserving query answering. In: Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II, pp. 353–364 (2013)
47. Zhang, J., Zhang, Z., Xiao, X., Yang, Y., Winslett, M.: Functional mechanism: Regression analysis under differential privacy. Proc. Vldb Endow. **5**(11), 1364–1375 (2012)
48. Zhao, T., McAuley, J.J., King, I.: Leveraging social connections to improve personalized ranking for collaborative filtering. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014. Shanghai, China, November 3-7, 2014, pp. 261–270 (2014)