Applying Differential Privacy to Matrix Factorization

Arnaud Berlioz, Arik Friedman,
Mohamed Ali Kaafar, Roksana Boreli
NICTA,* Australia
{firstname.lastname}@nicta.com.au

Shlomo Berkovsky
CSIRO
shlomo.berkosvky@csiro.au

ABSTRACT

Recommender systems are increasingly becoming an integral part of on-line services. As the recommendations rely on personal user information, there is an inherent loss of privacy resulting from the use of such systems. While several works studied privacy-enhanced neighborhood-based recommendations, little attention has been paid to privacy preserving latent factor models, like those represented by matrix factorization techniques. In this paper, we address the problem of privacy preserving matrix factorization by utilizing differential privacy, a rigorous and provable privacy preserving method. We propose and study several approaches for applying differential privacy to matrix factorization, and evaluate the privacy-accuracy trade-offs offered by each approach. We show that input perturbation yields the best recommendation accuracy, while guaranteeing a solid level of privacy protection.

Keywords

Differential privacy; matrix factorization

1. INTRODUCTION

In the last decade, recommender systems have become a fundamental tool in on-line services. One of the dominant recommendation approaches is collaborative filtering (CF), which can be partitioned into two families. Neighborhood methods learn correlations between items or users [5] and generate predictions based on their similarity. In contrast, latent factor models [12] derive models that characterize users and items with respect to a set of latent factors.

Matrix factorization (MF) methods [12] have evolved as the state-of-the-art latent factor technique. There, the rating matrix is decomposed into two low-dimensional matrices, capturing latent factors of users and items, respectively. MF has been shown to provide a higher predictive accuracy than the neighborhood methods, it is computationally cheaper, and easier to extend, for example, to consider temporal effects or ratings with varying levels of confidence.

Recommender systems rely on personal user information and raise privacy concerns related to the misuse of the collected data for inferring personal information [10]. The raw user ratings, even if anonymized, pose a privacy risk: the data can be de-anonymized using information obtained from other sources and then be used to infer sensitive information [18], e.g., gender, political views, or sexual orientation. Moreover, it was shown that even without direct access to user ratings, personal user information could be inferred from recommendations provided by the system to other users [2]. These inherent privacy risks of recommender systems motivated research of privacy-preserving recommenders [8, 10]. However, this body of research has mainly focused on neighborhood methods, with limited work on privacy preserving latent factor recommenders [14, 15].

In this paper, we approach the problem of privacy preserving MF by utilizing the concept of *differential privacy* [7], a rigorous and provable approach to privacy in statistical databases, previously applied to neighborhood based CF [13, 14]. While differential privacy sets constraints on privacy preserving computations, these computations can be carried out in various ways, which result in different privacy-accuracy trade-offs. We propose a number of approaches to alter MF, such that it maintains differential privacy guarantees. We study the privacy guarantees that can be achieved by the following approaches: (i) obfuscating the input data before applying the MF algorithm; (ii) adding noise within a stochastic gradient descent solver of the MF problem; and (iii) obfuscating the output of an alternating least squares MF mechanism. For these approaches, we provide a theoretical analysis of the (calibrated) noise level introduced in the algorithms, and empirically evaluate the resulting privacy-accuracy trade-offs by observing the effect of the noise on the computed rating predictions.

The contributions of our work are three-fold. We provide an analysis and evaluation of **three differentially private MF approaches**. The evaluation demonstrates that the best performing method, yielding the highest predictive accuracy while still ensuring a solid level of privacy protection, is input obfuscation. We further conduct an investigation of the **design choices** that affect the privacy-accuracy trade-off, showing the impact of the data pre-processing, the dependencies between the characteristics of a dataset (size,

*NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

RecSys'15, September 16–20, 2015, Vienna, Austria.

DOI: http://dx.doi.org/10.1145/2792838.2800173.

density, number of user/item ratings) and the choice of the algorithm, and the influence of the privacy constraints on parameter tuning. Finally, we **compare the accuracy** of the predictions generated by differentially private MF with that of privacy-preserving neighborhood based methods. Our experiments demonstrate that neighborhood methods are more resilient to the noise introduced by the differential privacy constraints, and are more appropriate when high levels of privacy protection are required. However, when weaker privacy levels are acceptable, privacy preserving MF techniques achieve higher levels of predictive accuracy than neighborhood based methods.

2. RELATED WORK

Personalization and recommender systems inherently bring to the fore the issue of privacy [8, 11]. Privacy hazards in recommender systems are aggravated by the fact that generation of quality recommendations requires large amounts of user data. For instance, the accuracy of CF recommendations is correlated with both the number of users in the system and the number of their ratings [5]. Hence, there is a trade-off between the accuracy of recommendations provided to users and the degree of user privacy.

We divide prior works on privacy enhanced recommender systems into two categories: distributed recommenders and data modification techniques. In the distributed group, user profiles are stored across several repositories. Canny proposed a decentralized storage of user profiles, which requires the adversary to compromise multiple systems when attacking a distributed recommender [3]. Vallet et al. [17] have shown how MF techniques can be leveraged to allow a central server to provide accurate recommendations without retention of user data, storing it on the client side instead.

Data modification techniques include approaches such as encryption [15], obfuscation [1], and randomization [16]. Polat and Du proposed to add uncertainty to user ratings through randomized data perturbation [16]: users substitute ratings in their profiles with modified ratings, resembling the real ones. Hence, if user data is exposed to an adversary, only the modified ratings will leak. Nikolaenko et al. showed how secure multiparty computation could be utilized in MF [15], so that the recommender learns only the item profiles, but not the user ratings. Such techniques, however, do not prevent the inference of user ratings from the output of MF, and are orthogonal to the techniques studied in this paper, as they address a different threat model.

Calandrino et al. [2] studied the privacy risks imposed by recommenders, such as Hunch, Last.fm, and Amazon. In item-to-item CF, when a user makes a transaction involving an item, this results in an increase of the similarity of the item to other items in the user's transaction history. Therefore, the attacker can track the similarity lists of items associated with a target user, and identify new items in the lists. When the same item appears in a number of tracked lists, the attacker can infer that the item was added to the target user's record. The authors pointed to differential privacy as a possible solution to this problem.

Differential privacy has drawn much research attention; it makes no assumptions about the adversary's background knowledge and computation power, and provides formally provable privacy guarantees [7]. To the best of our knowledge, only two works have investigated the application of differential privacy to recommender systems, although not in the immediate context of MF.

Machanavajjhala et al. studied the problem of privacy-preserving social recommendations on the basis of a graph linking between users and items, e.g., items purchased by users [13]. A utility vector derived from the graph captures the utility of each item for the target user, and the goal is to induce a probability distribution over the items, such as to maximize the user's utility, while keeping the vector private. It was found that good recommendations were achievable only under weak privacy parameters, or only for a small fraction of users.

McSherry and Mironov applied differential privacy to CF [14]. They used the Laplace mechanism to compute a differentially-private item-to-item covariance matrix, which was used to find neighbors and compute SVD recommendations. Their solution involved breaking the recommendation process into a learning phase, in which the private covariance matrix was derived, and a recommendation phase, in which the predictions were computed. In contrast, we consider direct privacy-preserving derivation of the latent factor models. Overall, our work explores additional approaches beyond those investigated in [14], and compares their performance.

3. PRELIMINARIES

3.1 MF Recommendations

The input to MF is typically a sparse rating matrix $R_{n\times m}$, containing the ratings of n users for m items. Each matrix element r_{ui} reflects the rating of user u for item i. MF factorizes $R_{n\times m}$ into two latent matrices of a lower dimension d: user-factor matrix $P_{n\times d}$ and item-factor matrix $Q_{m\times d}$. The factorization is done such that R is approximated as a product of P and Q, i.e., each known rating r_{ui} is approximated by $\hat{r}_{ui} = p_u \cdot q_i^\intercal$. To obtain P and Q, MF minimizes the regularized squared error:

$$\min_{P,Q} \sum_{r_{ui}\in R} \left[(r_{ui} - p_u q_i^\intercal)^2 + \lambda(||p_u||^2 + ||q_i||^2)\right] \quad , \qquad (1)$$

where λ regularizes the factors and prevents overfitting.

Two common ways to solve the resulting non-convex optimization problem are stochastic gradient descent (SGD) and alternating least squares (ALS). In SGD, the factors are learned by iteratively evaluating the error $e_{ui} = r_{ui} - p_u q_i^\intercal$ for each rating r_{ui}, and updating the user and item vectors by taking a step in the direction opposite to the gradient of the regularized loss function:

$$\begin{aligned} p_u &\leftarrow p_u + \gamma(e_{ui} q_i - \lambda p_u) \quad , \\ q_i &\leftarrow q_i + \gamma(e_{ui} p_u - \lambda q_i) \quad . \end{aligned} \qquad (2)$$

The constant γ determines the rate of minimizing the error and is often referred to as the learning rate.

In ALS, the optimization problem is solved iteratively. In each iteration, one latent matrix (say, P) is fixed, resulting in a convex optimization problem, where the solution (for Q) can be found efficiently. Then, the other matrix (Q) is fixed, and the optimization problem is solved again (this time for P). These steps are repeated until convergence.

3.2 Differential Privacy

Differential privacy is based on the principle that the output of a computation should not allow inference about any

particular record in the input [7]. This is achieved by requiring that the probability of any computation outcome is insensitive to small input changes. We denote two datasets A and B as *adjacent*, $A \approx B$, if they are identical in all records but one. Formally, there exist a user u and an item i such that $A = B \setminus \{r_{ui}\} \cup \{r'_{ui}\}$, where r'_{ui} and r_{ui} are the ratings that u assigned to i in A and B, respectively. The guaranteed privacy level is measured by a parameter ϵ. Formally, a randomized computation K maintains ϵ-differential privacy if for any two datasets $A \approx B$, and any subset S of possible outcomes in Range(K),

$$Pr[K(A) \in S] \leq exp(\epsilon) \times Pr[K(B) \in S] \quad , \qquad (3)$$

where the probability is over the randomness of K. Low values of ϵ correspond to a high degree of privacy. Setting the bounds for the acceptable value of ϵ is an open question. In the literature, privacy settings of $\epsilon = \ln 2$ or $\epsilon = \ln 3$ are considered as providing acceptable levels of privacy, although Dwork suggested that in some cases much higher values of ϵ could provide meaningful guarantees [6].

A common way to obtain differential privacy is by applying random noise to the measurement. The amount of noise added depends on the L_1-sensitivity of the evaluated function, which is the largest possible change in the measurement given a change in a single record in the dataset. In general, the L_k-sensitivity of a function g is given by:

$$S_k(g) = \max_{A \approx B} ||g(A) - g(B)||_k \quad , \qquad (4)$$

where $|| \cdot ||_k$ denotes the L_k-norm.

The *Laplace mechanism* [7] obtains ϵ-differential privacy by adding noise sampled from Laplace distribution, with a calibrated scale b. The probability density function of Laplace distribution with mean 0 and scale b ($x \sim Laplace(b)$) is $f_b(x) = \frac{1}{2b} \exp(-\frac{|x|}{b})$. Given a function $g : \mathcal{D} \rightarrow \mathbb{R}^d$, the following computation maintains ϵ-differential privacy [7]:

$$K(x) = g(x) + (Laplace(S_1(g)/\epsilon))^d \quad . \qquad (5)$$

For example, consider the function $\text{COUNT}_c(A)$, which counts the number of records in dataset A that satisfy condition c. It has sensitivity 1, since changing a single record affects the count by at most 1. Hence, $K(A) = \text{COUNT}_c(A) + Laplace(1/\epsilon)$ maintains ϵ-differential privacy. Consider also the function SUM(A), where $a_i \in [0, \Lambda]$. It has sensitivity Λ, which is the maximal change in the sum by changing one element of A. Hence, $K(A) = \text{SUM}(A) + Laplace(\Lambda/\epsilon)$ maintains ϵ-differential privacy.

We also rely in this work on the K-norm mechanism [9], which allows to calibrate noise to the L_2-sensitivity of the evaluated function. Given a function $g : \mathcal{D} \rightarrow \mathbb{R}^d$, the computation $K(x) = g(x) + r\alpha$ maintains ϵ-differential privacy, where r is a d-dimensional vector uniformly sampled from a d-dimensional sphere with radius 1, and $\alpha \sim \Gamma(d, S_2(g)/\epsilon)$.

4. DIFFERENTIALLY PRIVATE MF

Differential privacy sets the conditions that should be maintained to preserve privacy, but within these constraints it is often possible to implement various mechanisms that evaluate the same computation, resulting in different privacy-accuracy trade-offs. Considering the stages of the MF process, we highlight a number of possible approaches for adding differentially private noise, as shown in Figure 1.

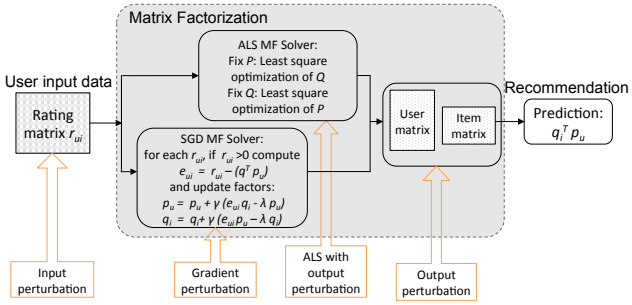


Figure 1: Various noise application points in regards to the input, output and the solver within the MF mechanism

Input Perturbation. The original MF input ratings are perturbed with a calibrated noise, and then the algorithm is trained using the noisy input ratings. Since input perturbation is performed before training the recommender, it can be followed by any recommendation algorithm, and in particular by (any variant of) MF.

In-process Mechanisms. In this approach, the algorithms used to decompose the rating matrix R into the latent matrices P and Q are adapted to maintain differential privacy. We focus in this work on two MF algorithms, and propose their differentially private variants:

SGD. In the training process of MF with SGD, in each iteration, the gradient of the regularized loss function determines the direction of the update and its magnitude. In the gradient perturbation approach the gradient is perturbed with noise in each iteration.

ALS with Output Perturbation. In each step of ALS, two optimization problems are solved to update the matrices P and Q. These empirical risk minimization problems can be solved in a differentially-private manner using the techniques proposed in [4]. In particular, we apply the output perturbation approach to obtain noisy versions of P and Q.

Output Perturbation. In this approach, a non-private MF algorithm is executed, and then the resulting latent factors are perturbed to maintain differential privacy. Unfortunately, the optimization problem in MF is non-convex, and a small change in the input could lead to a large change in the factors. Consequently, the sensitivity of the optimization problem would require introducing large noise, potentially resulting in poor utility.

Hence, in this work we restrict the evaluation to three variants of differentially private MF—input perturbation, SGD perturbation, and ALS with output perturbation—and do not consider the output perturbation approach, where noise is added to the latent factors, after a non-private MF. We first outline the data pre-processing steps that were taken before applying these approaches. For the pre-processing, we utilize the private versions of the following aggregate values, based on the training dataset (will be described in detail in Section 5.1): *global average* $GAvg(R)$ – average of all the ratings for all items; *item average* $IAvg(i)$ – average rating for item i; and *user average* $UAvg(u)$ – average rating of user u. We will now describe the three aforementioned differentially private MF approaches.

4.1 Private Preprocessing and Global Effects

Prior to applying differential privacy to MF, we preprocess the inputs as in [14]. A notable exception is that we

Algorithm 1: Evaluation of item averages

Input:
$R = \{r_{ui}\}$ – ratings of n users for m items,
β_i – stabilization parameter,
ϵ_1 – global average privacy parameter,
ϵ_2 – item average privacy parameter
Output:
Item averages IAvg(i)

GAvg $= \frac{(\sum_R r_{ui}) + \text{Laplace}(\Delta r/\epsilon_1)}{|R|}$
for $j = 1$ *to* m **do**
  Let $R_j = \{r_{ui} \in R | i = j\}$
  IAvg(j) $= \frac{(\sum_{R_j} r_{ui}) + \beta_i \cdot \text{GAvg} + \text{Laplace}(\Delta r/\epsilon_2)}{|R_j| + \beta_i}$
  Clamp IAvg(j) to $[r_{\min}, r_{\max}]$.

Algorithm 2: Evaluation of users effects

Input:
$R = \{r_{ui}\}$ – ratings of n users for m items,
β_u – stabilization parameter,
ϵ_1 – global average privacy parameter,
ϵ_2 – user average privacy parameter
Output:
User averages UAvg(u)

Let $R' = \{r_{ui} - \text{IAvg}(i) | r_{ui} \in R\}$
GAvg$' = \frac{(\sum_{R'} r'_{ui}) + \text{Laplace}(\Delta r/\epsilon_1)}{|R'|}$
for $v = 1$ *to* n **do**
  Let $R_v = \{r'_{ui} \in R' | u = v\}$
  UAvg(v) $= \frac{(\sum_{R_v} r'_{ui}) + \beta_u \cdot \text{GAvg}' + \text{Laplace}(\Delta r/\epsilon_2)}{|R_v| + \beta_u}$
  Clamp UAvg(v) to $[-2, 2]$

incorporate the user averages in rating predictions, as this allows to derive more accurate predictions when using MF. The preprocessing consists of the following three steps.

Firstly, we compute the (differentially-private) average item ratings according to the process described in Algorithm 1. We add a number of fictitious ratings β_i with the global average GAvg to stabilize the item averages—this limits the effect of noise for items with few ratings, while only slightly affecting the average for items with many ratings. If the added noise causes an item average to go beyond the range of ratings $[r_{\min}, r_{\max}]$, the average is clamped to fit the range. Differential privacy is guaranteed by adding noise calibrated to the L_1-sensitivity of ratings, given by $\Delta r = r_{\max} - r_{\min}$.

Secondly, we follow the same technique to compute the user averages, as outlined in Algorithm 2. The basis for the user averages is the ratings after the item average discounting. We stabilize the user effects with the addition of β_u fictitious ratings with the newly computed global average. The user averages are also clamped to a bounded range (in the experiments[1] we used $[-2, 2]$ for user averages).

Finally, the item and user averages are discounted from the rating matrix R, and the resulting ratings are clamped. The clamping reduces the L_1-sensitivity of the computations conducted during the MF process, and results in a lower magnitude of noise being introduced in the differentially private computation. We denote the clamping parameter B (set to 1 in the experiments), i.e., $r_{ui} \in [-B, B]$. The pre-processed matrix R is passed to the MF algorithm to derive the matrices P and Q. Predicted ratings are then obtained through $\hat{r}_{ui} = \text{IAvg}(i) + \text{UAvg}(u) + p_u q_i^\intercal$.

[1]In the evaluation we used the MovieLens dataset with the rating scale of 1 to 5 stars.

Algorithm 3: Differentially Private Input Perturbation

Input:
$R = \{r_{ui}\}$ – preprocessed user ratings,
d – number of factors,
λ – regularization parameter,
B – clamping parameter,
ϵ – privacy parameter
Output:
Latent factor matrices $P_{n\times d}$ and $Q_{m\times d}$

Let $R' = \{r_{ui} + \text{Laplace}(\frac{\Delta r}{\epsilon}) | r_{ui} \in R\}$
Clamp the ratings in R' to the range $[-B, B]$
$(P, Q) = \min_{P,Q} \sum_{R'} [(r'_{u,i} - p_u q_i^\intercal)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)]$
return P and Q

Differential privacy maintains the composability property: if each computation in a series of computations is ϵ_i-differentially private, then the overall algorithm is $\sum_i \epsilon_i = \epsilon$-differentially private. Accordingly, the overall privacy budget ϵ is divided between the computation of global averages, item effects, user effects, and, lastly, MF. Note that it is possible to predict ratings using only the user and item averages, $\hat{r}_{ui} = \text{IAvg}(i) + \text{UAvg}(u)$, which is referred to as *Global Effects* (see Comparison Baselines in Section 5.1). This leads to its differentially-private counterpart, as described in the above three pre-processing steps. In this case, as MF is not applied, the privacy budget is divided between three computations: global averages, item averages and user averages. We refer to this technique as *Private Global Effects*.

4.2 Private Input Perturbation

In input perturbation the Laplace mechanism is applied directly to each input rating. Following data pre-processing, the sensitivity of the inputs is $\Delta r = r_{\max} - r_{\min} = 2B$, and perturbing each rating with noise sampled from the distribution Laplace($\Delta r/\epsilon$) ensures ϵ-differential privacy.[2] The noisy ratings can then be clamped again, to limit the influence of excessive noise. Algorithm 3 summarizes this process.

4.3 Private SGD

The gradient perturbation approach, outlined in Algorithm 4, guarantees privacy throughout the MF process by introducing noise in the SGD step in each iteration of the algorithm. The error calculation conducted in each step is carried out with the Laplace mechanism to maintain differential privacy, and consequently the SGD step maintains differential privacy. Optionally, the noisy error can be clamped to constrain the effect of noise (in our experiments we used $e_{\max} = 2$). The number of iterations k should be known in advance, so the noise introduced in each iteration is calibrated to maintain ϵ/k-differential privacy. Composability ensures that the k iterations maintain the overall bound of ϵ-differential privacy.

[2]Proofs of the differential privacy properties of algorithms in Sections 4.2-4.4 are omitted due to space limitations.

Algorithm 4: Differentially Private SGD

Input:
$R = \{r_{ui}\}$ – preprocessed user ratings,
d – number of factors,
γ – learning rate parameter,
λ – regularization parameter,
k – number of gradient descent iterations,
$e_{\max}$ – upper bound on per-rating error,
ϵ – privacy parameter

Output:
Latent factor matrices $P_{n\times d}$ and $Q_{m\times d}$

Initialize random factor matrices P and Q.
for k *iterations* **do**
  for *each* $r_{ui} \in R$ **do**
    $e'_{ui} = r_{ui} - p_u q_i^\intercal + \text{Laplace}(k\Delta r/\epsilon)$
    Clamp e'_{ui} to $[-e_{\max}, e_{\max}]$
    $q_i \leftarrow q_i + \gamma(e'_{ui} \cdot p_u^\intercal - \lambda \cdot q_i)$
    $p_u \leftarrow p_u + \gamma(e'_{ui} \cdot q_i^\intercal - \lambda \cdot p_u)$
return P and Q.

4.4 Private ALS with Output Perturbation

The basic idea of ALS is to alternately fix one of the latent matrices P and Q, and optimize the regularized loss function for the other matrix. Once one matrix is fixed, the optimization problem becomes convex and can be solved analytically. For example, once Q is fixed, the overall regularized loss function can be minimized by considering for each user u the following loss function defined over the subset of ratings $R_u = \{r_{vi} \in R | v = u\}$:

$$J_Q(p_u, R) = \Big[\sum_{R_u}(r_{ui} - p_u q_i^\intercal)^2\Big] + n_u\lambda\|p_u\|^2 \quad , \qquad (6)$$

where $n_u = |R_u|$. Each user vector p_u is then obtained by solving the risk minimization problem

$$p_u(R, Q) = \arg\min_{p_u} J_Q(p_u, R) \quad . \qquad (7)$$

The problem of differentially private empirical risk minimization (ERM) was studied by Chaudhuri et al. [4]. An adaptation of their techniques shows that the L_2-sensitivity of $p_u(R, Q)$ in Equation 7, is $\Delta p_u = \frac{q_{\max}\Delta r}{n_u\lambda}$, where $q_{\max}$ is the upper bound on the L_2-norm of each row q_i in Q. Similarly, when fixing P and optimizing Q based on the regularized loss function $J_P(q_i, R) = [\sum_{R_i}(r_{ui} - p_u q_i^\intercal)^2] + n_i\lambda\|q_i\|^2$, the L_2-sensitivity of each row q_i is $\frac{p_{max}\cdot\Delta r}{\lambda n_i}$. For the preprocessed ratings, we have $\Delta r = 2B$, where B is the clamping parameter. Since we calculated the L_2-sensitivity of the user-vector p_u and item-vector q_i, the noise added to these vectors is taken from the Gamma distribution.

Following the above analysis, Algorithm 5 outlines a differentially-private ALS algorithm with output perturbation. Similarly to the SGD approach, we calibrate the noise so that each optimization problem is $\epsilon/2k$-differentially private and the overall ALS computation is ϵ-differentially private due to composability.

5. EVALUATION

In this section, we present the results of the evaluation of the proposed differentially private MF approaches.

Algorithm 5: Differentially Private ALS with Output Perturbation

Input:
$R = \{r_{ui}\}$ – preprocessed user ratings,
d – number of factors,
λ – regularization parameter,
k – number of ALS iterations,
ϵ – privacy parameter,
$p_{\max}$ – upper bound on $||p_u||_2$,
$q_{\max}$ – upper bound on $||q_i||_2$

Output:
Latent factor matrices $P_{n\times d}$ and $Q_{m\times d}$

Initialize random factor matrices P and Q.
for k *iterations* **do**
  for *each user* u*, given* Q **do**
    Sample noise vector b with pdf $f(b) \propto \exp\left(-\frac{\epsilon\cdot\|b\|_2}{2k} \cdot \frac{n_u\lambda}{p_{\max}\cdot\Delta r}\right)$
    $p_u \leftarrow \arg\min_{p_u} J_Q(p_u, R_u) + b$
    if $||p_u||_2 > p_{\max}$ **then** $p_u \leftarrow p_u \cdot \frac{p_{\max}}{||p_u||_2}$
  for *each item* i*, given* P **do**
    Sample noise vector b with pdf $f(b) \propto \exp\left(-\frac{\epsilon\cdot\|b\|_2}{2k} \cdot \frac{n_i\lambda}{q_{\max}\cdot\Delta r}\right)$
    $q_i \leftarrow \arg\min_{q_i} J_P(q_i, R_i) + b$
    if $||q_i||_2 > q_{\max}$ **then** $q_i \leftarrow q_i \cdot \frac{q_{\max}}{||q_i||_2}$
return P and Q.

	ML-100K	ML-1M	ML-10M
Users	943	6040	71567
Movies	1682	3952	65133
Density	6.3%	4.19%	0.21%
Average rating	3.5299	3.5816	3.5124
Variance of ratings	1.2671	1.2479	1.1245
Avg. ratings per user	106	165.6	139.7
Avg. ratings per item	59.4	253	153.5

Table 1: Statistical properties of the MovieLens datasets

5.1 Experimental Setting

We use in the evaluation the 100K, 1M and 10M MovieLens datasets. Table 1 summarizes selected statistical properties of the datasets.

We use 10-fold cross validation[3] to train and evaluate the recommender system. We measure the accuracy of the predicted ratings $\hat{r}_{ui}$ using the Root Mean Square Error (RMSE) metric (averaged over all the ratings), computed by $\text{RMSE} = \left(\sum_R (r_{ui} - \hat{r}_{ui})^2/|R|\right)^{\frac{1}{2}}$. Due to the possible discrepancies in the introduction of noise, the reported RMSE is averaged across multiple runs.

We compare the performance of the privacy-preserving MF approaches against the following baselines:

Global average: the average rating is computed over the entire training set, and used as the prediction for all the ratings in the test set, i.e., $\hat{r}_{ui} = GAvg(R)$. We treat the global average RMSE as the upper bound for error.

Item average: the average rating for each item is computed over all the available training item ratings, and used as the prediction for all the ratings for that item in the test

[3]We used Matlab and, specifically, *crossvalind*.

	ML-100K	ML-1M	ML-10M
Parameter settings			
Number of factors	3	5	7
Regularizer	0.06	0.045	0.03
Baseline RMSE			
Global average	1.1256	1.1171	1.0604
Item average (IA)	1.0278	0.9795	0.9436
Global effects (GE)	0.9571	0.9161	0.8738
ALS	0.9198	0.8604	0.8013
Private Global Effects			
IA ϵ-crossing	0.5	0.2	0.18
Input Perturbation (ISGD)			
IA ϵ-crossing	2	0.9	0.7
GE ϵ-crossing	5	2.7	2.1
Stochastic Gradient Perturbation (PSGD)			
IA ϵ-crossing	2	0.8	0.6
GE ϵ-crossing	20	8	5.5
ALS with Output Perturbation (PALS)			
IA ϵ-crossing	2	0.8	0.6
GE ϵ-crossing	19	8	6

Table 2: Summary of the experimental settings and results.

set, i.e., $\hat{r}_{ui} = IAvg(i)$. This baseline reflects the RMSE score attainable without personalization.

Global effects: the average ratings IAvg(i) for each item i and UAvg(u) for each user u are computed over the entire training set. The item and user biases are both used when predicting the test ratings, $\hat{r}_{ui}$ = IAvg(i) + UAvg(u). We treat this baseline as the most simple way to obtain personalization, and we consider RMSE scores below this baseline to represent effective personalization.

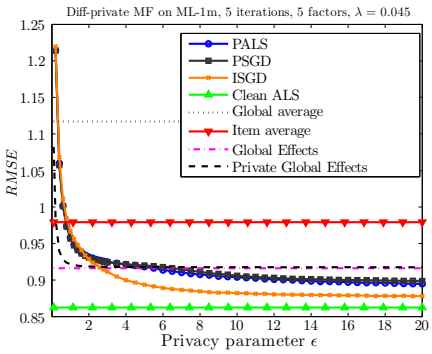
Clean MF: ALS is executed to solve the MF problem without any noise. These RMSE scores reflect the lower bound for error attainable with no privacy constraints.

We use the item average and global effects baselines to assess the privacy-accuracy trade-off offered by the private approaches. To this end, we measure the values of the privacy parameter ϵ for which the RMSE scores attained by each algorithm *cross the RMSE* scores of these baselines, where low values of ϵ indicate that the algorithm can provide the same level of accuracy as the baseline with a low cost in privacy. Thus, the focus is on the privacy-accuracy trade-offs of the approaches, rather than on evaluating their performance for certain values of ϵ. Also, we investigate several factors that may affect the system performance.

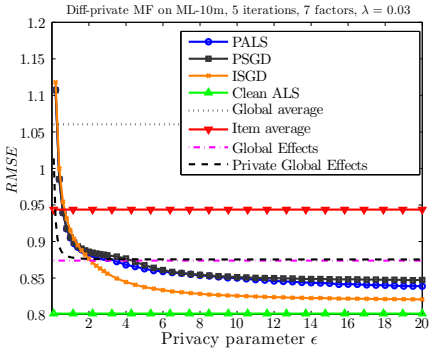
5.2 Comparison of MF Approaches

In each experiment, given the overall ϵ-differential privacy constraint, we allocated 0.3ϵ to pre-processing. Out of this, 0.02ϵ was used to compute the global averages (split between the user and item average calculations), whereas the user and item averages were computed with 0.14ϵ each. The remaining privacy budget of 0.7ϵ was allocated to MF. This distribution of the privacy budget is based on an offline optimization, which is beyond the scope of the paper.

Where applicable, we bounded the L_2-norm of the user vectors to $p_{\max} = 0.4$, and of the item vectors to $q_{\max} = 0.5$. In both the SGD and ALS experiments, we set the number of iterations to $k = 5$. The number of iterations for input perturbation was set to $k = 20$. Table 2 details several other dataset specific parameters, which were set in an offline optimization. We note that the selected number of factors and the number of iterations were lower than typical for these algorithms, to limit the amount of noise introduced by



(a) MovieLens-1M



(b) MovieLens-10M

Figure 2: Differentially-private MF approaches

differential privacy. Table 2 shows also the baseline RMSE scores measured for each dataset, and the values of ϵ for which each approach crossed those baselines.

Figures 2a and 2b show the privacy-accuracy trade-offs for all the approaches observed for the MovieLens-1M and MovieLens-10M datasets, respectively.[4] In addition to the aforementioned baselines, the figures show the results for the Input Perturbation approach followed by a non-private SGD algorithm (ISGD), the Private SGD approach (PSGD), and the Private ALS approach (PALS).

In general, the performance of all the approaches improves with the size of the dataset. For example, ISGD crosses the IA baseline for MovieLens-100K, MovieLens-1M, and MovieLens-10M at $\epsilon = 2$, $\epsilon = 0.9$ and $\epsilon = 0.7$, respectively. This is not surprising; the larger the dataset, the more resilient it is to the noise introduced through differential privacy. Since the noise is calibrated to mask the effect of a single rating, larger datasets provide a higher signal-to-noise ratio, thereby allowing better performance with respect to the baseline for any value of ϵ.

As expected, crossing of the IA is observed for lower values of ϵ than crossing of the GE baseline. This is explained by the lower degree of personalization offered by IA, which is achievable with higher levels of noise and, therefore, a higher degree of privacy. For all the datasets, the IA crossing values of the approaches are similar, but there is a substantial difference between the GE crossings. Specifically, the IA crossings of PSGD and PALS are very close, and both are

[4]Results obtained for MovieLens-100K exhibit a similar trend and are not shown, but are summarized in Table 2.

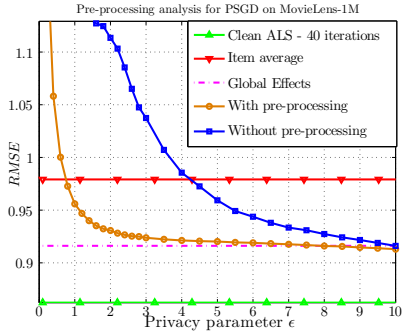


Figure 3: The effect of pre-processing

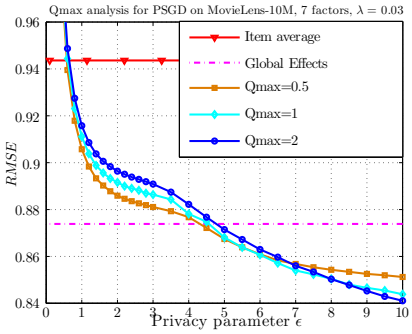


Figure 4: The effect of factor vector $L2$-norm bounds

slightly lower than that of ISGD. However, the GE crossing of ISGD is much lower than those of PSGD and PALS.

For example, consider the MovieLens-10M dataset. The PSGD and PALS approaches both cross the IA baseline at $\epsilon = 0.6$, whereas ISGD crosses it at $\epsilon = 0.7$. This is explained by the fact that the matrices P and Q in PALS and PSGD are bounded with $L2$-norm bounds p_{max} and q_{max}. Bounding the $L2$-norm provides a small improvement for low values of ϵ and gives PALS and PSGD a slightly earlier crossing. However, for higher ϵ, ISGD achieves a better performance; it crosses the GE baseline at $\epsilon = 2.1$, whereas PSGD and PALS cross it at $\epsilon = 5.5$ and $\epsilon = 6$, respectively. Similar trade-offs were observed also for other datasets.

5.2.1 Impact of pre-processing and L2-norm bounds

Figure 3 shows two variants of the PSGD approach, evaluated using the MovieLens-1M dataset: the RMSE curve of PSGD extracted from Figure 2a and the curve of PSGD with exactly the same parameters but with no data pre-processing. Data pre-processing has a substantial effect on the RMSE, as it reduces the sensitivity and the required levels of noise—in particular for low ϵ, when the IA baseline crossing is considered. Similar trends were observed also for the PALS and ISGD approaches, and for other datasets.

We also demonstrate the effect of bounding q_{max} in PSGD. Specifically, we set p_{max} to 80% of the q_{max} value, while the regularizer λ and the number of factors d are fixed to $\lambda = 0.03$ and $d = 7$. We conduct the experiment using three values of q_{max}: $q_{max} = 0.5$, $q_{max} = 1$, and $q_{max} = 2$. Figure 4 shows the results obtained for MovieLens-10M. While the value of q_{max} does not affect much the crossing of the IA baseline, it changes the value of the GE crossings and

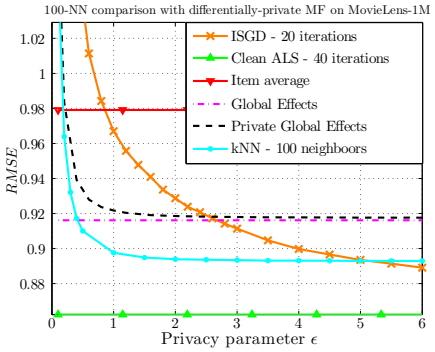


Figure 5: Comparison of private MF and kNN algorithms

the accuracy achieved for higher values of ϵ. In PSGD, the $L2$-norm bounds do not affect the noise added to P and Q and are used only to control the $L2$-norm of the latent vectors. For low ϵ, more noise is added and a small bound is preferable over greater bounds, since it removes the noisy elements. However, for higher values of ϵ, a small bound prevents MF from fully realizing the potential of the seven factors, and, therefore, a higher bound achieves a better predictive accuracy when less noise is added.

5.3 Comparison to other CF Approaches

In this experiment we compare the results of the privacy-preserving MF approach to two other privacy preserving CF algorithms: a private version of the GE baseline and private k-Nearest Neighbors (kNN) algorithm [5].

For private GE, we used the following allocation for the privacy budget: 0.02ϵ for the global average, 0.54ϵ for item average and 0.44ϵ for user average. For the private kNN recommendation algorithm, we followed the approach of McSherry and Mironov [14].[5] We applied a different privacy budget allocation: 0.9ϵ was allocated to data pre-processing, out of which 0.02ϵ was used for the global average, while the item and user averages were computed with 0.44ϵ each, and the remaining 0.1ϵ was used for the identification of nearest neighbors. It should be highlighted that kNN combines the differentially private item-to-item covariance matrix with the private user ratings, giving it an *a-priori* advantage over the proposed differentially private MF algorithms.

Figure 5 shows the comparison of the private versions of MF and kNN for the MovieLens-1M dataset.[6] For low values of ϵ, computing only the private GE was more effective than the MF approaches (both kNN and private GE cross the IA baseline at $\epsilon = 0.18$), since it makes the smallest number of computations and introduces the lowest amount of noise. However, this approach cannot outperform the GE baseline, and therefore cannot take advantage of weaker privacy constraints, when available.

While latent factors models typically outperform neighborhood based approaches in terms of predictive accuracy [5, 12], surprisingly this is not the case in the presence of privacy constraints. For lower values of ϵ, the improved ac-

[5]The differentially private implementation of kNN outlined in [14] is not publicly available, such that we were not able to reproduce the exact results reported therein.

[6]Due to memory limitations, kNN implementation for the MovieLens-10M dataset was not feasible.

curacy offered by MF in the non-private settings does not compensate for the higher noise required to meet the privacy constraints. However, for higher ϵ and weaker privacy, the predictive accuracy advantage of MF becomes apparent, and it outperforms the private kNN algorithm.

We posit that the superiority of private neighborhood based approaches over the MF approaches is explained by their better resilience to the noise introduced by differential privacy. Linking the average number of user ratings (Table 1) with the number of latent factors (Table 2), we observe that each factor relies, on average, on a few dozens of ratings. Hence, applying even moderate noise deteriorates the signal-to-noise ratio and affects the predictions. In contrast, the private item-to-item covariance matrix relies on thousands of ratings and is more resilient to noise. Due to this, kNN outperforms MF for lower values of ϵ (stringent privacy constraints). However, higher values of ϵ (lenient privacy constraints) allow decreasing the level of noise applied, such that MF approaches outperform kNN.

6. DISCUSSION

To address privacy concerns of recommender systems, we investigated the application of differential privacy to MF, the state-of-the-art recommendation approach. Differential privacy does not dictate a specific way to conduct a computation, but is rather a property that should be maintained. Hence, it is possible to design various approaches that carry out the same computation in a differentially private manner, with different levels of effectiveness. We proposed and evaluated three approaches reflecting the stages of MF: input perturbation, and differentially private variants of ALS and SGD. We also analyzed the sensitivity of the proposed approaches and compared private MF to other privacy preserving recommender approaches, namely, GE and kNN.

We showed that input perturbation yields the best performance amongst the three evaluated private MF approaches. However, when privacy is a priority and high degree of noise is applied, private kNN outperforms MF. We believe this observation is inherent to sparse datasets and stringent privacy requirements, as kNN is not as sensitive to noise as MF. On the other hand, when weaker privacy settings are acceptable, MF offers a better alternative: in that case, the predictive accuracy of the private algorithms gets closer to that of the respective non-private variants, and MF is shown to outperform other private recommendation approaches.

Following our evaluation, we identified the following design choices that should be considered when applying differential privacy to recommender systems.

Contextual considerations. Data characteristics, such as size, density and the distribution of ratings, may affect the privacy-accuracy trade-offs of the approaches. Beyond these, additional factors need to be considered. For example, the scalability and flexibility of the model-based approaches may outweigh the advantage of neighborhood methods in privacy protection, making privacy-preserving MF algorithm a viable option. Also, methods like input perturbation may be more amenable to the processing of streaming data, since each new rating can be perturbed independently, whereas for other approaches further work is required to adapt incremental learning models to the private setting.

Mind your parameters. Typically, MF parameters such as the number of factors, the regularizer, and the learning rate are tuned to increase prediction accuracy, while preventing over-fitting and ensuring convergence. In the private setting, these considerations should be augmented to incorporate their impact on the introduced noise. For example, increasing the number of factors results in larger L_2-norms of the latent vectors, and requires larger magnitudes of noise to obtain the same level of privacy. This noise abolishes the increased accuracy driven by the additional factors, and parameter tuning is needed to balance these effects.

7. REFERENCES

[1] S. Berkovsky, T. Kuflik, and F. Ricci. The impact of data obfuscation on the accuracy of collaborative filtering. *Expert Syst. Appl.*, 39(5):5033–5042, 2012.

[2] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. "You might also like:" privacy risks of collaborative filtering. In *IEEE S&P*, pages 231–246, 2011.

[3] J. Canny. Collaborative filtering with privacy. In *IEEE S&P*, pages 45–57, 2002.

[4] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Mach. Learn.*, 12:1069–1109, 2011.

[5] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Rec. Sys. Handbook*, pages 107–144. 2011.

[6] C. Dwork. Differential privacy: a survey of results. In *TAMC*, pages 1–19, 2008.

[7] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TOC*, pages 265–284, 2006.

[8] A. Friedman, B. Knijnenburg, K. Vanhecke, L. Martens, and S. Berkovsky. Privacy aspects of recommender systems. In *Rec. Sys. Handbook*. 2015.

[9] M. Hardt and K. Talwar. On the geometry of differential privacy. In *STOC*, pages 705–714, 2010.

[10] A. J. Jeckmans, M. Beye, Z. Erkin, P. Hartel, R. L. Lagendijk, and Q. Tang. Privacy in recommender systems. In *Social Media Retrieval*. 2013.

[11] A. Kobsa. Privacy-enhanced web personalization. In *The Adaptive Web*, pages 628–670, 2007.

[12] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computers*, 42(8):30–37, Aug. 2009.

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

[14] F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *KDD*, pages 627–636, 2009.

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

[16] H. Polat and W. Du. Achieving private recommendations using randomized response techniques. In *PAKDD*, pages 637–646, 2006.

[17] D. Vallet, A. Friedman, and S. Berkovsky. Matrix factorization without user data retention. In *PAKDD*, 2014.

[18] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft. BlurMe: inferring and obfuscating user gender based on ratings. In *RecSys*, pages 195–202, 2012.