The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Personalized Privacy-Preserving Social Recommendation

Xuying Meng,[1,2] Suhang Wang,[3] Kai Shu,[3] Jundong Li,[3]
Bo Chen,[4] Huan Liu,[3] Yujun Zhang[1]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Computer Science and Engineering, Arizona State University, Tempe, 85281, USA
[4]Department of Computer Science, Michigan Technological University, Houghton, 49931, USA
{mengxuying, zhmj}@ict.ac.cn, {suhang.wang, kai.shu, jundongl, huan.liu}@asu.edu, bchen@mtu.edu

Abstract

Privacy leakage is an important issue for 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 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 address the problem of achieving privacy-preserving social recommendation under personalized privacy settings. We propose *PrivSR*, a novel framework for privacy-preserving social recommendation, in which users can model ratings and social relationships privately. Meanwhile, by allocating different noise magnitudes to personalized sensitive and non-sensitive ratings, we can protect users' privacy against the 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.

Introduction

The recommender system has become an imperative component of myriad online commercial platforms. With increasing popularity of social networks, recommender systems can take advantage of rich social relationships to further improve effectiveness of recommendation (Tang, Hu, and Liu 2013; Wang et al. 2017; Shu et al. 2018). Despite their effectiveness, these social relationship-based recommender systems (i.e., *social recommendation*), however, may introduce another source of privacy leakage. For example, by observing victim users' ratings on products such as adult or medical items, the attacker may infer the victims' private sex inclinations and health conditions (Fredrikson et al. 2014), which may be even further abused for financial benefits (Nikolaenko et al. 2013).

In practice, a privacy-preserving social recommender system, which can produce accurate recommendation results without sacrificing users' privacy, is very necessary. There are a few mechanisms dedicated along this line. However, most of them suffer from following defects. First, a vast majority of existing efforts (Liu and Terzi 2010; Jorgensen and Yu 2014) 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, some other works (Hoens, Blanton, and Chawla 2010; Tang and Wang 2016) 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 (McSherry and Mironov 2009). In addition, the cryptographic process is usually expensive and may bring large computational overhead. Third, some of the existing works (Machanavajjhala, Korolova, and Sarma 2011; Jorgensen and Yu 2014; Hua, Xia, and Zhong 2015) rely on friends' history ratings to make recommendations. These methods, however, do not differentiate sensitive and non-sensitive ratings and simply treat them equally, which contradicts the real-world scenarios. In practice, social media sites such as IMDB and Facebook[1] allow users to specify the visibility of their ratings on products. Treating all the ratings as equally sensitive and thus not exposing any non-sensitive ratings will make it difficult to attract common-interest friends and make effective recommendations, sacrificing user experience in the long run. Our work actually allows to disclosing the non-sensitive rating, but prevents sensitive ratings from being leaked from the exposed non-sensitive ratings.

Resolving all the aforementioned defects is necessary for building an effective privacy-preserving social recommender system, which is a very challenging task due to the following reasons: First, to eliminate 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 rec-

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

ommender to collaborate together in the course of recommendation. Specifically, users can take part in the learning process upon their own ratings, while the recommender can only have access to non-sensitive ratings, and both parties interact with each other to make the final recommendation. In such a semi-centralized manner however, private information may still be leaked during each interaction, and eliminating such leakage is necessary yet challenging. Second, to avoid using expensive cryptographic techniques, *differential privacy* (Dwork et al. 2006) 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 non-sensitive ratings are exposed and used as background knowledge to infer sensitive ratings. Third, users are often allowed to configure their privacy settings in practice. Due to idiosyncrasy of different users, their personalized privacy settings could be quite diverse. Protecting sensitive ratings based on those personalized diversified privacy settings is not straightforward.

In this work, we perform an initial study of privacy-preserving social recommendation based on personalized privacy settings. In particular, we propose a novel framework, ***PrivSR***, that can protect sensitive ratings 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 divide the ratings into sensitive and non-sensitive ratings, and only attach sensitive ratings with small privacy budgets, i.e. big magnitude noises. In this way, the non-sensitive ratings' 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. The entire process can still satisfy the requirement of differential privacy. We summarize the contributions in the following:

- We are the first to study the problem of privacy-preserving social recommendation with personalized privacy.
- 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 theoretically prove that *PrivSR* can satisfy ϵ-differential privacy, and empirically validate its effectiveness using real-world datasets. The results are encouraging: *PrivSR* provides a good balance between privacy protection and recommendation accuracy.

Preliminaries and Related Work

Differential privacy. Differential privacy (Dwork et al. 2006) 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. Considering its provable privacy guarantee with light computational overhead, we will use differential privacy in our proposed framework.

Definition 1 *ϵ-Differential Privacy (Dwork et al. 2006): 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 $\tilde{D} \in Range(f)$,*

$$Pr[f(D_1) = \tilde{D}] \leq e^{\epsilon} \times Pr[f(D_2) = \tilde{D}] \qquad (1)$$

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 (Dwork et al. 2006) 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 (Kotz, Kozubowski, and Podgorski 2012) 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 $b\sqrt{2hc} \sim Lap(b)$.*

Inference and reconstruction attack. Inference attack is always conducted to infer whether an individual rating is included in the training set (Shokri et al. 2017), while differential privacy is widely used to defend against inference attack (Tang and Wang 2016) 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 (Fredrikson, Jha, and Ristenpart 2015; Komarova, Nekipelov, and Yakovlev 2013). For example, Komarova *et al.* (Komarova, Nekipelov, and Yakovlev 2013) attempted to infer the sensitive features of an individual given fixed statistical estimate from combined public and private sources. Fredrikson *et al.* (Fredrikson et al. 2014) 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.* (Wang, Si, and Wu 2015) were the first to propose to balance the utility and privacy from regression model based on functional mechanism (Zhang et al. 2012).

However, the existing proposed mechanisms can not be applied to handle the reconstruction attack in social recommendation since the way to reconstruct the recommendation

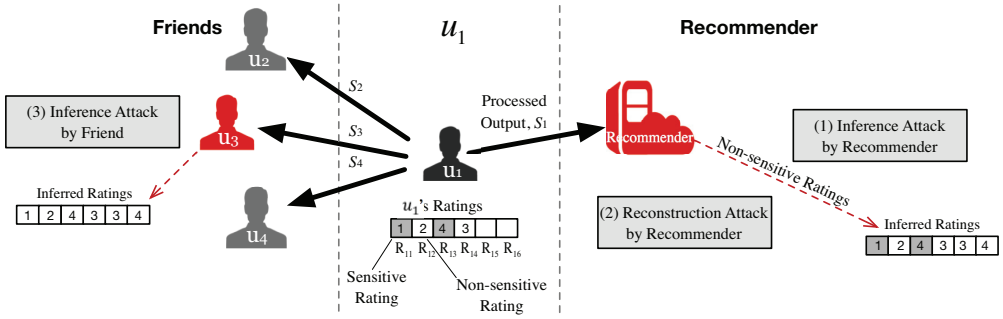


Figure 1: The figure presents an example of user's privacy attacks in social recommendation from the perspective of victim user u_1. Assume there are six items, and u_1 has rated four of them with personalized privacy settings. u_1 exposes processed outputs S_1, S_2, S_3 and S_4 to the recommender and to u_2, u_3 and u_4. The black arrows show the exposure directions. The attackers, who are colored red, conduct attacks as shown in gray boxes, and the red dashed arrows show the process of attacks.

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 (Koren, Bell, and Volinsky 2009).

Social recommendation. Considering users' preferences may be similar or influenced by their friends, social relationships are widely employed to improve recommendation effectiveness based on matrix factorization (Ma et al. 2011), which is selected as our basic model. 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 u_i's rating on item v_j as $\mathbf{R}_{ij}$ and use $\mathcal{F}_i$ to represent the set of u_i's friends. The social recommendation algorithm can be mathematically written as:

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

where $\mathbf{I}_{ij} = 1$ if we observed a rating from u_i to v_j, otherwise $\mathbf{I}_{ij} = 0$. Rating matrix $\mathbf{R} \in \mathbb{R}^{n\times m}$ are decomposed into user latent matrix $\mathbf{U} = [\mathbf{u}_i]_{i\in[n]} \in \mathbb{R}^{K\times n}$ and item latent matrix $\mathbf{V} = [\mathbf{v}_j]_{j\in[m]} \in \mathbb{R}^{K\times m}$, where $\mathbf{u}_i \in \mathbb{R}^K$ and $\mathbf{v}_j \in \mathbb{R}^K$ denote user latent vector for user u_i and item latent vector for v_j respectively, and K is the number of latent dimensions. $||\cdot||_F^2$ denotes the Frobenius norm, and S_{if} is the cosine similarity between ratings of u_i and u_f on the same items, which is applied to regularize the impact of friends' user latent vectors.

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 untrusted, who are curious about or even misuse users' sensitive ratings.

We use a concrete example (as shown in Figure 1) to show some potential privacy leakage. 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 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 being calculated from rating $\mathbf{R}_{1j}$ and user latent vector $\mathbf{u}_1$. However, when there is an item v_j, on which u_1 regards its $\mathbf{R}_{1j}$ as non-sensitive and publishes it, the recommender can obtain S_1 and $\mathbf{R}_{1j}$, compute $\mathbf{u}_1$, and further obtain sensitive ratings $\mathbf{R}_{11}$ and $\mathbf{R}_{13}$; (2) With the exposed non-sensitive ratings, the recommender may conduct reconstruction attack to infer an approximation latent vector $\tilde{\mathbf{u}}_1$, by which u_1's all ratings may be disclosed; 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 computing $\mathbf{u}_1^T\mathbf{V}$.

To formally define our problem, we first describe the notations used in this paper. When the user u_i rates 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 rating*, 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 rating*, 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 ratings. Then we define the set of sensitive ratings as $\mathcal{R}_s = \{\mathbf{R}_{ij}|\forall(i,j)\ s.t.\ \mathbf{F}_{ij} = 1\}$, and the set of non-sensitive ratings 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 ratings* $\mathcal{R}_s$*, as well as a set of non-sensitive ratings* $\mathcal{R}_n$*, we want to infer the missing values in* $\mathbf{R}$ *without privacy leakage of* $\mathcal{R}_s$*.*

Private Social Recommendation

Our proposed framework, *PrivSR*, aims to allow recommender systems to incorporate social relationships without leaking sensitive ratings 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 ratings. We divide the entire framework into *users' ratings* component and *social relationships* compo-

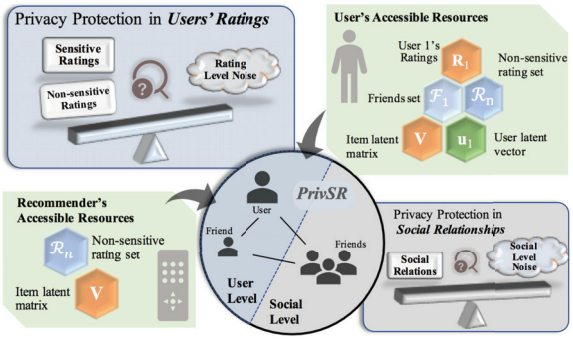


Figure 2: The proposed framework – *PrivSR*.

nent (Figure 2), and keep balanced noise perturbation on sensitive and non-sensitive ratings in users' ratings component, and meanwhile, only utilize non-sensitive ratings 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 require the recommender and the users to collaborate to perform recommendation. We allocate different resources to the recommender and individual users as shown in the green part of Figure 2, in which the recommender can only have access to non-sensitive ratings $\mathcal{R}_n$ 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 ratings $\mathbf{R}_i$ and friends set $\mathcal{F}_i$, in local machine. In particular, since the user latent vector $\mathbf{u}_i$ can be used to obtain sensitive ratings (e.g., by computing $\mathbf{u}_i^T\mathbf{V}$), $\mathbf{u}_i$ should be also kept locally.

Modeling Sensitive and Non-sensitive Ratings

A non-trivial task for our *PrivSR* design is to model ratings without leakage of sensitive ratings, especially in face of personalized privacy settings and public non-sensitive ratings, which may be used by the adversary as the background information to infer the sensitive ratings. We present the basic model based on matrix factorization model (Koren, Bell, and Volinsky 2009; Wang, Tang, and Liu 2015; Wang et al. 2015; Meng et al. 2018a). Since $\mathbf{I}_{ij} = \mathbf{F}_{ij} + \mathbf{G}_{ij}$, the objective function considering rating sensitivity can be written as follows:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n}\sum_{j=1}^{m}(\mathbf{F}_{ij}+\mathbf{G}_{ij})(\mathbf{R}_{ij}-\mathbf{u}_i^T\mathbf{v}_j)^2 \tag{3}$$

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 Eq.(3) 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 Eq.(3) 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: (1) Attackers can easily obtain $\mathbf{u}_i$ when $\mathbf{G}_{ij}=1$, then all sensitive ratings are exposed by $\mathbf{u}_i^T\mathbf{V}$; and (2) Attackers can conduct inference attack from the contribution of a particular user u_i. Although encryption techniques may solve the first 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 as the aforementioned second problem. To tackle all these problems, we apply the objective perturbation method (Chaudhuri, Monteleoni, and Sarwate 2011) with ϵ-differential privacy, and perturb individual's involvement by adding noise into the objective function. We then introduce noise to Eq.(3) as:

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

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 third privacy concerns that attackers can still obtain users' sensitive ratings easily with the exposed non-sensitive ratings by performing reconstruction attack. This can be prevented only when privacy budget ϵ for noise sampling is extremely small (Fredrikson et al. 2014), 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 ratings as:

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

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 ratings respectively. We allocate a much smaller privacy budget ϵ_s for sensitive ratings 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 ratings 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 ratings 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) ratings. 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.*[2]

Modeling Social Relationships

Social regularization, which is formulated as $\sum_{i=1}^n\sum_{f\in\mathcal{F}_i} S_{if}||\mathbf{u}_i - \mathbf{u}_f||_F^2$ based on Eq.(2), requires calculating similarity S_{if} for all the sensitive and non-sensitive ratings, 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 ratings from untrusted friends, we only utilize non-sensitive ratings 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 $\mathbf{u}_f$

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^n \sum_{f\in\mathcal{F}_i} \left(S_{if}||\mathbf{u}_i - \mathbf{u}_f||_F^2 + \mathbf{u}_i^T\mathbf{q}_i^f\right) \tag{6}$$

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 Eq.(6). 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 $2\sqrt{K\mathbf{h}_i}\mathbf{c}_{if}/\epsilon$, and the summation of noise $\sum_{f\in\mathcal{F}_i}(2\sqrt{K\mathbf{h}_i}\mathbf{c}_{if}/\epsilon) \sim Lap(2\sqrt{K}/\epsilon)$.

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.*[3]

[2]Detailed proof can be found in (Meng et al. 2018b)

[3]Detailed proof can be found in (Meng et al. 2018b)

Algorithm 1 *PrivSR* Algorithm

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

Initialize $\mathbf{U}$ and $\mathbf{V}$
while not converge **do**
  for $j = 1, ..., m$ **do**
    // Calculate $\mathbf{v}_j$ on recommender's side
    for i *in* $\mathcal{R}_{s,j}$ **do**
      Send $2(\mathbf{u}_i^T\mathbf{v}_j - \mathbf{R}_{ij})\mathbf{u}_i + \mathbf{x}_j^i$ to the recommender
    for i *in* $\mathcal{R}_{n,j}$ **do**
      Send $2(\mathbf{u}_i^T\mathbf{v}_j - \mathbf{R}_{ij})\mathbf{u}_i + \mathbf{y}_j^i$ to the recommender
    Update $\mathbf{v}_j$ as $\mathbf{v}_j = \mathbf{v}_j - \gamma\frac{\partial\mathcal{J}}{\partial\mathbf{v}_j}$
  for $i = 1, ..., n$ **do**
    // Calculate $\mathbf{u}_i$ on user u_i's side
    for f in $\mathcal{F}_i$ **do**
      Send $\mathbf{q}_i^f - 2S_{if}\mathbf{u}_f$ to user u_i
    Update $\mathbf{u}_i$ as $\mathbf{u}_i = \mathbf{u}_i - \gamma\frac{\partial\mathcal{J}}{\partial\mathbf{u}_i}$
Return $\widehat{\mathbf{R}} = \mathbf{U}^T\mathbf{V}$

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 ratings while retaining recommendation effectiveness is to solve the following optimization problem:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} \mathcal{J} =& \sum_{i=1}^n\sum_{j=1}^m \mathbf{F}_{ij}\left((\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2 + \mathbf{v}_j^T\mathbf{x}_j^i\right)\\ &+\sum_{i=1}^n\sum_{j=1}^m \mathbf{G}_{ij}\left((\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2 + \mathbf{v}_j^T\mathbf{y}_j^i\right)\\ &+\alpha\sum_{i=1}^n\sum_{f\in\mathcal{F}_i}\left(S_{if}||\mathbf{u}_i - \mathbf{u}_f||_F^2 + \mathbf{u}_i^T\mathbf{q}_i^f\right)\\ &+\lambda(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2)\end{aligned} \tag{7}$$

where α is a scalar to control the contribution of social relationships and $\lambda(||\mathbf{U}||_F^2+||\mathbf{V}||_F^2)$ is used to avoid over-fitting with λ being a scalar. We use gradient descent to minimize the objective function. The gradients of Eq.(7) w.r.t. $\mathbf{u}_i$ and $\mathbf{v}_j$ are given as follows:

$$\begin{aligned}\frac{\partial\mathcal{J}}{\partial\mathbf{v}_j} =& \sum_{i=1}^n \mathbf{F}_{ij}\left(2(\mathbf{u}_i^T\mathbf{v}_j - \mathbf{R}_{ij})\mathbf{u}_i + \mathbf{x}_j^i\right)\\ &+\sum_{i=1}^n \mathbf{G}_{ij}\left(2(\mathbf{u}_i^T\mathbf{v}_j - \mathbf{R}_{ij})\mathbf{u}_i + \mathbf{y}_j^i\right) + 2\lambda\mathbf{v}_j\end{aligned} \tag{8}$$

$$\begin{aligned}\frac{\partial\mathcal{J}}{\partial\mathbf{u}_i} =& 2\sum_{j=1}^m \mathbf{I}_{ij}(\mathbf{u}_i^T\mathbf{v}_j - \mathbf{R}_{ij})\mathbf{v}_j + 2\alpha\sum_{f\in\mathcal{F}_i} S_{if}(\mathbf{u}_i - \mathbf{u}_f)\\ &+\alpha\sum_{f\in\mathcal{F}_i}\mathbf{q}_i^f + 2\lambda\mathbf{u}_i\end{aligned} \tag{9}$$

To address the challenge of protecting sensitive ratings 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 9 with Eq.(8), 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 11 through 14 with Eq.(9), 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 $\mathbf{V}$.

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 $\tilde{\sigma}_j^i(t) - \tilde{\sigma}_j^i(t-1) = \sigma_j^i(t) - \sigma_j^i(t-1)$, the impact of noise is eliminated. Therefore, we need to ensure $\mathbf{x}_j^i$ is randomly sampled in each iteration to eliminate the influence of statistics (Rajkumar and Agarwal 2012). Similarly, $\mathbf{y}_j^i$ and $\mathbf{q}_i^f$ will also be updated in each iteration.

Security analysis. 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.*[4]

Experimental Evaluation

In this section, we conduct experimental evaluation to validate the effectiveness of *PrivSR*. We aim to answer two questions: (1) can *PrivSR* improve recommendation effectiveness by incorporating sensitive ratings and social relationships? and (2) can it protect sensitive ratings under reconstruction attack while retaining recommendation effectiveness? In the following, we first introduce our datasets and experimental settings, and then conduct experimental evaluation followed by analyzing impacts of parameters.

Datasets and experimental settings. Two publicly available datasets Ciao[5] and Epinions[6] are used for evaluation. For both datasets, users can rate products from 1 to 5 and 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.

Table 1: Statistics of datasets

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

We use a popular metric Mean Absolute Error (MAE), which is defined as $\sum_{(u_i,v_j)\in\mathcal{R}} |\widehat{\mathbf{R}}_{ij} - \mathbf{R}_{ij}|/|\mathcal{R}|$, and $\mathcal{R}$ is the set of ratings in the testing set. For recommendation effectiveness, smaller MAE indicates better performance. For reconstruction attack on sensitive rating, larger MAE indicates better privacy protection. Note that previous work demonstrated that small improvement in MAE can have a significant impact on the quality of the top-few recommendation (Koren 2008). We compare three representative state-of-the-art recommendation approaches:

- **MF**: matrix factorization (MF) tries to decompose the user-item rating matrix into two matrices for recommendation (Koren, Bell, and Volinsky 2009).
- **SoReg**: this method incorporates social regularization on matrix factorization to represent the social constrains on recommender systems (Ma et al. 2011).
- **DPMF**: differential private matrix factorization (DPMF) treats all ratings private and uses equally perturbed noise for latent matrix learning (Hua, Xia, and Zhong 2015).

For each approach, 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 $K = 10$. For convenience, we fix $\beta = 0.1$ for *PrivSR* in the first two experiments, and accordingly, $\epsilon_s = 1.1\epsilon$ and $\epsilon_n = 11\epsilon$. More details about parameter selection will be discussed in the following.

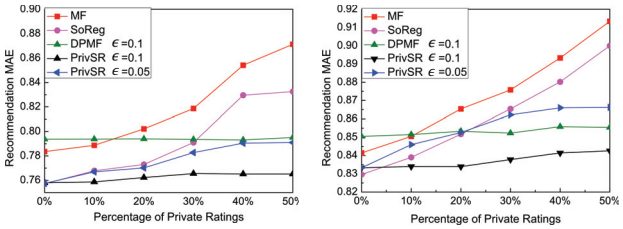
Recommendation effectiveness comparison. To answer the first question, we evaluate the recommendation effectiveness on the test datasets. We do not provide sensitive ratings to MF and SoReg, and only provide them to DPMF and *PrivSR*, since there is no privacy protection for sensitive ratings in MF and SoReg. The average MAE results are shown in Figure 3, 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.
- In general, *PrivSR* with $\epsilon = 0.1$ steadily outperforms other methods with different percentages of sensitive ratings, though we attach noise with low privacy budgets. This confirms the effectiveness of the well-balanced privacy budgets for sensitive and non-sensitive ratings.
- 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 still better than DPMF in most cases. In practice, the percentage of sensitive ratings is usually not too high, thus *PrivSR* can still achieve very good recommendation effectiveness.

[4]Detailed proof can be found in (Meng et al. 2018b)

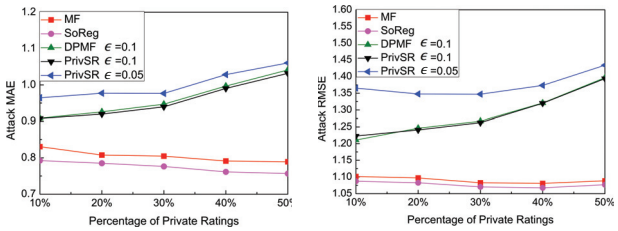
[5]http://www.ciao.co.uk/

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



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

Figure 3: Recommendation effectiveness comparison.



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

Figure 4: Privacy protection comparison.

Based on the aforementioned observations, we conclude that *PrivSR* outperforms the state-of-the-art recommender systems on recommendation effectiveness by utilizing rich social relationships and perturbing balanced noises.

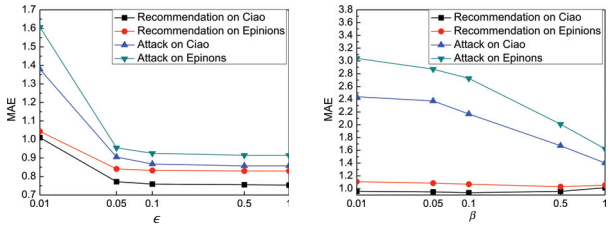
Privacy protection comparison. To answer the second question, we simulate the reconstruction attack. There are multiple options for conducting reconstruction attack (Fredrikson, Jha, and Ristenpart 2015). We conduct it using the matrix factorization-based model. Since attackers can obtain both $\mathbf{V}$ and $\mathcal{R}_n$, they can infer a rough user latent profile $\tilde{\mathbf{u}}_i$ of the victim u_i by solving the following equation:

$$\min_{\tilde{\mathbf{U}}} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{G}_{ij}(\mathbf{R}_{ij} - \tilde{\mathbf{u}}_i^T \mathbf{v}_j)^2 \tag{10}$$

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 value on sensitive ratings represents a better privacy protection. From Figure 4, 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 much 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 significant. These results indicate *PrivSR* can achieve a better balance between privacy protection and recommendation effectiveness.
- *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 not too large.

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.



(a) MAE when varying ϵ (b) MAE when varying β

Figure 5: MAE with varying parameters.

Impact of parameters ϵ and β. For simplification, we set $x = 10$, based on the real-world statistical results[7]. Then 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 MAE results are shown in Figure 5, from which we observe that: 1) Larger privacy 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 (Meng et al. 2016; Koren, Bell, and Volinsky 2009; Wang, Si, and Wu 2015). 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.

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 recommendation effectiveness. Theoretic analysis and experimental evaluation on real-world datasets demonstrate the effectiveness of the proposed framework for recommendation as well as privacy protection. Several directions can be further investigated. First, in this paper, we build our model based on matrix factorization, which is a point-based model. We will

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

study privacy preserving social recommendation using rank-based models such as BPR (Rendle et al. 2009) in the future. Second, we only consider static data in this paper. We will study this problem for temporal and dynamic data (Koren 2010) next.

Acknowledgment

This work was supported by, or in part by, National Science Foundation of China (61672500, 61572474 and 61402446), and Program of International S&T Cooperation (2016YFE0121500). Suhang Wang and Huan Liu were supported by National Science Foundation (NSF) under grant #1614576 and Office of Naval Research (ONR) under grant N00014-16-1-2257.

References

Chaudhuri, K.; Monteleoni, C.; and Sarwate, A. D. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12(3):1069–1109.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *TCC*, 265–284.

Fredrikson, M.; Lantz, E.; Jha, S.; Lin, S.; Page, D.; and Ristenpart, T. 2014. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX*, 17–32.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*, 1322–1333.

Hoens, T. R.; Blanton, M.; and Chawla, N. V. 2010. A private and reliable recommendation system for social networks. In *SocialCom*, 816–825.

Hua, J.; Xia, C.; and Zhong, S. 2015. Differentially private matrix factorization. In *IJCAI*, 1763–1770.

Jorgensen, Z., and Yu, T. 2014. A privacy-preserving framework for personalized, social recommendations. In *EDBT*, 571–582.

Komarova, T.; Nekipelov, D.; and Yakovlev, E. 2013. Estimation of treatment effects from combined data: Identification versus data security. In *Iccas-Sice*, 3066–3071.

Koren, Y.; Bell, R. M.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* 42(8):30–37.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, 426–434.

Koren, Y. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53(4):89–97.

Kotz, S.; Kozubowski, T.; and Podgorski, K. 2012. *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media.

Liu, K., and Terzi, E. 2010. A framework for computing the privacy scores of users in online social networks. *TKDD* 5(1):6:1–6:30.

Ma, H.; Zhou, D.; Liu, C.; Lyu, M. R.; and King, I. 2011. Recommender systems with social regularization. In *WSDM*, 287–296.

Machanavajjhala, A.; Korolova, A.; and Sarma, A. D. 2011. Personalized social recommendations: accurate or private. *PVLDB* 4(7):440–450.

McSherry, F., and Mironov, I. 2009. Differentially private recommender systems: building privacy into the net. In *SIGKDD*, 627–636.

Meng, X.; Xu, Z.; Chen, B.; and Zhang, Y. 2016. Privacy-preserving query log sharing based on prior n-word aggregation. In *Trustcom*, 722–729.

Meng, X.; Wang, S.; Liu, H.; and Zhang, Y. 2018a. Exploiting emotion on reviews for recommender systems. In *AAAI*.

Meng, X.; Wang, S.; Shu, K.; Jundong, L.; Chen, B.; Liu, H.; and Zhang, Y. 2018b. Personalized privacy-preserving social recommendation. https://github.com/mxyenguing/PrivSR/blob/master/appendix.pdf.

Nikolaenko, V.; Ioannidis, S.; Weinsberg, U.; Joye, M.; Taft, N.; and Boneh, D. 2013. Privacy-preserving matrix factorization. In *CCS*, 801–812.

Rajkumar, A., and Agarwal, S. 2012. A differentially private stochastic gradient descent algorithm for multiparty classification. In *AISTATS*, 933–941.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 452–461.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *SP*, 3–18.

Shu, K.; Wang, S.; Tang, J.; Wang, Y.; and Liu, H. 2018. Crossfire: Cross media joint friend and item recommendations. In *WSDM*.

Tang, Q., and Wang, J. 2016. Privacy-preserving friendship-based recommender systems. *IEEE Transactions on Dependable & Secure Computing* PP(99):1–1.

Tang, J.; Hu, X.; and Liu, H. 2013. Social recommendation: a review. *Social Netw. Analys. Mining* 3(4):1113–1133.

Wang, S.; Tang, J.; Wang, Y.; and Liu, H. 2015. Exploring implicit hierarchical structures for recommender systems. In *IJCAI*, 1813–1819.

Wang, S.; Wang, Y.; Tang, J.; Shu, K.; Ranganath, S.; and Liu, H. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *WWW*, 391–400.

Wang, Y.; Si, C.; and Wu, X. 2015. Regression model fitting under differential privacy and model inversion attack. In *IJCAI*, 1003–1009.

Wang, S.; Tang, J.; and Liu, H. 2015. Toward dual roles of users in recommender systems. In *CIKM*, 1651–1660.

Zhang, J.; Zhang, Z.; Xiao, X.; Yang, Y.; and Winslett, M. 2012. Functional mechanism: regression analysis under differential privacy. *PVLDB* 5(11):1364–1375.