Differentially Private Implicit Matrix Factorization

Xun Ran[1], Qingqing Ye[1,*], Xin Huang[2], Jianliang Xu[2], Haibo Hu[1]
[1]The Hong Kong Polytechnic University
qi-xun.ran@connect.polyu.hk;qqing.ye@polyu.edu.hk;haibo.hu@polyu.edu.hk
[2]Hong Kong Baptist University
xinhuang@comp.hkbu.edu.hk;xujl@comp.hkbu.edu.hk

ABSTRACT

Implicit Matrix Factorization (IMF) refers to Matrix Factorization (MF) based on users' implicit data (i.e., clients' actions or inactions). It serves as the backbone of many recommender systems for handling implicit feedback, such as webpage visis or bookmarks. Since these methods require a large amount of user data to provide accurate recommendations, data privacy has become a significant concern. Although the Differential Privacy (DP) technique has been widely applied to MF to protect explicit data, the resulting utility loss makes it challenging to successfully incorporate DP into IMF. In this study, we design a differentially private IMF, namely DPIMF, using objective perturbation. To enhance utility, we redesign the loss function and adopt the importance sampling to limit the noise scale in IMF. We provide formal utility guarantees for the proposed schemes and theoretically analyze the conditions for ensuring utility improvement, offering the optimal settings. Experimental results on three benchmark datasets validate our theoretical conclusions, demonstrating that the proposed schemes achieve a better trade-off between privacy and recommendation accuracy compared to state-of-the-art methods.

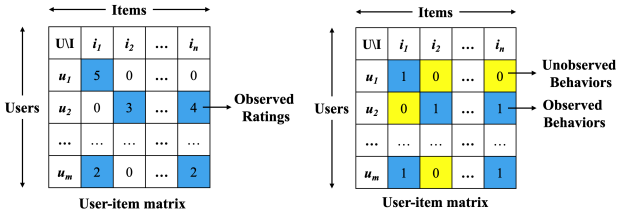
1 INTRODUCTION

As one of the most successful techniques for implementing recommender systems, Matrix Factorization (MF) predicts a target user's preference for items based on the historical behavior data from all users. In many recommender systems, generally user preferences are explicitly represented as numerical ratings (e.g., 1 to 5 stars in Netflix [11]). However, implicit feedback such as purchased items or browsing history is prevalent in many more situations due to its rich source and easy collection, as shown in Figure 1. MF in this setting is known as the Implicit Matrix Factorization (IMF) problem, which has been extensively researched over the past two decades [13, 28, 31, 39] and widely-deployed in a number of commercialized systems [1, 5, 16].

As the other side of the coin of using MF for recommendation, the users' privacy is at risk [9, 17, 19]. A privacy breach of MF can happen in various ways since an attacker can take advantage of model parameters [6, 8, 20] or even merely the recommendation results of MF [3, 24]. Differential Privacy (DP) has received significant attention in incorporating it into MF-based methods to mitigate the privacy risks. However, current studies primarily focus on developing differentially private MF for explicit feedback [2, 4, 14, 22], leaving a noticeable gap in research concerning implicit feedback. To bridge the gap, our work focuses on learning IMF for accurate recommendation while preserving differential privacy, which leads to our method for differentially private implicit matrix factorization (DPIMF).

(a) Explicit Feedback (b) Implicit Feedback

Figure 1: Examples of explicit and implicit feedbacks.



(a) Explicit Feedback (b) Implicit Feedback

Figure 2: Comparison between user-item matrices of explicit feedback and implicit feedback.

Table 1 summarizes previous relevant studies in terms of three aspects of privacy setting. ❶ **Object of Privacy Protection** refers to the observed user behaviors, including the items a user rated and the value of those ratings [4, 9, 22]. Some works focus on the protection of rating values only [2, 14, 15]. ❷ **Granularity of Privacy Protection** can be a user's single rating (i.e., per-record) [2, 9, 14], or a user's entire ratings (i.e., per-user) [4, 15, 22]. ❸ **DP Framework** adopted by the previous methods includes the strictest DP (i.e., ϵ-DP) [2, 9, 14] or the relaxed DP such as (ϵ, δ)-DP [22], or Joint DP [4, 15].

As opposed to previous studies, our work DPIMF considers a different object of privacy protection for both observed and unobserved behaviors. It is noteworthy that the existing methods cannot be applied to achieve DPIMF since they can handle observed behaviors only. Figure 2 shows a comparison between user-item matrices of explicit and implicit feedback. In the case of implicit feedback, the observed behavior (blue entries in the user-item matrix) is typically positive and contrasted with all unobserved behaviors (yellow

* Corresponding author

Table 1: Comparison of different privacy setting DPMF methods (✓ and × denote the method supports this property or not).

Method	Object of privacy protection			Granularity of privacy protection		DP Framework	
	Observed ratings	Observed behaviors	Unobserved	Per-record	Per-user	Relaxed DP	Rigorous DP
DPSGD;DPALS;DPInput [2]	✓	×	×	✓	×	✓	✓
DPSGD+;DPALS+ [9]	✓	✓	×	✓	×	✓	✓
DPMF [14]	✓	×	×	✓	×	✓	✓
DPFW [15]	✓	×	×	✓	✓	✓	×
PALS [4]	✓	×	×	✓	✓	✓	×
FMF [22]	✓	✓	×	✓	✓	✓	×
Ours (DPIMF)	✓	✓	✓	✓	✓	✓	✓

entries in the user-item matrix). The protection of unobserved behaviors is necessary, since they may indicate a user's personal information (e.g., gender) by revealing a user's lack of interest in certain items.

Our goal in DPIMF is to protect a user's all behaviors (i.e., per-user) in compliance with the strictest DP (i.e., ϵ) to ensure a rigorous privacy protection for individual data. Under this privacy requirement, the trivial extension of the previous methods to IMF leads to a significant loss in utility. Specifically, privacy mechanism is required to mask a user's entire behaviors, rather than just the observed ones. Even worse, due to the extreme sparsity of the data in recommender systems, a user's entire behavior is far more than the observed ones. Therefore, the noise scale of DP mechanism, will be unacceptably high and the utility of recommendation will be degraded by the severe perturbation.

Resolving the aforementioned problem paves the way to practical privacy-preserving recommender systems. For this aim, this paper first presents an IMF-based approach preserving DP. An objective perturbation-based method is adopted to privately learn and share the model. For enhancing utility, we propose several strategies built upon the following observations. First, the noise scale can be reduced by lowering the sensitivity, which is the maximum impact of a change in private object. Based on this idea, the loss function of IMF is redesigned without significantly sacrificing model utility. We found, in response to changes in the implicit data, the effects on the two loss terms — defined over observed and unobserved behaviors — can offset each other, which significantly reduces sensitivity under appropriate parameter settings.

Another strategy to reduce the noise scale is to enable privacy amplification, which reduces the injected noise but without increasing the overall privacy loss. Sampling methods have been shown effectiveness in achieving privacy amplification effect. However, the traditional uniform sampling-based methods cannot be directly applied to our setting. Due to the uneven distribution of implicit data, uniform sampling can lead to the loss of informative points, impacting the effectiveness of model training. To mitigate this problem, we designed an importance sampling-based method. The privacy amplification of importance sampling is analyzed and the effectiveness of the above strategies is theoretically proven and empirically validated.

The contributions of our work are summarized as follows:

(1) To our knowledge, DPIMF is the first work studying differentially private matrix factorization for the implicit feedback. To reduce the high sensitivity and alleviate the utility loss, we redesign the loss function which well suppresses the sensitivity, and implement symmetric noise which effectively preserves the characteristic of the loss before perturbed.
(2) DPIMF is the first to incorporate importance sampling to DP-enabled MF, resulting in an effective privacy amplification. By coupling the sampling probabilities to the individual privacy loss and/or a suitable measure of "informativeness", we improve the privacy and accuracy of the subsampled mechanism simultaneously.
(3) We provide the utility guarantees of DPIMF, incorporating each utility-enhanced technique, which offers an in-depth understanding of its theoretical properties. Furthermore, we prove the proposed schemes preserve the rigorous ϵ-DP.
(4) Through extensive evaluation over real-world datasets, DPIMF is shown to achieve the best recommendation accuracy among state-of-the-art solutions. The results validate our theoretical findings and demonstrate the effectiveness in improving model utility.

The remainder of this paper is organized as follows. In Section 2, we introduces preliminaries and the studied problem. Section 3 presents the proposed scheme in detail and Section 5 gives the theoretical analysis of the proposed scheme. The experimental results are reported and analyzed in Section 6. In Section 8, the conclusion is drawn.

2 PRELIMINARIES AND PROBLEM FORMULATION

2.1 Implicit Matrix Factorization

The task of IMF is to predict a target user's preference for unseen items while the training dataset only consists of binary data reflecting users' observed and unobserved data. The dataset is typically extremely sparse, and unobserved behaviors are non-positive data mixed with both negative and potential positive data. Matrix factorization is one of the most popular technique to handle the implicit feedback [30, 32].

Let $\boldsymbol{U}$ and $\boldsymbol{I}$ be the sets of users and items in a recommender system. The implicit data can be expressed as a matrix $\boldsymbol{R} \in \mathbb{R}^{|\boldsymbol{U}|\times|\boldsymbol{I}|}$, where its entry $r_{ui} \in \{1, 0\}$ ($u \in [1, |\boldsymbol{U}|]$, $i \in [1, |\boldsymbol{I}|]$) records whether the user u interacts with the item i or not. MF-based models factorize this binary matrix $\boldsymbol{R}$ into the user and item profile matrices $\boldsymbol{P} \in \mathbb{R}^{d\times|\boldsymbol{U}|}$, $\boldsymbol{Q} \in \mathbb{R}^{d\times|\boldsymbol{I}|}$, where d is the latent dimension

For MF-based methods, the difference is that $\boldsymbol{R}$ is not binary but contains numerical rating scores. All subsequent MF optimizations are changed accordingly.

satisfying $d \ll \min(|U|, |I|)$. The interacting probability of user u on item i is estimated by $\boldsymbol{p}_u^T \boldsymbol{q}_i$, where $\boldsymbol{p}_u \in \mathbb{R}^d$ and $\boldsymbol{q}_i \in \mathbb{R}^d$ are column vectors (of $\boldsymbol{P}$ and $\boldsymbol{Q}$) that represent the user and item profiles, respectively.

To learn $\boldsymbol{P}$ and $\boldsymbol{Q}$, the state-of-the-art methods [31, 32] suggest to minimize the following loss function:

$$L(\boldsymbol{P}, \boldsymbol{Q}) = \sum_{(u,i) \in S} (\boldsymbol{p}_u^T \boldsymbol{q}_i - 1)^2 + \alpha_0 \sum_{(u,i) \in \Omega} (\boldsymbol{p}_u^T \boldsymbol{q}_i)^2 + R(\boldsymbol{P}, \boldsymbol{Q}) \quad (1)$$

Here, Ω is the universe of all user-item pairs (i.e., $|\Omega| = |\boldsymbol{U}| \times |\boldsymbol{I}|$), and S includes the pairs of only positive data. The first term of $L(\boldsymbol{P}, \boldsymbol{Q})$ measures the distance between predictions for positive data in S and the ground-truth label (i.e., 1). The second term is defined over Ω and the trade-off between the two terms is controlled by α_0. To prevent overfitting, a regularization term $R(\boldsymbol{P}, \boldsymbol{Q})$ is introduced to penalize the energy of $\boldsymbol{P}$ and $\boldsymbol{Q}$ as

$$R(\boldsymbol{P}, \boldsymbol{Q}) = \lambda \left(\sum_{u \in \boldsymbol{U}} (|\boldsymbol{I}_u| + \alpha_0 |\boldsymbol{I}|) \|\boldsymbol{p}_u\|_2^2 + \sum_{i \in I} (|\boldsymbol{U}_i| + \alpha_0 |\boldsymbol{U}|) \|\boldsymbol{q}_i\|_2^2 \right), \quad (2)$$

where $\boldsymbol{I}_u = \{i : (u, i) \in S\}$, $\boldsymbol{U}_i = \{u : (u, i) \in S\}$. Eq. (2) is known as the frequency-based regularizer, which imposes heavier regularization on frequent items and users, and its strength is tuned by λ.

To minimize $L(\boldsymbol{P}, \boldsymbol{Q})$ in Eq. (1), one of the most commonly used algorithms is alternating least squares (ALS) algorithm which optimizes $\boldsymbol{P}$ and $\boldsymbol{Q}$ alternatively [2]. The rule of ALS for updating each user and item profile is as follows:

$$\boldsymbol{p}_u \leftarrow \left(\sum_{i \in \boldsymbol{I}_u} \boldsymbol{q}_i \otimes \boldsymbol{q}_i + \alpha_0 \sum_{i \in \boldsymbol{I}} \boldsymbol{q}_i \otimes \boldsymbol{q}_i + \rho_u \boldsymbol{E} \right)^{-1} \sum_{i \in \boldsymbol{I}_u} \boldsymbol{q}_i, \quad (3)$$

$$\boldsymbol{q}_i \leftarrow \left(\sum_{u \in \boldsymbol{U}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \alpha_0 \sum_{u \in \boldsymbol{U}} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \rho_i \boldsymbol{E} \right)^{-1} \sum_{i \in \boldsymbol{U}_i} \boldsymbol{p}_u, \quad (4)$$

where $\rho_u = \lambda(|\boldsymbol{I}_u| + \alpha_0 |\boldsymbol{I}|)$, $\rho_i = \lambda(|\boldsymbol{U}_i| + \alpha_0 |\boldsymbol{U}|)$, $u \in \boldsymbol{U}$, $i \in \boldsymbol{I}$, and $\boldsymbol{E} \in \mathbb{R}^{d \times d}$ is an identity matrix. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $\boldsymbol{x} \otimes \boldsymbol{y} \in \mathbb{R}^{d \times d}$ is denoted as their outer product.

2.2 Differential Privacy

Differential Privacy (DP) refers to the principle that releasing an aggregated report of a dataset should not allow the attacker to infer more information about any individual other than his prior knowledge about the individual. To achieve this principle, DP requires the outcome of a computation to be insensitive to any particular record in the input dataset. The formal definition of DP is as follows:

Definition 1. *(ε-differential privacy [7].) An algorithm $\mathcal{A}$ satisfies ε-differential privacy (ε-DP), where $\varepsilon \geq 0$, if and only if for $\forall\, O \subseteq Range(\mathcal{A})$ and any neighboring datasets D and D', it satisfies*

$$\Pr[\mathcal{A}(D) \in O] \leq \exp(\varepsilon) \Pr\left[\mathcal{A}(D') \in O\right], \quad (5)$$

where $\exp(\cdot)$ denotes the exponential function and $Range(\mathcal{A})$ denotes the set of all possible outputs of the algorithm $\mathcal{A}$.

In DP, ε is the privacy budget that controls the privacy protection strength, i.e., a larger ε yields weaker privacy protection, while a smaller ε leads to stronger protection. In MF-based methods, D and D' are neighboring if they differ in the values of observed ratings (e.g., changing the rating stars in Netflix [11]). However, in IMF, D and D' are neighboring if they differ in a user's entire behaviors (i.e., an entire row in user-item matrix). As shown below, the impact of this difference of neighboring datasets influences the sensitivity analysis, which is a critical concept for ensuring DP.

Definition 2. *(Sensitivity [7].) For a function $f : D \rightarrow \mathbb{R}$, the sensitivity of f, denoted as Δ_f, is defined as:*

$$\Delta_f = \max_{D,D'} \left\| f(D) - f(D') \right\|_1, \quad (6)$$

where D and D' are neighboring datasets.

The Laplace mechanism is a prevailing mechanism for implementing DP, which adds noises to the outcome of a query. In particular, it generates noises from the Laplace distribution according to the sensitivity of a query. The Laplace mechanism can be formally defined as:

Definition 3. *(Laplace mechanism [7].) Given a function $f : D \rightarrow \mathbb{R}^d$, the following mechanism $\mathcal{A}$ satisfies ε-DP:*

$$\mathcal{A}(D) = f(D) + Lap\left(\frac{\Delta_f}{\varepsilon}\right)^d, \quad (7)$$

where $Lap(b)$ denotes a random variable drawn from a Laplace distribution with zero mean and scale b, Δ_f is the sensitivity of f and d represents the dimension of f.

In practical applications of DP, an algorithm is often a combination of a series of operations over the same dataset or disjoint subsets of a single dataset [7]. For these cases, the overall privacy protection level the algorithm provides can be quantified through the following theorems.

Theorem 1. *(Parallel composition [25].) Let $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k$ be k algorithms that satisfy ε_1-DP, ε_2-DP, $\cdots$, ε_k-DP, respectively. $D_1, D_2 \cdots D_k$ are k disjoint partitions of a dataset D, where $D_1 \cup D_2, \cdots, \cup D_k = D$. Then publishing $\mathcal{A}_1(D_1), \mathcal{A}_2(D_2), \cdots, \mathcal{A}_k(D_k)$ satisfies $\max_{i \in \{1, \cdots, k\}} \{\varepsilon_i\}$-DP.*

Theorem 2. *(Sequential composition [25].) Let $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k$ be k algorithms that satisfy ε_1-DP, ε_2-DP, $\cdots$, ε_k-DP, respectively, with respect to the input dataset D. Publishing the output $\boldsymbol{t} = (t_1, t_2, \cdots, t_k)$ satisfies $\left(\sum_{i=1}^k \varepsilon_i\right)$-DP, where $t_1 = \mathcal{A}_1(D)$, $t_2 = \mathcal{A}_2(t_1, D)$, $t_k = \mathcal{A}_k((t_1, t_2, \cdots, t_{k-1}), D)$.*

As per Definition 2, for IMF, the sensitivity is the maximum change to the solution of Eq. (1) when a user's entire behaviors are changed. For this fact, the sensitivity of IMF is higher than MF with explicit data which only needs to hide the value of observed behaviors. From Definition 3, for a fixed privacy budget ε, the sensitivity determines the amount of noise needed to mask the change to the solution of Eq. (1). The large noise brought by high sensitivity degrades the recommendation accuracy.

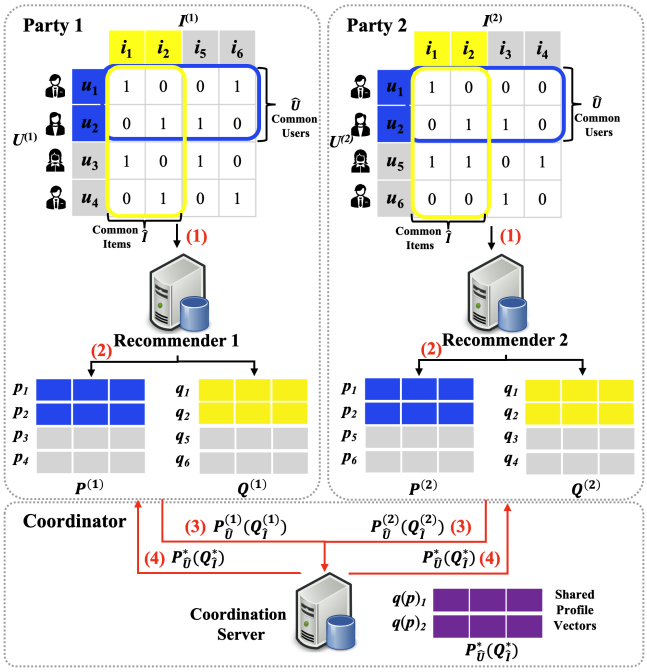


Figure 3: Scenario of DPIMF.

2.3 Problem Formulation

Our method works in the setting of Federated learning (FL), where multiple parties collaboratively train a recommendation model without directly sharing their data. All parties share a common subset of users or items, but each owns different user behaviors. A typical application for this scenario is that two companies, e.g., an online content platform and a local retailer, share some common users but each has different user-behavior information; they want to collaboratively train a recommendation model that learns from both users' behaviors on online content (e.g., videos) and items in local store. There may be an overlap in the items offered, allowing the model to learn from the feedback of both online and local users for those items.

Figure 3 illustrates this process using an example of two parties. The users of party j, i.e., $\boldsymbol{U}^{(j)}$, provide their implicit feedback on item set $\boldsymbol{I}^{(j)}$ to the trusted recommender of their party (Step (1)). As shown in the figure, $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{I}}$ represent the shared users and items, respectively. The data from the common users and items are highlighted in blue and yellow in the user-item matrix, respectively. Using the data collected from local users, each recommender j trains its own IMF model to obtain user profile and item profile matrices, i.e., $\boldsymbol{P}^{(j)}$ and $\boldsymbol{Q}^{(j)}$ (Step (2)). During the federated learning procedure, different parties exchange the local profile vectors (i.e., $\boldsymbol{P}^{(j)}_{\hat{U}}$ or $\boldsymbol{Q}^{(j)}_{\hat{I}}$) (highlighted in blue and yellow) with the coordination server (Step (3)); the shared profile vectors are aggregated at the coordination server and sent the results (i.e., $\boldsymbol{P}^{*}_{\hat{U}}$ or $\boldsymbol{Q}^{*}_{\hat{I}}$) back to each party to refine local user/item profile vectors for the next iterations (Step (4)).

As for threat model, we assume the coordination server and all recommenders are honest-but-curious, i.e., they follow the protocol we designed but are interested in inferring user data by analyzing the transcript during the communication between recommenders and coordination server (i.e., red arrows in Figure 3).

According to the threat model, learning and publishing user or item profile vectors privately serves as an important building block of a privacy-preserving recommender system in the considered scenario [14, 23, 41]. Indeed, as proved in [43], when the local profile vectors are solved in compliance with DP (Step (3)), the overall procedure (including Step (4)) satisfies DP according to the post-processing theorem [21]. Following this line, the goal of this study is to design a scheme for learning and publishing the private profile vectors, satisfying the rigorous privacy setting shown in Table 1 and preserving the model utility as much as possible.

3 A STRAWMAN SOLUTION FOR DPIMF

In this section, we present a strawman solution for differentially private implicit matrix factorization (DPIMF) using an objective perturbation mechanism. Since the user and item profile matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ are symmetrically defined in the loss function of Eq. (1), the optimization processes for both matrices are identical. For the conciseness and readability, we introduce our method through solving the private item profile matrix $\bar{\boldsymbol{Q}}$ without loss of generality.

Given a specific item i, the non-private loss function of its profile $\boldsymbol{q}_i$ is derived from Eq. (1) as

$$L^{\mathrm{O}}(\boldsymbol{q}_i) = \sum_{u \in \boldsymbol{U}_i} \left(\boldsymbol{p}_u^T \boldsymbol{q}_i - 1\right)^2 + \alpha_0 \sum_{u \in \boldsymbol{U}} \left(\boldsymbol{p}_u^T \boldsymbol{q}_i\right)^2 + \lambda \left(|\boldsymbol{U}_i| + \alpha_0 |\boldsymbol{U}|\right) \|\boldsymbol{q}_i\|_2^2 . \tag{8}$$

where $\boldsymbol{U}$ denotes the universe of users involved in the recommender system. $\boldsymbol{U}_i$ denotes the set of users who interact with item i (which is labelled as 1 in the user-item matrix), and this set is sensitive in nature.

Since the sophisticated form of $L^{\mathrm{O}}(\boldsymbol{q}_i)$ hinders the sensitivity analysis, we expand $L^{\mathrm{O}}(\boldsymbol{q}_i)$ as a polynomial. By omitting the constant term, Eq. (8) can be rewritten as

$$l(\boldsymbol{q}_i) = \boldsymbol{q}_i^T \left(\sum_{u \in \boldsymbol{U}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \alpha_0 \sum_{u \in \boldsymbol{U}} \boldsymbol{p}_u \otimes \boldsymbol{p}_u \right) \boldsymbol{q}_i - \boldsymbol{q}_i^T \left(2 \sum_{u \in \boldsymbol{U}_i} \boldsymbol{p}_u \right) + \lambda \left(|\boldsymbol{U}_i| + \alpha_0 |\boldsymbol{U}|\right) \|\boldsymbol{q}_i\|_2^2 . \tag{9}$$

Note that the change of implicit matrix (either $1 \rightarrow 0$ or $0 \rightarrow 1$) only affects $\boldsymbol{U}_i$ of Eq. (9). Without loss of generality, let $\boldsymbol{U}_i'$ be the set obtained by adding a user v to $\boldsymbol{U}_i$. The sensitivity in terms of $\boldsymbol{U}_i$ and $\boldsymbol{U}_i'$ satisfies

$$\|l_{U_i}(q_i) - l_{U_i'}(q_i)\|_1 \leq \boldsymbol{q}_i^T \|A(\boldsymbol{U}_i) - A(\boldsymbol{U}_i')\|_1 \boldsymbol{q}_i$$

where $\|A(\boldsymbol{U}_i) - A(\boldsymbol{U}_i')\|_1$ measures the difference by the sum of coefficients denoted as $A(\boldsymbol{U}_i)$ and $A(\boldsymbol{U}_i')$, which satisfies the following

inequalities:

$$\begin{aligned}
&\left\| \sum_{1\le j,l\le d} \left(\sum_{u\in U_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u - \sum_{u\in U_i'} \boldsymbol{p}_u \otimes \boldsymbol{p}_u \right)_{jl} \right. \\
&\left. + 2\sum_{j=1}^{d} \left(\sum_{u\in U_i} \boldsymbol{p}_u - \sum_{u\in U_i'} \boldsymbol{p}_u \right)_j + (|U_i| - |U_i'|) \right\|_1 \\
&\le \sum_{1\le j,l\le d} |(\boldsymbol{p}_v \otimes \boldsymbol{p}_v)_{jl}| + 2\sum_{j=1}^{d} |(\boldsymbol{p}_v)_j| + 1 \\
&\le \max_{u\in U} \left(2\|\boldsymbol{p}_u\|_1 + \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{full}} + 1\right).
\end{aligned}$$

Here, $\|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{full}}$ denotes $\sum_{1\le j,l\le d} |(\boldsymbol{p}_u \otimes \boldsymbol{p}_u)_{jl}|$. Thus, the sensitivity on the corresponding coefficients of $l(\boldsymbol{q}_i)$ is given by

$$\Delta = \max_{u\in U} \left(2\|\boldsymbol{p}_u\|_1 + \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{full}} + 1\right). \tag{10}$$

To ensure DP, for a given privacy budget ε, Laplacian noises are sampled and added to $l(\boldsymbol{q}_i)$. The perturbed loss thus becomes

$$\begin{aligned}
\bar{l}(\boldsymbol{q}_i) = \boldsymbol{q}_i^T \left(\sum_{u\in U_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \alpha_0 \sum_{u\in U} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \boldsymbol{B} \right) \boldsymbol{q}_i - \\
\boldsymbol{q}_i^T \left(2\sum_{u\in U_i} \boldsymbol{p}_u + \boldsymbol{b} \right) + \lambda \left(|U_i| + \alpha_0|U| + \eta\right) \|\boldsymbol{q}_i\|_2^2,
\end{aligned} \tag{11}$$

where $\boldsymbol{B} \sim Lap\left(\frac{\Delta}{\varepsilon}\right)^{d\times d}$, $\boldsymbol{b} \sim Lap\left(\frac{\Delta}{\varepsilon}\right)^{d}$ and $\eta \sim Lap\left(\frac{\Delta}{\varepsilon}\right)$. It can be proved that the optimal profile $\boldsymbol{q}_i^* = \arg\min L^{\mathrm{O}}(\boldsymbol{q}_i)$ derived by minimizing Eq. (8) in non-private setting satisfies $\|\boldsymbol{q}_i^*\|_2 \le \sqrt{1/\lambda}$. For brevity, we defer this proof to the appendix.

Therefore, to fulfill the same constraint and enhance the utility, we derive the private profile $\bar{\boldsymbol{q}}_i$ by minimizing $\bar{l}(\boldsymbol{q}_i)$ over the convex set $C = \{\boldsymbol{q}_i \mid \|\boldsymbol{q}_i\|_2 \le \sqrt{1/\lambda}\}$. Repeating the above steps for all items ($i \in [1, |\boldsymbol{I}|]$), we can obtain the private matrix $\bar{\boldsymbol{Q}}$.

Remark. The method discussed above strictly satisfies ε-DP. However, it suffers from high sensitivity. According to Eq. (9), the sensitivity analysis needs to aggregate the effects of the polynomial terms with different orders, as well as the regularization term. Specifically, by letting $c = \max_{u\in U, j\in\{1,2,\ldots,d\}} |p_{uj}|$, Δ is bounded by $(2cd + c^2d^2 + 1)$. This bound is dominated by its quadratic term that can be extremely large when c or d is large. The high sensitivity comes at an expensive cost of model utility.

4 THE FULL-FLEDGED DPIMF

Although the strawman solution works for DPIMF, it still suffers from utility degradation due to the large noise scale required to protect the vast amount of unobserved data. In this section, we address these issues and propose a full-fledged solution for differentially private implicit matrix factorization (DPIMF), resulting in a more accurate recommender system.

4.1 Overview

To limit the noise scale, our solution is based on several key intuitions. First, we observe that the noise scale is determined by the sensitivity, which quantifies how much the loss function responds to changes in implicit data. We aim to explore whether there is an optimal approach to constructing the loss function that better balances sensitivity and utility. Second, considering that the increase in noise scale is due to the larger number of private data, we investigate whether a sampling method can enable training on a subset of the data without compromising utility, while still ensuring differential privacy.

Inspired by this, we have developed two strategies based on the strawman solution — building complementary loss and importance sampling. Both of the two strategies are verified to effectively reduce noise scale and enhance the utility of the method. An overview is illustrated in Figure 4. Using the implicit feedback, we construct a complementary loss defined over the observed and unobserved data (Step (1)). Then, the sensitivity is derived on the complementary loss (Step (2)). The importance weight of the data are solved and utilitzed for importance sampling (Step (3),(4)). By adding noise from Laplacian distribution to the loss after sampling, we obtain a private loss $\bar{L}$ (Step (5)), which then allows us to solve for the private user or item profile matrix (Step (6)).

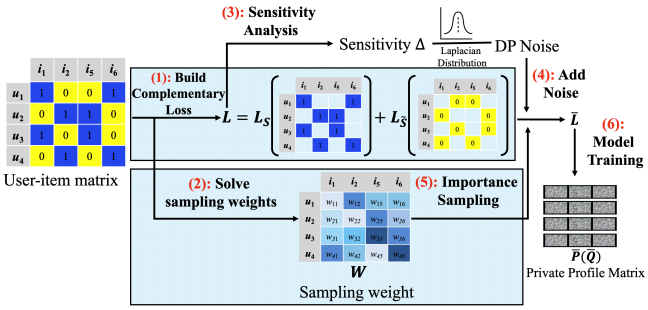


Figure 4: Overview of DPIMF.

4.2 Complementary Loss

In this section, our design of complementary loss is presented, which improves the utility of the private IMF by constraining the sensitivity.

Our first design on complementary loss is built upon the following observation. The sensitivity analysis for the loss function Eq. (9) only considers the term defined over the set of positive users, i.e., $\boldsymbol{U}_i$. This is because that Eq. (9) is built upon the original objective function Eq. (1) that is defined over the positive samples S and the overall samples Ω. Consequently, the effect of the change in implicit rating data only occurs on positive samples in S. We find such an effect can be offset by a term defined over the non-positive samples, i.e., $\tilde{S} = \Omega \backslash S$, thus leading to a decrease in sensitivity.

Without loss of generality, for a given dataset D and function f, let's consider a cumulative query as $F(D) = \sum_{d\in D} f(d)$. Both $F(S)$ and $F(\tilde{S})$ will respond to a change in the dataset S (or $\tilde{S}$), and their responses are always opposite to each other. For example, a record k removed from S will appear in $\tilde{S}$ due to the two sets being mutually exclusive and exhaustive (i.e., $S \cap \tilde{S} = \emptyset, S \cup \tilde{S} = \Omega$). Corresponding to the change, the increment of $F(S)$ is $\Delta F(S) = F(S^{-k}) - F(S^k) = -f(k)$, while the increment of $F(\tilde{S})$ is $\Delta F(\tilde{S}) =$

$F(\tilde{S}^k) - F(\tilde{S}^{-k}) = f(k)$, where D^k and D^{-k} denote the dataset D with and without record k respectively. It is noteworthy that $\Delta F(S)$ and $\Delta F(\tilde{S})$ complement each other, and their sum equals zero. Thus, a decreased sensitivity can be expected by combining a term defined over $\tilde{S}$ to offset the overall impact when implicit rating data changes.

Based on the observation above, we redesign the loss function of MF in Eq. (1) as

$$L(\boldsymbol{P},\boldsymbol{Q}) = \sum_{(u,i)\in S} (\boldsymbol{p}_u^T\boldsymbol{q}_i - 1)^2 + \alpha_0 \sum_{(u,i)\in \tilde{S}} (\boldsymbol{p}_u^T\boldsymbol{q}_i)^2 + R(\boldsymbol{P},\boldsymbol{Q}) \quad (12)$$

Compared with Eq. (1), the second term in $L(\boldsymbol{P},\boldsymbol{Q})$ is modified to keep the predictions close to zero only for the non-positive entries of $\tilde{S}$, rather than all entries of Ω. Moreover, the regularizer $R(\boldsymbol{P},\boldsymbol{Q})$ needs to be modified in a similar way:

$$R(\boldsymbol{P},\boldsymbol{Q}) = \lambda \left(\sum_{u\in\boldsymbol{U}} \left(|\boldsymbol{I}_u| + \alpha_0 |\tilde{\boldsymbol{I}}_u| \right) \|\boldsymbol{p}_u\|_2^2 + \sum_{i\in I} \left(|\boldsymbol{U}_i| + \alpha_0 |\tilde{\boldsymbol{U}}_i| \right) \|\boldsymbol{q}_i\|_2^2 \right), \quad (13)$$

where $\tilde{\boldsymbol{I}}_u := \{i : (u,i) \in \tilde{S}\}, \tilde{\boldsymbol{U}}_i := \{u : (u,i) \in \tilde{S}\}$. Armed with the new formulation, the loss function $L(\boldsymbol{q}_i)$ for item i becomes

$$L(\boldsymbol{q}_i) = \underbrace{\sum_{u\in\boldsymbol{U}_i} \left(\boldsymbol{p}_u^T\boldsymbol{q}_i - 1 \right)^2 + \alpha_0 \sum_{u\in\tilde{\boldsymbol{U}}_i} \left(\boldsymbol{p}_u^T\boldsymbol{q}_i \right)^2}_{L_I(\boldsymbol{q}_i)} + \underbrace{\lambda \left(|\boldsymbol{U}_i| + \alpha_0 |\tilde{\boldsymbol{U}}_i| \right) \|\boldsymbol{q}_i\|_2^2}_{R(\boldsymbol{q}_i)}. \quad (14)$$

We will prove in Sec. 5.1 that the effect of this change in the loss function is always bounded and can be made arbitrarily small under proper parameter settings.

For the ease of analysis, we divide $L(\boldsymbol{q}_i)$ into two components: $L_I(\boldsymbol{q}_i)$ and $R(\boldsymbol{q}_i)$. The first component $L_I(\boldsymbol{q}_i)$ consists of two quadratic terms defined over the positive and non-positive rating data associated with user sets $\boldsymbol{U}_i$ and $\tilde{\boldsymbol{U}}_i$, respectively. The second component $R(\boldsymbol{q}_i)$ is the regularization term, where its weight is also defined over $\boldsymbol{U}_i$ and $\tilde{\boldsymbol{U}}_i$. Omitting the constants, the expansion of $L_I(\boldsymbol{q}_i)$ is thus

$$l_I(\boldsymbol{q}_i) = \boldsymbol{q}_i^T \Big(\sum_{u\in\boldsymbol{U}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \alpha_0 \sum_{u\in\tilde{\boldsymbol{U}}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u \Big) \boldsymbol{q}_i - 2\boldsymbol{q}_i^T \sum_{u\in\boldsymbol{U}_i} \boldsymbol{p}_u. \quad (15)$$

Different from Eq. (10), we treat the polynomial terms with different orders as independent queries. This will allow us to calibrate the noise for different terms based on their own sensitivities. Same as in Sec. 3, we use $\boldsymbol{U}_i'$ to denote the neighboring data obtained by adding a user v to $\boldsymbol{U}_i$. For the first-order term of Eq. (15), same as in Eq. (10), its coefficients satisfy

$$\left\| 2 \sum_{j=1}^{d} \left(\sum_{u\in\boldsymbol{U}_i} \boldsymbol{p}_u - \sum_{u\in\boldsymbol{U}_i'} \boldsymbol{p}_u \right)_j \right\|_1 \le \max_{u\in\boldsymbol{U}} (2\|\boldsymbol{p}_u\|_1).$$

Thus, we define the sensitivity of the first-order term as

$$\Delta_1 = \max_{u\in\boldsymbol{U}} (2\|\boldsymbol{p}_u\|_1). \quad (16)$$

For the second-order term of Eq. (15), the following bound is hold

$$\begin{aligned} &\left\| \sum_{1\le j,l\le d} \left(\sum_{u\in\boldsymbol{U}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u - \sum_{u\in\boldsymbol{U}_i'} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \alpha_0 \sum_{u\in\tilde{\boldsymbol{U}}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u - \alpha_0 \sum_{u\in\tilde{\boldsymbol{U}}_i'} \boldsymbol{p}_u \otimes \boldsymbol{p}_u \right)_{jl} \right\|_1 \\ &\le \sum_{1\le j,l\le d} (1-\alpha_0) |(\boldsymbol{p}_v \otimes \boldsymbol{p}_v)_{jl}| \\ &\le \max_{u\in\boldsymbol{U}} \left((1-\alpha_0) \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{full}} \right). \end{aligned}$$

Therefore, the sensitivity of the second-order term is

$$\Delta_2 = \max_{u\in\boldsymbol{U}} \left((1-\alpha_0) \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{full}} \right). \quad (17)$$

For the regularization term $R(\boldsymbol{q}_i)$ of Eq. (14), we have

$$\left\| |\boldsymbol{U}_i| - |\boldsymbol{U}_i'| + \alpha_0 |\tilde{\boldsymbol{U}}_i| - \alpha_0 |\tilde{\boldsymbol{U}}_i'| \right\|_1 \le 1 - \alpha_0$$

The sensitivity of $R(\boldsymbol{q}_i)$ is in turn defined as

$$\Delta_3 = 1 - \alpha_0. \quad (18)$$

Compared to the counterparts in Eq. (10), the sensitivities of the second-order term and the regularization term are both reduced by a factor of $(1-\alpha_0)$. This agrees with our intuition mentioned earlier that the effect of the removing/adding one rating value in $\boldsymbol{U}_i$ can be offset by introducing the loss defined over $\tilde{\boldsymbol{U}}_i$. By controlling the parameter α_0, the sensitivity can be significantly reduced. For the trade-off between the sensitivity scale and model utility, the setting of α_0 will be theoretically analyzed in Sec. 5.

Our second design is built upon the following observation. In Eq. (17), the coefficients of the polynomial Eq. (15) are defined over complementary sets $\boldsymbol{U}_i$ and $\tilde{\boldsymbol{U}}_i$, implying the fact that they are independent of each other. Thus, the matrix $(\boldsymbol{p}_u \otimes \boldsymbol{p}_u)$ is symmetric. From this observation, we can get a lower bound for the second-order term as

$$\begin{aligned} &\left\| \sum_{j\le l} \left(\sum_{u\in\boldsymbol{U}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u - \sum_{u\in\boldsymbol{U}_i'} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \alpha_0 \sum_{u\in\tilde{\boldsymbol{U}}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u - \alpha_0 \sum_{u\in\tilde{\boldsymbol{U}}_i'} \boldsymbol{p}_u \otimes \boldsymbol{p}_u \right)_{jl} \right\|_1 \\ &\le \sum_{j\le l} (1-\alpha_0) |(\boldsymbol{p}_v \otimes \boldsymbol{p}_v)_{jl}| \\ &\le \max_{u\in\boldsymbol{U}} \left((1-\alpha_0) \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{triu}} \right). \end{aligned}$$

Thus, a tighter bound of the sensitivity Δ_2 can be achieved as

$$\Delta_2 = \max_{u\in\boldsymbol{U}} \left((1-\alpha_0) \|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{triu}} \right), \quad (19)$$

where $\|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{triu}} = \sum_{j\le l} |(\boldsymbol{p}_u \otimes \boldsymbol{p}_u)_{jl}|$ is computed on the values in the upper triangular elements of the matrix. Further, the symmetric noise matrix associated with the sensitivity Δ_2 is

$$\tilde{\boldsymbol{B}} = \text{triu}(\boldsymbol{B}) + \text{tril}_{-1}(\boldsymbol{B}^T), \quad (20)$$

where $\boldsymbol{B}$ is a noise matrix of i.i.d. Laplacian entries, the function $\text{triu}(\cdot)$ outputs a copy of the input matrix with elements below the main diagonal zeroed, and $\text{tril}_{-1}(\cdot)$ outputs a copy of the input with elements above and the main diagonal zeroed. We will prove that the sensitivity and the symmetric noise matrix complies with DP in Sec. 5.

Corresponding to the sensitivities Δ_1, Δ_2 and Δ_3, the privacy budget ε is divided into three parts as ε_1, ε_2, ε_3, where $\varepsilon_i = \varepsilon\beta_i$ with $\beta_i \in [0,1]$ and $\sum_{i=1}^{3}\beta_i = 1$. And the Laplacian noises are sampled and added to $l_I(\boldsymbol{q}_i)$ and $R(\boldsymbol{q}_i)$ as

$$\begin{aligned}\bar{l}_I(\boldsymbol{q}_i) &= \boldsymbol{q}_i^T\Big(\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \alpha_0\sum_{u\in\tilde{\boldsymbol{U}}_i}\boldsymbol{p}_u\otimes\boldsymbol{p}_u + \tilde{\boldsymbol{B}}\Big)\boldsymbol{q}_i \\ &\quad - \boldsymbol{q}_i^T\Big(2\sum_{u\in\boldsymbol{U}_i}\boldsymbol{p}_u + \boldsymbol{b}\Big), \\ \bar{R}(\boldsymbol{q}_i) &= \lambda\big(|\boldsymbol{U}_i| + \alpha_0|\tilde{\boldsymbol{U}}_i| + \eta\big)\,\|\boldsymbol{q}_i\|_2^2 . \end{aligned} \tag{21}$$

Here, $\boldsymbol{b} \sim Lap\left(\frac{\Delta_1}{\varepsilon_1}\right)^d$, $\eta \sim Lap\left(\frac{\Delta_3}{\varepsilon_3}\right)$, and $\tilde{\boldsymbol{B}}$ is generated from $\boldsymbol{B} \sim Lap\left(\frac{\Delta_2}{\varepsilon_2}\right)^{d\times d}$. In particular, when $\alpha_0 = 1$, $\Delta_2 = \Delta_3 = 0$ and no noise needs to be added to the second-order term in $l_I(\boldsymbol{q}_i)$ and $R(\boldsymbol{q}_i)$. In this case, the whole privacy budget ε is used to generate the noise vector $\boldsymbol{b}$. The effect of this setting will be analyzed in Sec. 5. Combining the above two components, we get the private objective function

$$\bar{l}(\boldsymbol{q}_i) = \bar{l}_I(\boldsymbol{q}_i) + \bar{R}(\boldsymbol{q}_i). \tag{22}$$

Same as in Sec. 3, we minimize $\bar{l}(\boldsymbol{q}_i)$ over the convex set $C = \{\boldsymbol{q}_i \mid \|\boldsymbol{q}_i\|_2 \le \sqrt{1/\lambda}\}$ to solve the private profile $\bar{\boldsymbol{q}}_i$. Repeating the above steps through all item in $\boldsymbol{I}$, the private matrix $\bar{\boldsymbol{Q}}$ is obtained.

4.3 The Importance Sampling

In the implicit feedback setting, the number of private objects increases due to the inclusion of unobserved behaviors, resulting in a large DP noise scale. This section introduces an importance sampling method, which allows DPIMF to be trained on only the sampled subset of private objects without sacrificing too much utility. We first present our theoretical contribution on the privacy amplification via importance sampling. Then, we describe how to construct sampling distributions that achieve a given DP guarantee with minimal sample size.

Importance Sampler. We begin by introducing the sampling strategy we use. It is a weighted version of the Poisson sampling [35], which we refer to as *Poisson importance sampling*.

Definition 4 (Poisson Importance Sampling). *Let $q\colon \boldsymbol{U}\times\boldsymbol{I} \to [0,1]$ be a function, and $\Omega = \boldsymbol{U}\times\boldsymbol{I}$ be the dataset of user-item pairs. A Poisson importance sampler for q is a randomized mechanism $S_q(\Omega) = \{(w_{ui}, u, i) \mid \gamma_{ui} = 1\}$, where $w_{ui} = 1/q(u,i)$ are weights and γ_{ui} are independent Bernoulli variables with parameters $p_{ui} = q(u,i)$.*

Here, the importance sampler S_q returns a weighted user-item pair. The varying sampling weights lead to different privacy losses for each data. To capture this privacy heterogeneity, we define the function $\phi : [1,\infty)\times\Omega \to \mathbb{R}_{\ge 0}$ to represent the privacy loss profile of each user-item pair. However, an overall privacy loss is required since our algorithm works in compliance with the rigorous ε-differential privacy. This total privacy loss can be the upper bound of ϕ across all user-item pairs. Based this notion in place, we can state the first main result.

Theorem 3 (Privacy Amplification by Importance Sampling). *Let $\mathcal{M} : [1,\infty)\times\Omega \to \mathcal{Y}$ be an ε-DP mechanism that operates on weighted data sets, $q\colon \Omega \to [0,1]$ be a function, and $S_q(\cdot)$ be a Poisson importance sampler for q. The mechanism $\widehat{\mathcal{M}} = \mathcal{M}\circ S_q$ satisfies ε^*-DP where ε^* satisfies*

$$\log\left(1 + q(u,i)\left(e^{\phi(w,u,i)} - 1\right)\right) \le \varepsilon^*, \quad \forall u, i, \tag{23}$$

and $w = 1/q(u,i)$.

Sampling with optimal privacy. We now describe how to construct a sampling distribution that achieves a given privacy guarantee with minimal sample size. The motivation for this is two-fold. First, by imposing a overall privacy loss as a constraint, we can ensure that the importance sampling mechanism satisfies ε-DP by design. Second, the sample size of private objects is a primary indicator of the noise scale and the efficiency of mechanism. Minimizing the expected sample size subject to a given ε^*-DP constraint can be described as the following optimization problem.

Problem 1 (Privacy-optimal sampling). *For a privacy loss profile $\phi\colon [1,\infty)\times\Omega \to \mathbb{R}_{\ge 0}$, a target privacy guarantee ε^*, and a data set of user-item pairs Ω, we define the privacy-optimal sampling problem as*

$$\underset{\boldsymbol{W}\in\mathbb{R}^{|\boldsymbol{U}|\times|\boldsymbol{I}|}}{\arg\min} \quad \sum_{(u,i)\in\Omega}\frac{1}{w_{ui}} \tag{24a}$$

$$\text{s.t.} \quad \log\left(1 + \frac{1}{w_{ui}}\left(e^{\phi(w_{ui},u,i)} - 1\right)\right) \le \varepsilon^*, \quad \forall u, i, \tag{24b}$$

$$w_{ui} \ge 1, \qquad \forall u, i. \tag{24c}$$

The constraint in Eq. (24b) captures the requirement that ϕ should be bounded by ε^* for all $(u,i) \in \Omega$, and the constraint in Eq. (24c) ensures that $1/w_{ui}$ is a probability. This problem is guaranteed to have a unique solution if ϕ is convex. This problem can be solved by the off-the-shelf methods for convex problem. We defer the proof and the algorithm for solving this problem to the full version [].

4.4 DPIMF Implementation

Based on the above two strategies, we shows the implementation details of DPIMF. We first provides the process for solving the private item profile matrix, as shown in Algorithm 1. Then we show how to extend the algorithm to solve the private user profile matrix.

It starts by initializing the profile matrix and sensitivities (Lines 1-2). Here, we uniformly sample random values between 0 and 1 by following the approach used in existing literature [2]. To reduce the computational overhead, we pre-calculate $\boldsymbol{G}_U$, which will be reused in the iterative process (Line 3). Next, the privacy profile is initialized to compute the importance weight by solving an optimization problem that is detailed in Section 4.3 (Line 5-6). As described in Section 3, we compute the coefficients of the loss for adding noise. And the data points are weighted according to $\boldsymbol{W}$

Algorithm 1: Full-fledged DPIMF for Private Item Profile Matrix

Input : The local user profile matrix $\boldsymbol{P}$, the user set $\boldsymbol{U}$, the item set $\boldsymbol{I}$, the set of users interacted with each item $\{\boldsymbol{U}_i \mid i \in \boldsymbol{I}\}$, α_0, the privacy budgets $\varepsilon_1, \varepsilon_2, \varepsilon_3$.

Output : $\bar{\boldsymbol{Q}}$

1 Randomly initialize $\bar{\boldsymbol{Q}}$ such that each element q_{ij} is randomly drawn from $[0, 1]$.

2 Compute sensitivities: $\Delta_1, \Delta_2, \Delta_3$.

3 Pre-compute: $\boldsymbol{G}_{\boldsymbol{U}}=\sum_{u\in\boldsymbol{U}} \boldsymbol{p}_u \otimes \boldsymbol{p}_u$.

4 \\ **Solve the weights for importance sampling**.

5 Initialize privacy profile ϕ

6 Solve the optimal $\boldsymbol{W}$ by minimizing Eq. (24).

7 **for** *each* $i \in \boldsymbol{I}$ **do**

8 \\ **Build the private complementary loss with** $\boldsymbol{W}$.

9 Compute the coefficients of complementary loss:
$\boldsymbol{G}_{\boldsymbol{U}_i} = \sum_{u\in\boldsymbol{U}_i} w_{ui}(\boldsymbol{p}_u \otimes \boldsymbol{p}_u)$,
$\boldsymbol{A}_1 = 2\sum_{u\in\boldsymbol{U}_i} w_{ui}\boldsymbol{p}_u$,
$\boldsymbol{A}_2 = \boldsymbol{G}_{\boldsymbol{U}_i} + \alpha_0(\boldsymbol{G}_{\boldsymbol{U}} - \boldsymbol{G}_{\boldsymbol{U}_i}) + \lambda(|\boldsymbol{U}_i| + \alpha_0|\tilde{\boldsymbol{U}}_i|)\boldsymbol{E}$.

10 Sample $\boldsymbol{b} \sim Lap\left(\frac{\Delta_1}{\varepsilon_1}\right)^d, \boldsymbol{B} \sim Lap\left(\frac{\Delta_2}{\varepsilon_2}\right)^{d\times d}, \eta \sim Lap\left(\frac{\Delta_3}{\varepsilon_3}\right)$

11 Compute symmetric noise: $\tilde{\boldsymbol{B}} = \mathrm{triu}(\boldsymbol{B}) + \mathrm{tril}_{-1}(\boldsymbol{B}^T)$.

12 Derive the private loss:
$\bar{L}(\boldsymbol{q}_i) = \boldsymbol{q}_i^T(\boldsymbol{A}_2 + \lambda\eta\boldsymbol{E} + \boldsymbol{B})\boldsymbol{q}_i - \boldsymbol{q}_i^T(\boldsymbol{A}_1 + \boldsymbol{b})$.

13 Solve $\bar{\boldsymbol{q}}_i = \underset{\boldsymbol{q}_i:\|\boldsymbol{q}_i\|_2\le\sqrt{1/\lambda}}{\operatorname{argmin}} \bar{L}(\boldsymbol{q}_i)$.

14 **return** $\bar{\boldsymbol{Q}}$.

(Lines 9). Then, Laplacian noises are sampled and added to the coefficients to build the final private loss (Lines 10-12). The optimal private profile vector is solved within a convex set (Line 13).

Alogrithm 1 is for solving item profile matrix. Now we discuss how to extend it to user profile matrix. When solving a user's profile vector $\boldsymbol{p}_u$, we focus on the items that u has interacted with, denoted as $\boldsymbol{I}_u$. In this case, the input $\boldsymbol{P}$ and $\{\boldsymbol{U}_i \mid i \in \boldsymbol{I}\}$ should be replaced by $\boldsymbol{Q}$ and $\{\boldsymbol{I}_u \mid u \in \boldsymbol{U}\}$, respectively. The importance sampling steps (Lines 5-6) remains unchanged since $\boldsymbol{W}$ is solved by an independent problem. To build the private loss of $\boldsymbol{p}_u$, just change all $\boldsymbol{U}_i$ and $\boldsymbol{p}_u$ in the algorithm (Lines 9-12) to $\boldsymbol{I}_u$ and $\boldsymbol{q}_i$, respectively. Then we can derive $\bar{\boldsymbol{p}}_u$ using the same solution method as for $\bar{\boldsymbol{q}}_i$ (Line 13).

5 THEORETIC ANALYSES

In this section, we establish utility and privacy guarantees of our proposed DPIMF methods.

5.1 Utility Analysis

The utility guarantee of the proposed DPIMF methods are theoretically analyzed in this section. To bound the sensitivity and improve utility, we modify the loss function of Eq. (8) to the form in Eq. (14). The effect of such modification is bounded, and this fact is shown by the theorem below.

Theorem 4. *Let* $N = \sqrt{1/\lambda}, c = \max_{u\in\boldsymbol{U}, j\in\{1,2,\ldots,d\}} |p_{uj}|$. *For the the redesigned loss function* $L(\boldsymbol{q}_i)$ *of Eq. (14), and the original loss function* $L^O(\boldsymbol{q}_i)$ *of Eq. (8), say* $\boldsymbol{q}_i^* = \arg\min L(\boldsymbol{q}_i)$, $\boldsymbol{q}_i' = \arg\min L^O(\boldsymbol{q}_i)$, *we have:*

$$L(\boldsymbol{q}_i^*) - L^O(\boldsymbol{q}_i') \le \alpha_0|\boldsymbol{U}_i|N^2(\lambda + 2dc^2). \tag{25}$$

Proof: See the Appendix. ∎

Theorem 4 implies that the minimizer $\boldsymbol{q}_i^*$ of the redesigned loss function is always bounded to the minimizer of the original loss, and they can be arbitrarily close to each other under proper setting of parameters α_0, λ and d. The empirical results in Sec. 6 further corroborate that the impact of the modification on utility is negligible compared to the impact of the perturbation introduced by DP.

To inspect the effectiveness of each technique proposed in Sec. 4, we denote $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ and $\mathcal{M}_4$ as four strategies we proposed to inspect each method's contribution for utility improvement. In particular, the strategy $\mathcal{M}_1$ represents the scheme proposed in Sec. 4 where the private loss is defined as Eq. (11). The strategies $\mathcal{M}_2, \mathcal{M}_3$ and $\mathcal{M}_4$ represent the utility-enhanced strategies proposed in Sec. 4. The difference among them lies only in the perturbation techniques. In $\mathcal{M}_2$, Δ_2 is computed as Eq. (17), and a noise matrix of i.i.d. entries is added to the second-order term in the loss. In $\mathcal{M}_3$, Δ_2 is computed as Eq. (19), and a symmetric noise matrix is sampled for perturbation. The method $\mathcal{M}_4$ represents our assumed optimal algorithm. It implements $\mathcal{M}_3$ with the presumed optimal settings, specifically setting α_0 to 1 and utilizing importance sampling. For these four different strategies, we have the following theorem.

Theorem 5. *Let* $N = \sqrt{1/\lambda}$, $\varepsilon_1 = \beta_1\varepsilon, \varepsilon_2 = \beta_2\varepsilon, c = \max_{u,j}|p_{uj}|$ *and* $\mathcal{S}_q$ *be the importance sampler solved in Eq. (24). Given the non-private loss* $L(\boldsymbol{q}_i)$ *of Eq. (14) and its minimizer* $\boldsymbol{q}_i^*$. *Let* $\bar{\boldsymbol{q}}_i^{(1)}, \bar{\boldsymbol{q}}_i^{(2)}, \bar{\boldsymbol{q}}_i^{(3)}$, *be the minimizer of the private loss* $\bar{l}(\boldsymbol{q}_i)$ *of Eq. (??) with strategies* $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$, *respectively. Let* $\bar{l}_{\mathcal{S}_q}(\boldsymbol{q}_i)$ *be the loss with importance sampling, and* $\bar{\boldsymbol{q}}_i^{(4)}$ *be its minimizer. Then, the utility of the profile vectors satisfies*

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(1)}) - L(\boldsymbol{q}_i^*)] \le \frac{\sqrt{2}cdN\left[(N + \frac{2\sqrt{d}}{d})(d+1)^2\right]}{\varepsilon}, \tag{26}$$

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(2)}) - L(\boldsymbol{q}_i^*)] \le \frac{\sqrt{2}cdN\left[\frac{1}{\beta_2}Nd^2(1-\alpha_0) + \frac{4}{\beta_1}\sqrt{d}\right]}{\varepsilon}, \tag{27}$$

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(3)}) - L(\boldsymbol{q}_i^*)] \le \frac{\sqrt{2}cdN\left[\frac{1}{2\beta_2}Nd(1+d)(1-\alpha_0) + \frac{4}{\beta_1}\sqrt{d}\right]}{\varepsilon}, \tag{28}$$

$$\mathbb{E}[L(\bar{\boldsymbol{q}}_i^{(4)}) - L(\boldsymbol{q}_i^*)] \le \frac{4\sqrt{2d}cdN}{\varepsilon}. \tag{29}$$

Proof: See the Appendix. ∎

The above theorem bounds the empirical risk of the four strategies for the proposed differentially private IMF. A higher bound indicates a larger loss in the utility of the strategy. Denote the above error bounds for $\bar{\boldsymbol{q}}_i^{(1)}, \bar{\boldsymbol{q}}_i^{(2)}, \bar{\boldsymbol{q}}_i^{(3)}$ and $\bar{\boldsymbol{q}}_i^{(4)}$ as $\gamma_1, \gamma_2, \gamma_3, \gamma_4$. The relationship among the these bounds are concluded in the following theorem as a corollary of Theorem 5.

Corollary 1. *Given* $\beta_1, \beta_2 \in [0, 1]$, $\gamma_1 > \gamma_2$ *when* $\alpha_0 \ge 1 - \beta_2$ *and* $d \ge \sqrt{1/\beta_1} - 1$. *For all* $\alpha_0, \beta_1, \beta_2 \in [0, 1]$ *and* $d \in \mathbb{N}^+$, *we have* $\gamma_2 \ge \gamma_3 \ge \gamma_4$ *and* $\gamma_1 > \gamma_4$.

Proof: See the Appendix. ∎

The Corollary 1 proves that $\mathcal{M}_2$ enjoys better utility than $\mathcal{M}_1$ with a proper setting of α_0. It demonstrates the effectiveness of our modification to the loss function for enhancing utility. The condition α_0 needs to meet to guarantee the improvement can be understood intuitively. For example, a large α_0 is required to limit the noise when a small proportion of privacy budget is allocated to the second-order term. Moreover, the improvement from $\mathcal{M}_2$ to $\mathcal{M}_3$ is attributed to the technique of symmetric noise matrix. Moreover, Corollary 1 presents the optimal utility guarantee achieved is when α_0 is set to 1. The vanished sensitivities prevent the second-order term and regularization term from perturbed, leading to a great improvement in utility. Moreover, unlike uniform sampling, importance sampling gives an unbiased estimate of the loss, thus ensuring that the improvements still hold after sampling. The above theoretical findings will be empirically validated by the experimental results presented in Sec. 6.

5.2 Privacy Analysis

Theorem 6. *DPIMF satisfies ε-DP.*

Proof: We prove that, solving $\bar{\boldsymbol{q}}_i$ of the private loss $\bar{l}(\boldsymbol{q}_i)$ of Eq. (22) for an arbitrary item i satisfies ε-DP. Let $\boldsymbol{T} \in \mathbb{R}^{d\times d}, \boldsymbol{t} \in \mathbb{R}^d$, and $\tau \in \mathbb{R}$ be the perturbed coefficients of the second and first-order terms in $\bar{l}_I(\boldsymbol{q}_i)$, and the perturbed weight of the regularization term $\bar{R}(\boldsymbol{q}_i)$ of Eq. (21), respectively. Define dataset D_i that records both the positive and non-positive users for i, then its neighboring dataset D_i' is obtained from D_i by moving one user from the positive set to the non-positive set (and vice versa). For a privacy budget ε divided into $\varepsilon_1, \varepsilon_2, \varepsilon_3$, we have

$$
\begin{aligned}
&\frac{\Pr\{\bar{l}(\boldsymbol{q}_i) \mid D_i\}}{\Pr\{\bar{l}(\boldsymbol{q}_i) \mid D_i'\}} \\
&= \frac{\Pr\{\sum_{u\in U_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \alpha_0 \sum_{u\in \tilde{U}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \tilde{\boldsymbol{B}} = \boldsymbol{T}\}}{\Pr\{\sum_{u\in U_i'} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \alpha_0 \sum_{u\in \tilde{U}_i'} \boldsymbol{p}_u \otimes \boldsymbol{p}_u + \tilde{\boldsymbol{B}}' = \boldsymbol{T}\}} \\
&\cdot \frac{\Pr\{2\sum_{u\in U_i} \boldsymbol{p}_u + \boldsymbol{b} = \boldsymbol{t}\} \cdot \Pr\{|U_i| + \alpha_0|\tilde{U}_i| + \eta = \tau\}}{\Pr\{2\sum_{u\in U_i'} \boldsymbol{p}_u + \boldsymbol{b}' = \boldsymbol{t}\} \cdot \Pr\{|U_i'| + \alpha_0|\tilde{U}_i'| + \eta' = \tau\}} \\
&= \frac{\prod_{j\le l} \Pr\left\{\tilde{B}_{jl} = T_{jl} - (\boldsymbol{G}_{U_i} + \alpha_0 \boldsymbol{G}_{\tilde{U}_i})_{jl}\right\}}{\prod_{j\le l} \Pr\left\{\begin{array}{l}\tilde{B}'_{jl} = T_{jl} - (\boldsymbol{G}_{U_i} + \alpha_0 \boldsymbol{G}_{\tilde{U}_i})_{jl} - \\ (1-\alpha_0)(\boldsymbol{p}_v \otimes \boldsymbol{p}_v)_{jl}\end{array}\right\}} \\
&\cdot \frac{\prod_{j=1}^{d} \Pr\left\{b_j = t_j - (2\sum_{u\in U_i'} \boldsymbol{p}_u)_j\right\}}{\prod_{j=1}^{d} \Pr\left\{b_j = t_j - (2\sum_{u\in U_i'} \boldsymbol{p}_u)_j - 2p_{vj}\right\}} \\
&\cdot \frac{\Pr\left\{\eta = \tau - |U_i| - \alpha_0|\tilde{U}_i|\right\}}{\Pr\left\{\eta = \tau - |U_i| - \alpha_0|\tilde{U}_i| - (1-\alpha_0)\right\}} \\
&\le \exp\left(\frac{\varepsilon_2 \max_{u\in U}((1-\alpha_0)\|\boldsymbol{p}_u \otimes \boldsymbol{p}_u\|_{\text{triu}})}{\Delta_2}\right) \\
&\quad \cdot \exp\left(\frac{\varepsilon_1 \max_{u\in U}(2\|\boldsymbol{p}_u\|_1)}{\Delta_1}\right) \exp\left(\frac{(1-\alpha_0)\varepsilon_3}{\Delta_3}\right) \\
&= \exp(\varepsilon_1 + \varepsilon_2 + \varepsilon_3) = \exp(\varepsilon),
\end{aligned}
$$

where $\boldsymbol{G}_{U_i} = \sum_{u\in U_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u$, $\boldsymbol{G}_{\tilde{U}_i} = \sum_{u\in \tilde{U}_i} \boldsymbol{p}_u \otimes \boldsymbol{p}_u$. Thus solving $\bar{\boldsymbol{q}}_i$ satisfies ε-DP. Since the ratings of each item are disjoint, according to Theorem 1 (i.e., parallel composition of DP), we conclude that DPIMF satisfies ε-DP. ∎

6 EXPERIMENTAL EVALUATION

In this section, we first introduce our experiment setup in Section 6.1, and then present the experimental results and our analysis in Section 6.2.

6.1 Experiment Setup

6.1.1 Datasets. In the experiment, three datasets, i.e., 10M MovieLens , YahooMusic and Amazon are used to evaluate the performance of our proposed method. **10M MovieLens** (denoted as ML-10M in the sequel) were collected by GroupLens through their experimental movie recommendation system and all ratings assigned by the users to movies are in a 1-to-5 stars scale. **YahooMusic** consists of ratings supplied by users during normal interaction with YahooMusic services. **Amazon** records the ratings provided by users for products in Amazon Instant Video (an online shopping website). The details about these datasets are shown in Table 2.

According to our problem setting in Section 2.3, the experiments will be conducted in two scenarios. (1) Different parties share the same items but have different user sets, requiring them to exchange item profile matrices. (2) Different parties share the same users but have different item sets, in which the user profile matrices will be exchanged. For the former scenario, we split the datasets into s disjoint sub-datasets by users, such that the ratings of each user can appear in only one dataset. For the latter, we split the datasets by items in the same way. Each of the sub-dataset simulates the local dataset for a party. In this section, we mainly present the results of the first scenario in the interest of space, while the results for the latter scenario are provided in the full version [].

Table 2: The datasets used in our experiments.

Property	ML-10M	YahooMusic	Amazon
Users	71567	8089	5130
Items	10681	1000	1685
Density	4.3%	3.4%	0.7%
Avg. #ratings per user	97	33	12
Avg. #ratings per item	40.5	270	36

6.1.2 Evaluation Indicator. We adopt the widely used leave-one-out evaluation, where the latest interaction of each user is held out for prediction. We train all models on the remaining data and generate ranked recommendation list. For recommendation effectiveness, we mainly consider two metrics, namely Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). HR measures whether the test item appears in the target user's recommendation list, and NDCG measures the ranked position of the hit. HR is defined as follows:

$$\text{HR@k} = \frac{\sum_{u\in U} \left|I_p(u)@k \cap I_a(u)\right|}{\sum_{u\in U} |I_a(u)|}, \tag{30}$$

http://www.grouplens.org.
http://research.yahoo.com/AcademicRelations.
https://jmcauley.ucsd.edu/data/amazon/

where $I_p(u)@k$ is the set of top k items in the ranked recommendation list. The symbol $I_a(u)$ represents the set of items the user u has interacted with. NDCG is given by

$$\text{NDCG@k} = \frac{1}{|U|} \sum_{u \in U} \frac{\text{DCG}_\text{u}@\text{k}}{\text{IDCG}_\text{u}@\text{k}}, \tag{31}$$

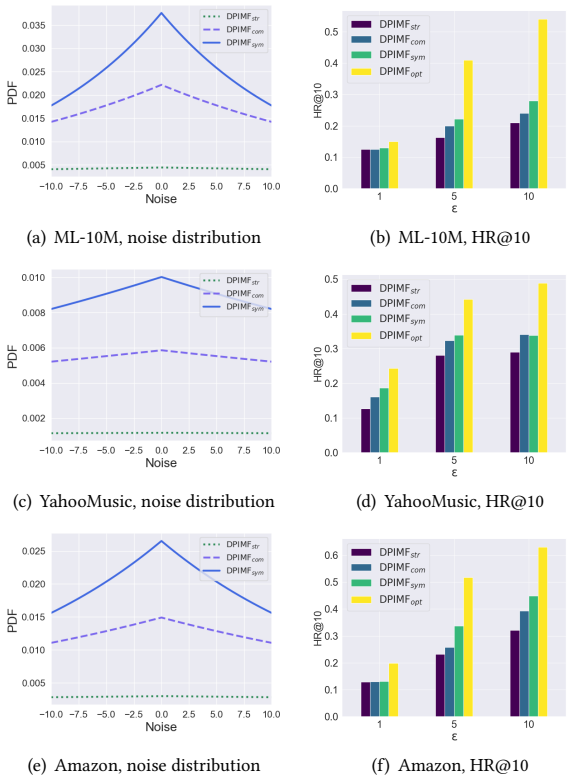
$$\text{DCG}_\text{u}@\text{k} = \sum_{\text{idx}=1}^{k} \frac{2^{rel_{\text{idx}}} - 1}{\log_2(\text{idx}+1)}, \tag{32}$$

where $rel_{\text{idx}} \in \{0, 1\}$ indicates whether there is an interaction between user u and the idx-th item in the ranked list, and $\text{IDCG}_\text{u}@\text{k}$ is the ideal $\text{DCG}_\text{u}@\text{k}$ computed on the recommendation list sorted by rel_{idx} in descending order. The larger values of HR and NDCG, the better recommendation quality.

6.1.3 Competitors. We compare our scheme with the ground truth results (i.e., recommender system without privacy protection) and some state-of-the-art differentially private collaborative filtering techniques for implicit data, listed as follows.

- Ground Truth (GT) [32]: This method uses alternating least square to solve the IMF problem without considering user privacy. It optimizes Eq. (1) by alternately fixing $\boldsymbol{P}$ or $\boldsymbol{Q}$ and solves the other. Since iALS is one of the most commonly used methods to solve IMF problem, we regard this as the ground truth for comparison.
- DPMF [14]: Differentially Private Matrix Factorization for explicit feedback uses an objective perturbation mechanism.
- DPLCF [10]: This method protects users' implicit data by introducing random flipping technique based on nearest neighbors. Then the server publishes a sanitized similarity matrix, and each user generates recommendation results locally using that similarity matrix.
- LDPICF [12]: Similar to DPLCF, this method is also based on nearest neighbors. It uses random flipping method to perturb user data and reconstruct the relationship between item pairs using two frequency estimation techniques.
- $\text{DPIMF}_{\text{str}}$: The strawman DPIMF proposed in Section 3;
- $\text{DPIMF}_{\text{com}}$: The complementary loss-based DPIMF proposed in Section 4 without adding symmetric noise;
- $\text{DPIMF}_{\text{sym}}$: The complementary loss-based method proposed in Section 4 with symmetric noise;
- $\text{DPIMF}_{\text{opt}}$: $\text{DPIMF}_{\text{sym}}$ with optimal setting (i.e., $\alpha_0 = 1$).
- $\text{DPIMF}_{\text{opt}}$-IS: $\text{DPIMF}_{\text{opt}}$ with importance sampling.

6.1.4 Parameter Setting. For the setting of hyper-parameters, the number of sub-datasets s is set to 10. For all MF-based methods, the total and local iterations of model learning are set to 100 and 20, respectively. This means each party queries the local dataset 2000 times during the federated learning. The parameters of MF-based methods (i.e., including DPMF and our DPIMF solutions) are empirically set according to optimal results of hyperparameter searching (We repeat each combination of possible hyperparameter values 10 times across all datasets) of different datasets involved. Specifically, the regularization parameters λ is set to 0.07, 1.0, 1.0 for the ML-10M, YahooMusic and Amazon, respectively. The number of latent factor d is set to 20, 16 and 8 for ML-10M, YahooMusic and Amazon dataset, respectively. For two neighbor-based methods (i.e., DPLCF and LDPICF), we set the number of neighbors to 100 according to the best results of parameter tuning.



(a) ML-10M, noise distribution (b) ML-10M, HR@10

(c) YahooMusic, noise distribution (d) YahooMusic, HR@10

(e) Amazon, noise distribution (f) Amazon, HR@10

Figure 5: Performance of DPIMF methods on noise distribution and HR@10.

6.2 Experimental Results and Analysis

6.2.1 The Effect of Complementary Loss. In this section, we evaluate the effectiveness of the proposed complementary loss in Section 4 by comparing the recommendation performance of $\text{DPIMF}_{\text{str}}$, $\text{DPIMF}_{\text{com}}$, $\text{DPIMF}_{\text{sym}}$ and $\text{DPIMF}_{\text{opt}}$.

We first assess the effectiveness of complementary loss in limiting the magnitude of noises added to the the loss function. We record the distributions of the noises in the second-order coefficients $\tilde{\boldsymbol{B}}$ (or $\boldsymbol{B}$) in $\text{DPIMF}_{\text{str}}$, $\text{DPIMF}_{\text{com}}$, $\text{DPIMF}_{\text{sym}}$, respectively. For $\text{DPIMF}_{\text{com}}$, $\text{DPIMF}_{\text{sym}}$, we set $\alpha_0 = 0.8$, $\varepsilon = 0.1$, $\beta_1 = 0.1$, $\beta_2 = 0.8$ and $\beta_3 = 0.1$. The distributions in ML-10M, YahooMusic and Amazon datasets are reported in Figures 5(a), (c) and (e). It is clearly shown in the figure that the noise distribution of $\text{DPIMF}_{\text{sym}}$ is sharper than other two schemes, while the curve of $\text{DPIMF}_{\text{str}}$ is the flattest. Such shapes convey that the variance of the noise in $\text{DPIMF}_{\text{sym}}$ is the lowest. And the variance of the noise in $\text{DPIMF}_{\text{str}}$ is higher than that in $\text{DPIMF}_{\text{com}}$. The differences of noise distributions indicate that the magnitude of the noise added to $\text{DPIMF}_{\text{sym}}$ is lower than the other two schemes, and in turn it enjoys an improvement in recommendation accuracy.

In order to validate the above claim, we evaluate the HRs of the recommendation lists generated by the three schemes on ML-10M, YahooMusic and Amazon datasets, respectively. For $\text{DPIMF}_{\text{com}}$ and $\text{DPIMF}_{\text{sym}}$, α_0 is set to 0.8. With varying privacy budget ε from 1 to

10, the test results are reported in Figure 5 (b), (d) and (f). Overall, the utility of the recommendations improves as the privacy budget increases. This is consistent with the property of DP.

It is obvious that $\text{DPIMF}_{\text{str}}$ suffers the largest utility loss. When $\varepsilon > 1$, the other three schemes consistently outperform $\text{DPIMF}_{\text{str}}$ in HR across all datasets. For example, on the Amazon dataset, the HR value of $\text{DPIMF}_{\text{str}}$ is 7%, 13%, 31% lower than than of $\text{DPIMF}_{\text{com}}$, $\text{DPIMF}_{\text{sym}}$ and $\text{DPIMF}_{\text{opt}}$, respectively. This validates the claim that the scheme injected noise with largest variance performs worst. The improvements also demonstrate the effectiveness of the offset technique for boosting recommendation accuracy.

The figures also show that $\text{DPIMF}_{\text{sym}}$ is more accurate than $\text{DPIMF}_{\text{com}}$ in most cases across all datasets. For example, in Figures 5 (b), (d) and (f), we observe that within the privacy budget range of $[1, 10]$, the HRs of $\text{DPIMF}_{\text{sym}}$ is on average 2.2% higher than $\text{DPIMF}_{\text{com}}$ on ML-10M. The results validate the theoretical conclusion that $\text{DPIMF}_{\text{sym}}$ reduces unnecessary noises by using the symmetric strategy. Summing up the upper triangular entries rather than the full entries in the coeffecients matrix brings $\text{DPIMF}_{\text{sym}}$ a lower sensitivity, which in turn enhances the utility of recommendations. For the performance on the YahooMusic and Amazon datasets shown in Figures 5 (d) and (f), $\text{DPIMF}_{\text{com}}$ outperforms $\text{DPIMF}_{\text{sym}}$ in some cases (e.g., the HR on Amazon when $\varepsilon = 5$). This is because the noise variances of the two methods are both relatively large on YahooMusic and Amazon, as shown in Figures 5(c) and (e).

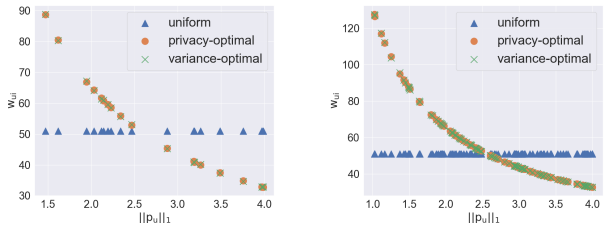
Note that $\text{DPIMF}_{\text{opt}}$ always performs the best, achieving significantly higher HRs under different privacy budgets across all datasets. On Amazon dataset with $\varepsilon = 10$, the HR of $\text{DPIMF}_{\text{opt}}$ reaches 0.64, which is 32%, 25% and 19% higher than that of $\text{DPIMF}_{\text{str}}$, $\text{DPIMF}_{\text{com}}$ and $\text{DPIMF}_{\text{sym}}$, respectively. The results are consistent with the conclusions of utility analysis. By setting α_0 to 1, the effect of the change in data is completely offset. The ramification of this setting is that the second-order coefficients are free from perturbation. It not only reduces the injected noise, but also saves the privacy budget. More privacy budget are utilized to inject less noise into the objective function. As a consequence, the utility of recommendation increases dramatically, which explains the superiority of $\text{DPIMF}_{\text{opt}}$ shown in Figures 5 (b), (d) and (f).

6.2.2 The Effect of Importance Sampling. To study the effect of importance sampling, we generate synthetic data and compare informative importance sampling distributions to uniform sampling in terms of privacy and variance at a fixed expected sample size. To test the effect of noise, we focus on the perturbed term of first-order coefficients in the private loss, i.e., $\mathcal{A}_1(\mathcal{D}) = 2\sum_{u\in U_i} w_{ui}\boldsymbol{p}_u + \boldsymbol{b}$. When the data $\mathcal{D} = \boldsymbol{U}_i$ is sampled by Poisson importance sampler $\mathcal{S}_q$, the mechanism $\mathcal{A}\circ\mathcal{S}_q$ for the j-th dimension has the variance:

$$\text{Var}\left[\mathcal{A}(\mathcal{D})_j\right] = 2 + \sum\nolimits_{u\in U_i}(w_{ui} - 1)\, p_{uj}^2.$$

Let m be the sample size, we compare our important sampling with two sampling strategies, namely uniform sampling where the sampling rate is $m/|\Omega|$, and variance-optimal sampling, defined as the solution of minimizing the variance $\text{Var}\left[\mathcal{A}(\mathcal{D})_j\right]$.

We generate 1000 points from an isotropic multivariate normal distribution in $d = 10$ dimensions with variance $\sigma^2 = 1/d$ in each dimension. We visualize the importance weights w_{ui} for each sampling strategy. For this, we fix a target sample size at $m \in \{20, 100\}$ and compute the weights w_{ui} for each sampling strategy that achieve the target sample size in expectation. The results are shown in Figure 6. Remarkably, the privacy-optimal weights by our importance sampling and the variance-optimal weights are almost identical. The results demonstrates that (i) privacy and accuracy are well-aligned goals in importance sampling and (ii) uniform sampling is highly suboptimal at least in the case of IMF. The overall improvement on model utility will be demonstrated in the following results.



(a) m=20 (b) m=100

Figure 6: Comparison of sampling strategies.

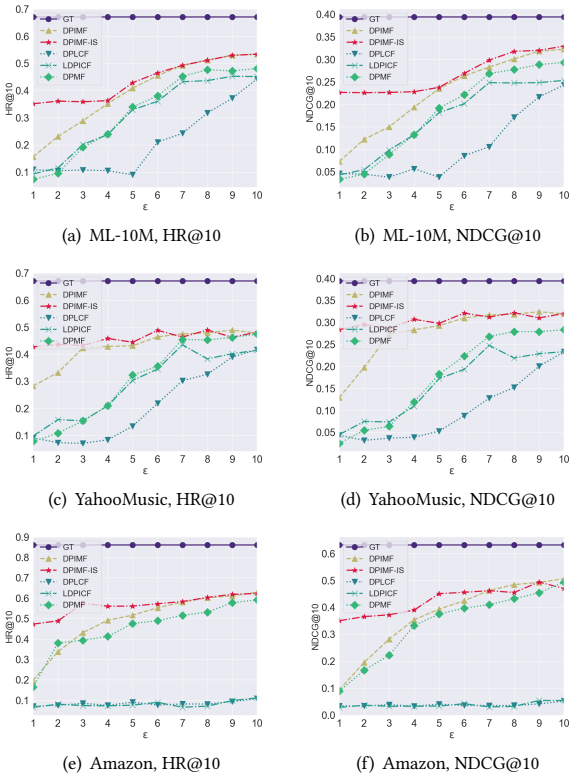
6.2.3 Comparing with other schemes. In order to demonstrate the effectiveness of our methods, we compare the recommendation quality of our DPIMP methods (i.e., $\text{DPIMF}_{\text{opt}}$ and $\text{DPIMF}_{\text{opt}}$-IS, which are referred to as DPIMF and DPIMF-IS in the sequel for simplicity) with the non-private scheme (i.e., GT) and other differentially private methods for the IMF problem. With varying privacy budgets from 1 to 10, the HRs and NDCGs of the competitors on ML-10M, YahooMusic and Amazon datasets are reported in Figure 7.

Overall, the performance curves of all private schemes converge to the ground truth (i.e., GT) with the increasing privacy budget ε. This is consistent with the feature of DP that utility will be improved when weakening privacy insurance. The recommendation accuracy of DPIMF is always better than that of DPLCF and LDPICF on all datasets. Furthermore, DPIMF-IS consistently outperforms the other private methods over all privacy levels. On ML-10M, as shown in Figures 7(a) and (b), the HR of DPIMF-IS is on average 18.3% and 8.3% higher than DPLCF and LDPICF. In NDCG, DPIMF-IS is 12.1% and 5.4% higher than DPLCF and LDPICF. The improvements becomes more significant on YahooMusic as shown in Figures 7(c) and (d). When $\varepsilon = 3$, the HR and NDCG of DPIMF-IS are 26%, 20% higher than DPLCF, and 35%, 22% higher than LDPICF, respectively. Furthermore, this accuracy gap even reaches 50% around both in HR and NDCG on Amazon dataset.

The higher utility of DPIMF methods over DPLCF and LDPICF first validate the better generalization ability of matrix factorization-based methods than KNN-based methods. Second, the DPIMF methods not only benefit the expressiveness of matrix factorization models, but also ensure the model utility with privacy guarantee. By carefully designing the loss function, the sensitivity and the added noise, our DPIMF methods effectively limit the perturbation error introduced into the model. In particular, comparing with DPIMF, DPIMF-IS achieves a more significant utility improvement under stricter privacy requirements (e.g., $\varepsilon < 5$). This aligns with the privacy amplification conclusion proposed in Section 5.1, as the

effect of privacy amplification becomes more pronounced when the overall privacy budget is smaller, leading to reduced noise and better utility.

Moreover, the KNN-based methods suffer from the difficulty in finding proper neighbors in data with high sparsity level. This is demonstrated in Figures 7(e) and (f). The figures tell that the HRs and NDCGs of DPLCF and LDPICF are just around 0.1 on Amazon, the sparsest dataset among the three. The results demonstrate that the DPLCF and LDPICF are far from practical on extremely sparse datasets. On the contrary, the HR and NDCG of DPIMF on the same dataset increase steadily when increasing the privacy budget. When $\varepsilon = 10$, the HR and NDCG of DPIMF exceed 0.6 and 0.5, respectively. The utility loss is reduced to around 0.2. This result verifies the advantage of DPIMF in dealing with highly sparse datasets.



(a) ML-10M, HR@10 (b) ML-10M, NDCG@10

(c) YahooMusic, HR@10 (d) YahooMusic, NDCG@10

(e) Amazon, HR@10 (f) Amazon, NDCG@10

Figure 7: Overall results on HR@10 and NDCG@10.

7 RELATED WORK

Differential Privacy, as a rigorous privacy notion which guarantees the privacy of any user participating in a statistical computation, has been widely applied to recommender systems. The seminal work by McSherry et al. [24] first introduced DP to the domain of collaborative filtering through perturbing the item covariance matrix. The studies in [34, 43] incorporated DP into KNN-based methods, where the perturbation is injected into the similarity computation and the score prediction phases. The schemes in [18, 23] employed the mechanism proposed in [35], which learns the item profile matrix while preserving DP via stochastic gradient Langevin dynamics. Considering the privacy of a user's overall data, Jain et al. [15] adopted a relaxed differential privacy, i.e., joint differential privacy (JDP), and proposed a private Frank-Wolfe algorithm in which the noise is added into the feasible gradient descent direction. Following this work, Chien et al. [4] designed MF methods preserving JDP based on the alternating least squares (ALS) algorithm. Berlioz et al. [2] designed a framework for DP matrix factorization (MF), where the privacy mechanisms are classified into input, in-process and output perturbations according to the process of MF. The schemes in [14, 29] proposed to perturb the objective function of the MF problem, and learn the differentially private MF model based on the perturbed objective.

Most of the privacy-preserving studies focus on the case of explicit user preference. However, a privacy-preserving approach for implicit feedback with good utility is still missing. From the privacy perspective, implicit preferences that record which items the user is interested in also reveal a lot about the user's privacy. Indeed, Weinsberg et al. [37] validated that the implicit preference data are highly relevant with sensitive attributes such as gender and age. By exploiting merely a small part of implicit data, Narayanan et al. [26] successfully de-anonymized the user records in Netflix. As far as we know, only the works [10, 12] attempted to introduce DP to protect the implicit data, and their schemes are based on the KNN model. These methods first apply the random bit flipping technique to obfuscate the implicit data in compliance with DP, then estimate cardinality to extract similar items from the obfuscated data for predicting user preference. However, the KNN model cannot provide sufficient generalization capability due to the fact that the implicit dataset is extremely sparse and its dimension is typically very high [32].

8 CONCLUSION

This paper proposes a differentially private IMF (DPIMF) method. We introduce objective perturbation to perturb the loss function, and the MF model is learned based on this perturbed loss, ensuring compliance with differential privacy (DP). To address utility degradation, we redesign the loss function and adopt an importance sampling strategy, effectively limiting the noise scale and improving the utility of IMF. We provide theoretical proofs of the utility guarantees for the proposed schemes. Experimental results on three benchmark datasets demonstrate that the proposed strategies effectively improve model utility, and our proposed DPIMF achieves a good trade-off between privacy levels and recommendation quality.

REFERENCES

[1] Xavier Amatriain and Justin Basilico. 2015. Recommender systems in industry: A netflix case study. In *Recommender systems handbook*. Springer, 385–419.

[2] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. 2015. Applying differential privacy to matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 107–114.

[3] Joseph A Calandrino, Ann Kilzer, Arvind Narayanan, Edward W Felten, and Vitaly Shmatikov. 2011. " You might also like:" Privacy risks of collaborative filtering. In *2011 IEEE symposium on security and privacy*. IEEE, 231–246.

[4] Steve Chien, Prateek Jain, Walid Krichene, Steffen Rendle, Shuang Song, Abhradeep Thakurta, and Li Zhang. 2021. Private Alternating Least Squares: Practical Private Matrix Completion with Tighter Rates. In *International Conference on Machine Learning*. PMLR, 1877–1887.

[5] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*. 293–296.

[6] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*. Springer, 1–19.

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 265–284.

[8] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1322–1333.

[9] Arik Friedman, Bart P Knijnenburg, Kris Vanhecke, Luc Martens, and Shlomo Berkovsky. 2015. Privacy aspects of recommender systems. In *Recommender systems handbook*. Springer, 649–688.

[10] Chen Gao, Chao Huang, Dongsheng Lin, Depeng Jin, and Yong Li. 2020. DPLCF: differentially private local collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 961–970.

[11] Carlos A Gomez-Uribe and Neil Hunt. 2015. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6, 4 (2015), 1–19.

[12] Taolin Guo, Junzhou Luo, Kai Dong, and Ming Yang. 2019. Locally differentially private item-based collaborative filtering. *Information Sciences* 502 (2019), 229–246.

[13] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 549–558.

[14] Jingyu Hua, Chang Xia, and Sheng Zhong. 2015. Differentially Private Matrix Factorization. In *Proceedings of the 24th International Conference on Artificial Intelligence* (Buenos Aires, Argentina) *(IJCAI'15)*. AAAI Press, 1763–1770.

[15] Prateek Jain, Om Dipakbhai Thakkar, and Abhradeep Thakurta. 2018. Differentially private matrix completion revisited. In *International Conference on Machine Learning*. PMLR, 2215–2224.

[16] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender systems: an introduction*. Cambridge University Press.

[17] Arjan JP Jeckmans, Michael Beye, Zekeriya Erkin, Pieter Hartel, Reginald L Lagendijk, and Qiang Tang. 2013. Privacy in recommender systems. In *Social media retrieval*. Springer, 263–281.

[18] Jia-Yun Jiang, Cheng-Te Li, and Shou-De Lin. 2019. Towards a more reliable privacy-preserving recommender system. *Information Sciences* 482 (2019), 248–265.

[19] Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User modeling and user-adapted interaction* 22, 4 (2012), 441–504.

[20] Shyong K Lam, Dan Frankowski, John Riedl, et al. 2006. Do you trust your recommendations? An exploration of security and privacy issues in recommender systems. In *International conference on emerging trends in information and communication security*. Springer, 14–29.

[21] Ninghui Li, Min Lyu, Dong Su, and Weining Yang. 2016. Differential privacy: From theory to practice. *Synthesis Lectures on Information Security, Privacy, & Trust* 8, 4 (2016), 1–138.

[22] Zitao Li, Bolin Ding, Ce Zhang, Ninghui Li, and Jingren Zhou. 2021. Federated matrix factorization with privacy guarantee. *Proceedings of the VLDB Endowment* 15, 4 (2021).

[23] Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. 2015. Fast differentially private matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 171–178.

[24] Frank McSherry and Ilya Mironov. 2009. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 627–636.

[25] Frank D McSherry. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. 19–30.

[26] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 111–125.

[27] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh. 2013. Privacy-preserving matrix factorization. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 801–812.

[28] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 502–511.

[29] Xun Ran, Yong Wang, Leo Yu Zhang, and Jun Ma. 2022. A differentially private nonnegative matrix factorization for recommender system. *Information Sciences* 592 (2022), 21–35.

[30] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *Fourteenth ACM conference on recommender systems*. 240–248.

[31] Steffen Rendle, Walid Krichene, Li Zhang, and Yehuda Koren. 2021. iALS++: Speeding up Matrix Factorization with Subspace Optimization. *arXiv preprint arXiv:2110.14044* (2021).

[32] Steffen Rendle, Walid Krichene, Li Zhang, and Yehuda Koren. 2021. Revisiting the Performance of iALS on Item Recommendation Benchmarks. *arXiv preprint arXiv:2110.14037* (2021).

[33] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[34] Jun Wang and Qiang Tang. 2017. Differentially private neighborhood-based recommender systems. In *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 459–473.

[35] Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. 2015. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning*. PMLR, 2493–2502.

[36] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469.

[37] Udi Weinsberg, Smriti Bhagat, Stratis Ioannidis, and Nina Taft. 2012. BlurMe: Inferring and obfuscating user gender based on ratings. In *Proceedings of the sixth ACM conference on Recommender systems*. 195–202.

[38] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. 2019. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 13, 3 (2019), 1–207.

[39] Baolin Yi, Xiaoxuan Shen, Hai Liu, Zhaoli Zhang, Wei Zhang, Sannyuya Liu, and Naixue Xiong. 2019. Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Transactions on Industrial Informatics* 15, 8 (2019), 4591–4601.

[40] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhunmin Chen, Pengfei Hu, and Yang Zhang. 2021. Membership Inference Attacks Against Recommender Systems. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 864–879.

[41] Shun Zhang, Laixiang Liu, Zhili Chen, and Hong Zhong. 2019. Probabilistic matrix factorization with personalized differential privacy. *Knowledge-Based Systems* 183 (2019), 104864.

[42] Zhaoxi Zhang, Leo Yu Zhang, Xufei Zheng, Bilal Hussain Abbasi, and Shengshan Hu. 2022. Evaluating Membership Inference Through Adversarial Robustness. *Comput. J.* (2022).

[43] Tianqing Zhu, Yongli Ren, Wanlei Zhou, Jia Rong, and Ping Xiong. 2014. An effective privacy preserving algorithm for neighborhood-based collaborative filtering. *Future Generation Computer Systems* 36 (2014), 142–155.