Differentially Private Neighborhood-Based Recommender Systems

Jun Wang[1](✉) and Qiang Tang[2]

[1] University of Luxembourg, Esch-sur-Alzette, Luxembourg
jun.wang@uni.lu
[2] Luxembourg Institute of Science and Technology, Esch-sur-Alzette, Luxembourg
tonyrhul@gmail.com

Abstract. In this paper, we apply the differential privacy concept to neighborhood-based recommendation methods (NBMs) under a probabilistic framework. We first present a solution, by directly calibrating Laplace noise into the training process, to differential-privately find the maximum a posteriori parameters *similarity*. Then we connect differential privacy to NBMs by exploiting a recent observation that sampling from the scaled posterior distribution of a Bayesian model results in provably differentially private systems. Our experiments show that both solutions allow promising accuracy with a modest privacy budget, and the second solution yields better accuracy if the sampling asymptotically converges. We also compare our solutions to the recent differentially private matrix factorization (MF) recommender systems, and show that our solutions achieve better accuracy when the privacy budget is reasonably small. This is an interesting result because MF systems often offer better accuracy when differential privacy is not applied.

Keywords: Recommender · Collaborative filtering · Differential privacy

1 Introduction

Recommender systems, particularly collaborative filtering (CF) systems, have been widely deployed due to the success of E-commerce [25]. There are two dominant approaches in CF. One is matrix factorization (MF) [12] which models the user preference matrix as a product of two low-rank user and item feature matrices, and the other is neighborhood-based method (NBM) which leverages the *similarity* between items or users to estimate user preferences [7]. Generally, MF is more accurate than NBM [25], while NBM has an irreplaceable advantage that it naturally explains the recommendation results. In reality, industrial CF recommender and ranking systems often adopt a client-server model, in which a single server (or, server cluster) holds databases and serves a large number of users. CF exploits the fact that similar users are likely to prefer similar products, unfortunately this property facilitates effective user de-anonymization and

© IFIP International Federation for Information Processing 2017
Published by Springer International Publishing AG 2017. All Rights Reserved
S. De Capitani di Vimercati and F. Martinelli (Eds.): SEC 2017, IFIP AICT 502, pp. 459–473, 2017.
DOI: 10.1007/978-3-319-58469-0_31

history information recovery through the recommendation results [5,18]. To this end, NBM is more fragile (e.g. [5,16]), since it is essentially a simple linear combination of user history information which is weighted by the normalized *similarity* between users or items. In this paper, we aim at preventing information leakage from the recommendation results, for the NBM systems. Note that a related research topic is to avoid the server from accessing the users' plaintext inputs, and many solutions exist for this (e.g. [19,26]). We skip the details here.

Differential privacy [9] provides rigorous privacy protection for user information in statistical databases. Intuitively, it offers a participant the possibility to deny his participation in a computation. Some works, such as [14,33], have been proposed for some specific NBMs, which adopt correlations or artificially defined metrics as *similarity* [7] and are less appealing from the perspective of accuracy. It remains as an open issue to apply the differential privacy concept to more sophisticated NBM models, which automatically learn *similarity* from training data (e.g. [22,27,29]). Particularly, probabilistic NBM [29] models the dependencies among observations (ratings) which leads user preference estimation to a penalized risk minimization problem to search optimal unobserved factors (In our context, the unobserved factor is *similarity*). It has been shown that the instantiation in [29] outperforms most other NBM systems and even the MF or probabilistic MF systems in many settings.

1.1 Our Contribution

Due to its accuracy advantages, we focus on the probabilistic NBM systems in our study. Inspired by [4,13], we propose two methods to instantiate differentially private solutions. First, we calibrate noise into the training process (i.e. SGD) to differential-privately find the maximum a posteriori *similarity.* This instantiation achieves differential privacy for each rating value. Second, we link the differential privacy concept to probabilistic NBM, by sampling from scaled posterior distribution. For the sake of efficiency, we employ a recent MCMC method, namely Stochastic Gradient Langevin Dynamics (SGLD) [32], as the sampler. In order to use SGLD, we derive an unbiased estimator of *similarity* gradient from a mini-batch. This instantiation achieves differential privacy for every user profile (rating vector). Our experimental results show that differentially private MFs are more accurate when privacy loss is large (extremely, in a non-private case), but differentially private NBMs are better when privacy loss is set in a more reasonable range. Even with the added noises, both our solutions consistently outperform non-private traditional NBMs in accuracy. Despite the complexity concern, our solution with posterior sampling (i.e. SGLD) outperforms the other from the accuracy perspective.

2 Preliminary

Generally, NBMs can be divided into user-user approach (relies on *similarity* between users) and item-item approach (relies on *similarity* between items) [7].

Probabilistic NBM can be regarded as a generic methodology, to be employed by any other specific NBM system. Commonly, the item-item approach is more accurate and robust than the user-user approach [7, 16]. In this paper, we take the item-item approach as an instance to introduce the probabilistic NBM concept from [29]. We also review the concept of differential privacy.

2.1 Review Probabilistic NBM

r_{ui}	The rating that user u gave item i
s_{ij}	The similarity between item i and j
$R \in \mathbb{R}^{N\times M}$	Rating matrix
$R^{>0} \subset R$	All the observed ratings or training data
$S \in \mathbb{R}^{M\times M}$	Item similarity matrix
$S_i \in \mathbb{R}^{1\times M}$	Similarity vector of item i
$R_u^- \in \mathbb{R}^{M\times 1}$	u's rating vector without the item being modeled
α_S, α_R	Hyperparameters of S_i and r_{ui} respectively
$f(S_i, R_u^-)$	Any NBM which takes as input the S_i and R_u^-
$p(*)$	Prior distribution of $*$
$p(S_i\|\alpha_S)$	Likelihood function of S_i conditioned on α_S
$p(r_{ui}\|f(*), \alpha_R)$	Likelihood function of r_{ui}

Suppose we have a dataset with N users and M items. Probabilistic NBM [29] assumes the observed ratings $R^{>0}$ conditioned on historical ratings with Gaussian noise. Some notation is summarized in the above table. The likelihood function of observations $R^{>0}$ and prior of *similarity* S are written as

$$p(R^{>0}|S, R^-, \alpha_R) = \prod_{i=1}^{M}\prod_{u=1}^{N}[\mathcal{N}(r_{ui}|f(S_i, R_u^-), \alpha_R^{-1})]^{I_{ui}}; \qquad p(S|\alpha_S) = \prod_{i=1}^{M}\mathcal{N}(S_i|0, \alpha_S^{-1}\mathbf{I}) \tag{1}$$

where $\mathcal{N}(x|\mu, \alpha^{-1})$ denotes the Gaussian distribution with mean μ and precision α. R^- indicates that if item i is being modeled then it is excluded from the training data $R^{>0}$. $f(S_i, R_u^-)$ denotes any NBM which takes as inputs the S_i and R_u^-. In the following, we instantiate it to be a typical NBM [7]:

$$\hat{r}_{ui} \leftarrow f(S_i, R_u^-) = \bar{r}_i + \frac{\sum_{j\in\mathcal{I}\backslash\{i\}} s_{ij}(r_{uj} - \bar{r}_j)I_{uj}}{\sum_{j\in\mathcal{I}\backslash\{i\}} |s_{ij}|I_{uj}} = \frac{S_i R_u^-}{|S_i|I_u^-} \tag{2}$$

$\hat{r}_{ui}$ denotes the estimation of user u's preference on item i, $\bar{r}_i$ is item i's mean rating value, I_{uj} is the rating indicator $I_{uj} = 1$ if user u rated item j, otherwise, $I_{uj} = 0$. Similar with R_u^-, I_u^- denotes user u's indicator vector but set $I_{ui} = 0$ if i is the item being estimated. For the ease of notation, we will omit the term $\bar{r}_i$

and present Eq. (2) in a vectorization form in favor of a slightly more succinct notation. The log of the posterior distribution over the *similarity* is

$$-\operatorname{log}p(S|R^{>0},\alpha_S,\alpha_R) = -\log p(R^{>0}|S,R^{-},\alpha_R)p(S|\alpha_S) =$$
$$\frac{\alpha_R}{2}\sum_{i=1}^{M}\sum_{u=1}^{N}(r_{ui}-\frac{S_iR_u^{-}}{|S_i|I_u^{-}})^2+\frac{\alpha_s}{2}\sum_{i=1}^{M}(||S_i||_2)+M^2\log\frac{\alpha_s}{\sqrt{2\pi}}+\log\frac{\alpha_R}{\sqrt{2\pi}}\sum_{i=1}^{M}\sum_{u=1}^{N}I_{ui} \tag{3}$$

Thanks to the simplicity of the log-posterior distribution (i.e. $\sum_{i=1}^{M}\sum_{u=1}^{N}(r_{ui}-\frac{S_iR_u^{-}}{|S_i|I_u^{-}})^2+\sum_{i=1}^{M}(||S_i||_2)$, where we omit the constant terms in Eq. (3)). We can have two approaches to solve this risk minimization problem.

- *Stochastic Gradient Descent (SGD).* In this approach, $\log p(S|R^{>0},\alpha_S,\alpha_R)$ is treated as an error function. SGD can be adopted to minimize the error function. In each SGD iteration we update the gradient of *similarity* $(-\frac{\partial\log p(S|R^{>0},\alpha_S,\alpha_R)}{\partial S_{ij}})$ with a set of randomly chosen ratings Φ by

$$S_{ij}\leftarrow S_{ij}-\eta(\sum_{(u,j)\in\Phi}(\hat{r}_{ui}-r_{ui})\frac{\partial\hat{r}_{ui}}{\partial S_{ij}}+\lambda S_{ij}) \tag{4}$$

  where η is the learning rate, $\lambda=\frac{\alpha_S}{\alpha_R}$ is the regular parameter, the set Φ may contain $n\in[1,N]$ users. In Sect. 3, we will introduce how to build the differentially private SGD to train probabilistic NBM.
- *Monte Carlo Markov Chain (MCMC).* We estimate the predictive distribution of an unknown rating by a Monte Carlo approximation. In Sect. 4, we will connect differential privacy to samples from the posterior $p(S|R^{>0},\alpha_S,\alpha_R)$, via Stochastic Gradient Langevin Dynamics (SGLD) [32].

2.2 Differential Privacy

Differential privacy [9], which is a dominate security definition against inference attacks, aims to rigorously protect sensitive data in statistical databases. It allows to efficiently perform machine learning tasks with quantified privacy guarantee while accurately approximating the non-private results.

Definition 1 *(Differential Privacy [9]). A random algorithm $\mathcal{M}$ is (ϵ,σ) -differentially private if for all $\mathcal{O}\subset$ Range$(\mathcal{M})$ and for any of all $(\mathcal{D}_0,\mathcal{D}_1)$ which only differs on one single record such that $||\mathcal{D}_0-\mathcal{D}_1||\le 1$ satisfies*

$$Pr[\mathcal{M}(\mathcal{D}_0)\in\mathcal{O}]\le exp(\epsilon)Pr[(\mathcal{M}(\mathcal{D}_1)\in\mathcal{O}]+\sigma$$

And $\mathcal{M}$ guarantees ϵ-differential privacy if $\sigma=0$.

The parameter ϵ states the difference of algorithm $\mathcal{M}$'s output for any $(\mathcal{D}_0,\mathcal{D}_1)$. It measures the privacy loss. Lower ϵ indicates stronger privacy protection.

Laplace Mechanism [8] is a common approach to approximate a real-valued function $f : \mathcal{D} \rightarrow \mathbb{R}$ with a differential privacy preservation using additive noise sampled from Laplace distribution: $\mathcal{M}(\mathcal{D}) \overset{\Delta}{=} f(\mathcal{D}) + Lap(0, \frac{\Delta\mathcal{F}}{\epsilon})$, where the $\Delta\mathcal{F}$ indicates the largest possible change between the outputs of the function f which takes as input any neighbor databases $(\mathcal{D}_0, \mathcal{D}_1)$. It is referred to as the L_1-sensitivity which is defined as: $\Delta\mathcal{F} = \underset{(\mathcal{D}_0,\mathcal{D}_1)}{max} ||f(\mathcal{D}_0) - f(\mathcal{D}_1)||_1$.

Sampling from the posterior distribution of a Bayesian model with bounded log-likelihood, recently, has been proven to be differentially private [30]. It is essentially an *exponential mechanism* [15]. Formally, suppose we have a dataset of $\mathcal{L}$ i.i.d examples $\mathcal{X} = \{x_i\}_{i=1}^{\mathcal{L}}$ which we model using a conditional probability distribution $p(x|\theta)$ where θ is a parameter vector, with a prior distribution $p(\theta)$. If $p(x|\theta)$ satisfies $sup_{x\in\mathcal{X},\theta\in\Theta}|\log p(x|\theta)| \leq B$, then releasing one sample from the posterior distribution $p(\theta|\mathcal{X})$ with any prior $p(\theta)$ preserves $4B$-differential privacy. Alternatively, ϵ differential privacy can be preserved by simply rescaling the log-posterior distribution by a factor of $\frac{\epsilon}{4B}$, under the regularity conditions where asymptotic normality (Bernstein-von Mises theorem) holds.

3 Differentially Private SGD

When applying the differential privacy concept, treating the training model (process) as a black box, by only working on the original input or finally output, may result in very poor utility [1,4]. In contrast, by leveraging the tight characterization of training data, NBM and SGD, we directly calibrate noise into the SGD training process, via Laplace mechanism, to differential-privately learn *similarity*. Algorithm 1 outlines our differentially-private SGD method for training probabilistic NBM.

Algorithm 1. Differentially Private SGD

```
Require: Database $R^{>0}$, privacy parameter $\epsilon$, regular parameter $\lambda$, rescale parameter
    $\beta$, learning rate $\eta$, the total number of iterations $K$, initialized similarity $S^{(1)}$.
$S^{(1)} = S^{(1)} \cdot \beta$                                   ▷ rescale the initialization
for $t = 1 : K$ do
    • uniform-randomly sample a mini-batch $\Phi \subset R^{>0}$.
    $\Delta\mathcal{F} = 2e_{max}\frac{\tau}{C}$                  ▷ $e_{max} = 0.5 + \frac{\varphi-1}{t+1}$; $|S_i|I_u \geq C$
    $e_{ui} = min(max(e_{ui}, -e_{max}), e_{max})$                 ▷ $e_{ui} = \hat{r}_{ui} - r_{ui}$
    $\mathcal{G} = \sum_{(u,i)\in\Phi} e_{ui}\frac{\partial \hat{r}_{ui}}{\partial S_i} + Laplace(\frac{\gamma K \Delta\mathcal{F}}{\epsilon})$    ▷ $\gamma = \frac{L}{\mathcal{L}}$
    $S^{(t+1)} \leftarrow S^{(t)} - \eta(\beta\mathcal{G} + \lambda S^{(t)})$    ▷ up-scale the update
end for
return $S^{(t+1)}$
```

According to Eqs. (3) and (4), for each user u (in a randomly chosen mini-batch Φ) the gradient of *similarity* is

$$\mathcal{G}_{ij}(u) = e_{ui}\frac{\partial \hat{r}_{ui}}{\partial S_{ij}} = e_{ui}(\frac{r_{uj}}{S_i I_u^-} - \hat{r}_{ui}\frac{I_{uj}}{S_i I_u^-}) \tag{5}$$

where $e_{ui} = \hat{r}_{ui} - r_{ui}$. For the convenience of notation, we omit $S_{ij} < 0$ part in Eq. (5) which does not compromise the correctness of bound estimation.

To achieve differential privacy, we update the gradient $\mathcal{G}$ by adding Laplace noise (Algorithm 1, line 6). The amount of noise is determined by the bound of gradient $\mathcal{G}_{ij}(u)$ (sensitivity $\Delta\mathcal{F}$) which further depends on e_{ui}, $(r_{uj} - \hat{r}_{ui} I_{uj})$ and $|S_i| I_u^-$. We reduce the sensitivity by exploiting the characteristics of training data, NBM and SGD respectively, by the following tricks.

Preprocessing is often adopted in machine learning for utility reasons. In our case, it can contribute to privacy protection. For example, we only put users who have more than 20 ratings in the training data. It results in a bigger $|S_i| I_u^-$ thus will reduce sensitivity. Suppose the rating scale is $[r_{min}, r_{max}]$, removing "paranoid" records makes $|r_{uj} - \hat{r}_{ui} I_{uj}| \leq \varphi$ hold, where $\varphi = r_{max} - r_{min}$.

Rescaling the value of similarity allows a lower sensitivity. NBM, see Eq. (2), allows us to rescale the *similarity* S to an arbitrarily large magnitude such that we can further reduce the sensitivity (by increasing the value of $|S_i| I_u$). However, the initialization of *similarity* strongly influences the convergence of the training. Thus, it is crucial to balance the convergence (accuracy) and the value of *similarity* (privacy). Another observation is that the gradient downscales when enlarging the *similarity*, see Eq. (5). We can up-scale the gradient monotonically during the training process (Algorithm 1, lines 1 and 7).

The prediction error $e_{ui} = \hat{r}_{ui} - r_{ui}$ decreases when the training goes to convergence such that we can clamp e_{ui} to a lower bound dynamically. In our experiments, we bound the prediction error as $|e_{ui}| \leq 0.5 + \frac{\varphi - 1}{t+1}$, where t is the iteration index. This constraint trivially influences the convergence under non-private training process.

After applying all the tricks, we have the dynamic gradient bound at iteration t as follows: $max(|\mathcal{G}^{(t)}|) \leq (0.5 + \frac{\varphi-1}{t+1})\frac{\varphi}{C}$. The *sensitivity of* each iteration is $\Delta\mathcal{F} = 2max(|\mathcal{G}^{(t)}|) \leq 2(0.5 + \frac{\varphi-1}{t+1})\frac{\varphi}{C}$.

Theorem 1. *Uniform-randomly sample L examples from a dataset of the size $\mathcal{L}$, Algorithm 1 achieves ϵ-differential privacy if in each SGD iteration t we set $\epsilon^{(t)} = \frac{\epsilon}{K\gamma}$ where K is the number of iterations and $\gamma = \frac{L}{\mathcal{L}}$.*

Proof. In Algorithm 1, suppose the number of iterations K is known in advance, and each SGD iteration maintains $\frac{\epsilon}{K\gamma}$-differential privacy. The privacy enhancing technique [3,11] indicates that given a method which is ϵ-differentially private over a deterministic training set, then it maintains $\gamma\epsilon$-differential privacy with respect to a full database if we uniform-randomly sample training set from the database where γ is the sampling ratio. Finally, combining the privacy enhancing technique with composition theory [9], it ensures the K iterations SGD process maintain the overall bound of ϵ-differential privacy. □

4 Differentially Private Posterior Sampling

Sampling from the posterior distribution of a Bayesian model with bounded log-likelihood has free differential privacy to some extent [30]. Specifically, for probabilistic NBM, releasing a sample of the *similarity* S,

$$S \sim p(S|R^{>0}, \alpha_S, \alpha_R) \propto exp(\sum_{i=1}^{M}\sum_{u=1}^{N}(r_{ui} - \frac{S_i R_u^-}{|S_i|I_u^-})^2 + \lambda \sum_{i=1}^{M} ||S_i||_2) \tag{6}$$

achieves $4B$-differential privacy at user level, if each user's log-likelihood is bounded to B, i.e. $\underset{u \in R^{>0}}{max} \sum_{i \in R_u} (\hat{r}_{ui} - r_{ui})^2 \leq B$. Wang et al. [30] showed that we can achieve ϵ-differential privacy by simply rescaling the log-posterior distribution with $\frac{\epsilon}{4B}$, i.e. $\frac{\epsilon}{4B} \cdot \log p(S|R^{>0}, \alpha_S, \alpha_R)$.

Posterior sampling is computationally costly. For the sake of efficiency, we adopt a recent introduced Monte Carlo method, Stochastic Gradient Langevin Dynamics (SGLD) [32], as our MCMC sampler. To successfully use SGLD, we need to derive an unbiased estimator of *similarity* gradient from a mini-batch which is a non-trivial task.

Next, we first overview the basic principles of SGLD (Sect. 4.1), then we derive an unbiased estimator of the true *similarity* gradient (Sect. 4.2), and finally present our privacy-preserving algorithm (Sect. 4.3).

4.1 Stochastic Gradient Langevin Dynamics

SGLD is an annealing of SGD and Langevin dynamics [23] which generates samples from a posterior distribution. Intuitively, it adds an amount of Gaussian noise calibrated by the step sizes (learning rate) used in the SGD process, and the step sizes are allowed to go to zero. When it is far away from the basin of convergence, the update is much larger than noise and it acts as a normal SGD process. The update decreases when the sampling approaches to the convergence basin such that the noise dominated, and it behaves like a Brownian motion. SGLD updates the candidate states according to the following rule.

$$\varDelta\theta_t = \frac{\eta_t}{2}(\varDelta \log p(\theta_t) + \frac{\mathcal{L}}{L}\sum_{i=1}^{L} \varDelta \log p(x_{ti}|\theta_t)) + z_t; \qquad z_t \sim \mathcal{N}(0, \eta_t) \tag{7}$$

where η_t is a sequence of step sizes. $p(x|\theta)$ denotes conditional probability distribution, and θ is a parameter vector with a prior distribution $p(\theta)$. L is the size of a mini-batch randomly sampled from dataset $\mathcal{X}^{\mathcal{L}}$. To ensure convergence to a local optimum, the following requirements of step size η_t have to be satisfied: $\sum_{t=1}^{\infty} \eta_t = \infty$; $\sum_{t=1}^{\infty} \eta_t^2 < \infty$. Decreasing step size η_t reduces the discretization error such that the rejection rate approaches zero, thus we do not need accept-reject test. Following the previous works, e.g. [13,32], we set step size $\eta_t = \eta_1 t^{-\xi}$, commonly, $\xi \in [0.3, 1]$. In order to speed up the burn-in phase of SGLD, we multiply the step size η_t by a temperature parameter ϱ $(0 < \varrho < 1)$ where $\sqrt{\varrho \cdot \eta_t} \gg \eta_t$ [6].

4.2 Unbiased Estimator of The Gradient

The log-posterior distribution of *similarity* S has been defined in Eq. (3). The true gradient of the *similarity* S over $R^{>0}$ can be computed as

$$\mathcal{G}(R^{>0}) = \sum_{(u,i) \in R^{>0}} g_{ui}(S; R^{>0}) + \lambda S \tag{8}$$

where $g_{ui}(S;R^{>0}) = e_{ui}\frac{\partial \hat{r}_{ui}}{\partial S_i}$. To use SGLD and make it converge to true posterior distribution, we need an unbiased estimator of the true gradient which can be computed from a mini-batch $\Phi \subset R^{>0}$. Assume that the size of Φ and $R^{>0}$ are L and $\mathcal{L}$ respectively. The stochastic approximation of the gradient is

$$\mathcal{G}(\Phi) = \mathcal{L}\bar{g}(S,\Phi) + \lambda S \circ \mathbb{I}[i,j \in \Phi] \tag{9}$$

where $\bar{g}(S,\Phi) = \frac{1}{L}\sum_{(u,i)\in\Phi} g_{ui}(S,\Phi)$. $\mathbb{I} \subset \mathbb{B}^{M\times M}$ is symmetric binary matrix, and $\mathbb{I}[i,j \in \Phi] = 1$ if any item-pair (i,j) exists in Φ, otherwise 0. $\circ$ presents element-wise product. The expectation of $\mathcal{G}(\Phi)$ over all mini-batches is,

$$\mathbb{E}_{\Phi}[\mathcal{G}(\Phi)] = \sum_{(u,i)\in R^{>0}} g_{ui}(S;R^{>0}) + \lambda \mathbb{E}_{\Phi}[S \circ \mathbb{I}[i,j \in \Phi]] \tag{10}$$

$\mathbb{E}_{\Phi}[\mathcal{G}(\Phi)]$ is not an unbiased estimator of the true gradient $\mathcal{G}(R^{>0})$ due to the prior term $\mathbb{E}_{\Phi}[S \circ \mathbb{I}[i,j \in \Phi]]$. Let $\mathbb{H} = \mathbb{E}_{\Phi}[\mathbb{I}[i,j \in \Phi]]$, we can remove this bias by multiplying the prior term with $\mathbb{H}^{-1}$ thus to obtain an unbiased estimator. Follow previous approach [2], we assume the mini-batches are sampled with replacement, then $\mathbb{H}$ is $\mathbb{H}_{ij} = 1 - \frac{|I_i||I_j|}{\mathcal{L}^2}(1-\frac{|I_j|}{\mathcal{L}})^{L-1}(1-\frac{|I_i|}{\mathcal{L}})^{L-1}$, where $|I_i|$ (resp. $|I_j|$) denotes the number of ratings of item i (resp. j) in the complete dataset $R^{>0}$. Then the SGLD update rule is the following:

$$S^{(t+1)} \leftarrow S^{(t)} - \frac{\eta_t}{2}(\mathcal{L}\bar{g}(S^{(t)},\Phi) + \lambda S^{(t)} \circ \mathbb{H}^{-1}) + z_t \tag{11}$$

4.3 Differential Privacy via Posterior Sampling

To construct a differentially private NBM, we exploit a recent observation that sampling from scaled posterior distribution of a Bayesian model with bounded log-likelihood can achieve ϵ-differential privacy [30]. We summarize the differentially private sampling process (via SGLD) in Algorithm 2.

Algorithm 2. Differentially Private Posterior Sampling (via SGLD)

```
Require: Temperature parameter $\varrho$, privacy parameter $\epsilon$, regular parameter $\lambda$, initial
    learning rate $\eta_1$. Let $K$ larger than burn-in phase.
for $t = 1 : K$ do
    • Randomly sample a mini-batch $\Phi \subset R^{>0}$.
    $\bar{g}(S^{(t)},\Phi) = \frac{1}{L}\sum_{(u,i)\in\Phi} e_{ui}\frac{\partial \hat{r}_{ui}}{\partial S_i^{(t)}}$          ▷ gradient of $S$ (mini-batch)
    $z_t \sim \mathcal{N}(0, \varrho \cdot \eta_t)$          ▷ $\sqrt{\varrho \cdot \eta_t} \gg \eta_t$
    $S^{(t+1)} \leftarrow S^{(t)} - \frac{\epsilon}{4B} \cdot \frac{\eta_t}{2}(\mathcal{L}\bar{g}(S^{(t)},\Phi) + \lambda S^{(t)} \circ \mathbb{H}^{-1}) + z_t$
    $\eta_{t+1} = \frac{\eta_1}{t^{\gamma}}$
end for
return $S^{(t+1)}$
```

Now, a natural question is how to determine the log-likelihood bound B? ($\underset{u\in R^{>0}}{max}\sum_{i\in R_u}(\hat{r}_{ui} - r_{ui})^2 \leq B$, and see Eq. (6)). Obviously, B depends on the max rating number per user. To those users who rated more than τ items, we randomly remove some ratings thus to ensure that each user at most has τ ratings. In our context, the rating scale is [1,5], let $\tau = 200$, we have $B = (5-1)^2 \times 200$ (In reality, most users have less than 200 ratings [13]).

Theorem 2. *Algorithm 2 provides $(\epsilon, (1+e^{\epsilon})\delta)$-differential privacy guarantee to any user if the distribution $P'_{\mathcal{X}}$ where the approximate samples from is δ-far away from the true posterior distribution $P_{\mathcal{X}}$, formally $||P'_{\mathcal{X}} - P_{\mathcal{X}}||_1 \leq \delta$. And $\delta \to 0$ if the MCMC sampling asymptotically converges.*

Proof. Essentially, differential privacy via posterior sampling [30] is an exponential mechanism [15] which protects ϵ-differential privacy when releasing a sample θ with probability proportional to $exp(-\frac{\epsilon}{2\Delta\mathcal{F}}p(\mathcal{X}|\theta))$, where $p(\mathcal{X}|\theta)$ serves as the utility function. If $p(\mathcal{X}|\theta)$ is bounded to B, we have the sensitivity $\Delta\mathcal{F} \leq 2B$. Thus, release a sample by Algorithm 2 preserves ϵ-differential privacy. It compromises the privacy guarantee to $(\epsilon, (1+e^{\epsilon})\delta)$ if the distribution (where the sample from) is δ-far away from the true posterior distribution, proved by [30]. □

Note that when $\epsilon = 4B$, the differentially private sampling process is identical to the non-private sampling. This is also the meaning of *some extent of free privacy*. It starts to lose accuracy when $\epsilon < 4B$. One concern of this sampling approach is the distance δ between the distribution where the samples from and the true posterior distribution, which compromises the differential privacy guarantee. Fortunately, [24,28] proved that SGLD can converge in finite iterations. As such we can have arbitrarily small δ with a (large) number of iterations.

5 Experiments and Evaluation

We test our solutions on two real world datasets, ML100K and ML1M [17], which are widely employed for evaluating recommender systems. ML100K dataset has 100 K ratings that 943 users assigned to 1682 movies. ML1M dataset contains 1 million ratings that 6040 users gave to 3952 movies. In the experiments, we adopt 5-fold cross validation for training and evaluation. We use root mean square error (RMSE) to measure accuracy performance: $RMSE = \sqrt{\frac{\sum_{(u,i)\in R^T}(r_{ui}-\hat{r}_{ui})^2}{|R^T|}}$, where $|R^T|$ is the total number of ratings in the test set R^T.

5.1 Experiments Setup

In the following, the differentially-private SGD based PNBM is referred to as DPSGD-PNBM, and the differentially-private posterior sampling PNBM is referred as DPPS-PNBM. The experiment source code is available at Github[1]. We compare their performances with the following baseline algorithms.

[1] https://github.com/lux-jwang/Experiments/tree/master/dpnbm.

- *non-private PCC and COS:* Differentially-private Pearson correlation (PCC) or Cosine similarity (COS) NBMs exist (e.g. [10,14,33]), with worse accuracy than the non-private algorithms. We directly use the non-private ones.
- *DPSGD-MF:* Differentially private matrix factorization from [4], which calibrates Laplacian noise into the SGD training process.
- *DPPS-MF:* Differentially private matrix factorization from [13], which exploits the posterior sampling technique.

We optimize model parameters using a heuristic grid search method, as follows.

- *DPSGD-PNBM:* The learning rate η is searched in $\{0.1, 0.4\}$, and the iteration number $K \in [1, 20]$, the regular parameter $\lambda \in \{0.05, 0.005\}$, the rescale parameter $\beta \in \{10, 20\}$. The neighbor size $N_k = 500$, the lower bound of $|S_i|I_u$: $C \in \{10, 15\}$. In the training process, we decrease K and increase $\{\eta, C\}$ when requiring a stronger privacy guarantee (a smaller ϵ).
- *DPPS-PNBM:* The initial learning rate $\eta_1 \in \{8 \cdot 10^{-8}, 4 \cdot 10^{-7}, 8 \cdot 10^{-6}\}$, $\lambda \in \{0.02, 0.002\}$, the temperature parameter $\varrho = \{0.001, 0.006, 0.09\}$, the decay parameter $\xi = 0.3$. $N_k = 500$.
- *DPSGD-MF:* $\eta \in \{6{\cdot}10^{-4}, 8{\cdot}10^{-4}\}$, $K \in [10, 50]$ (the smaller privacy loss ϵ the less iterations), $\lambda \in \{0.2, 0.02\}$, the latent feature dimension $d \in \{10, 15, 20\}$.
- *DPPS-MF:* $\eta \in \{2 \cdot 10^{-9}, 2 \cdot 10^{-8}, 8 \cdot 10^{-7}, 8 \cdot 10^{-6}\}$, $\lambda \in \{0.02, 0.05, 0.1, 0.2\}$, $\varrho = \{1 \cdot 10^{-4}, 6 \cdot 10^{-4}, 4 \cdot 10^{-3}, 3 \cdot 10^{-2}\}$, $d \in \{10, 15, 20\}$,$\xi = 0.3$.
- *non-private PCC and COS:* For ML100K, we set $N_K = 900$. For ML1M, we set $N_K = 1300$.

5.2 Comparison Results

We first compare the accuracy between DPSGD-PNBM, DPSGD-MF, non-private PCC and COS and show the results in Fig. 1 for the two datasets respectively. When $\epsilon \geq 20$, DPSGD-MF does not lose much accuracy, and it is better than non-private PCC and COS. However, the accuracy drops quickly (or, the

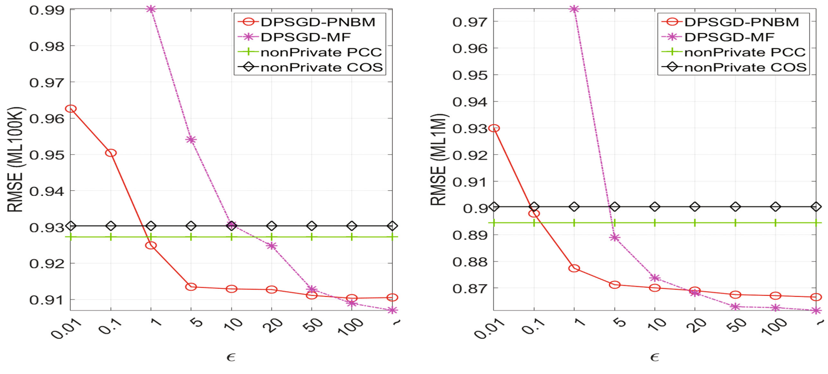

Fig. 1. Accuracy comparison: DPSGD-PNBM, DPSGD-MF, non-private PCC, COS.

RMSE increase quickly) when the privacy loss ϵ is reduced. This matches the observation in [4]. In the contrast, DPSGD-PNBM maintains a promising accuracy when $\epsilon \geq 1$, and is better than non-private PCC and COS.

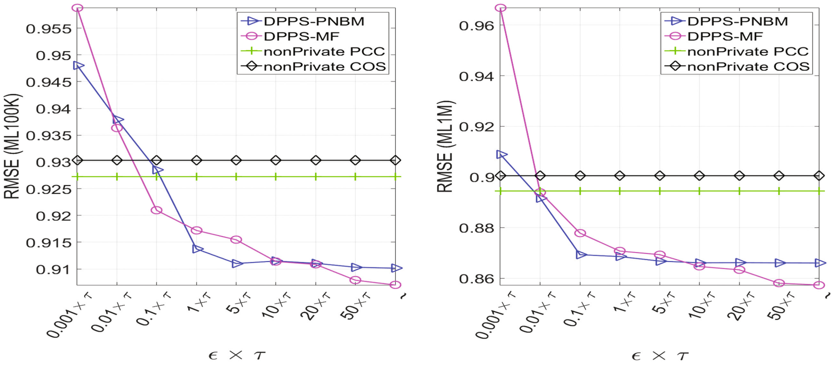

Fig. 2. Accuracy comparison: DPPS-PNBM, DPPS-MF, non-private PCC, COS.

DPPS-PNBM and DPPS-MF preserve differential privacy at user level. We denote the privacy loss ϵ in form of $x \times \tau$ where x is a float value which indicates the average privacy loss at a rating level, and τ is the max rate number per user. The comparison is shown in Fig. 2. In our context, for both datasets, $\tau = 200$. Both DPPS-PNBM and DPPS-MF allow accurate estimations when $\epsilon \geq 0.1 \times 200$. It may seem that $\epsilon = 20$ is a meaningless privacy guarantee. We remark that the average privacy of a rating level is 0.1. Besides the accuracy performance is better than the non-private PCC and COS, from the point of privacy loss ratio, our models match previous works [13,14], where it is showed that differentially private systems may not lose much accuracy when $\epsilon > 1$.

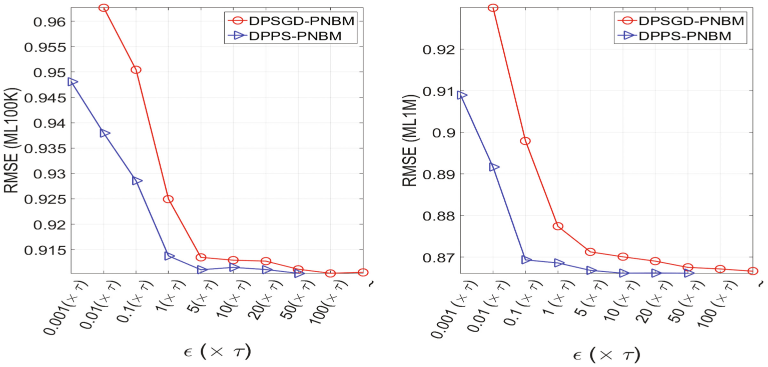

Fig. 3. Accuracy comparison between DPSGD-PNBM and DPPS-PNBM

DPSGD-PNBM and DPPS-PNBM achieve differential privacy at rating level (a single rating) and user level (a whole user profile) respectively. Below, we try to compare them at rating level, precisely at the average rating level for DPPS-PNBM. Figure 3 shows that both solutions can obtain quite accurate predictions with a privacy guarantee ($\epsilon \approx 1$). With the same privacy guarantee, DPPS-PNBM seems to be more accurate. However, DPPS-PNBM has its potential drawback. Recall from Sect. 4, the difference δ between the distribution where samples from and the true posterior distribution compromises differential privacy guarantee. In order to have an arbitrarily small δ, DPPS-PNBM requires a large number of iterations [24,28]. At this point, it is less efficient than DPSGD-PNBM. In our comparison, we assume $\delta \to 0$.

5.3 Summary

In summary, DPSGD-MF and DPPS-MF are more accurate when privacy loss is large. DPSGD-PNBM and DPPS-PNBM are better when we want to reduce the privacy loss to a meaningful range. Both our models consistently outperform non-private traditional NBMs, with a meaningful differential privacy guarantee. Note that *similarity* is independent of NBM itself, thus other neighborhood-based recommenders can use our models to differential-privately learn *Similarity*, and deploy it to their existing systems without requiring extra effort.

6 Related Work

A number of works have demonstrated that an attacker can infer the user sensitive information, such as gender and politic view, from public recommendation results without using much background knowledge, e.g. [5,31]. Randomized data perturbation is one of earliest approaches to prevent user data from inference attack in which people either add random noise to their profiles or substitute some randomly chosen ratings with real ones, e.g. [20,21]. While this approach is very simple, it does not offer rigorous privacy guarantee. Differential privacy [9] aims to precisely protect user privacy in statistical databases, and the concept has become very popular recently. [14] is the first work to apply differential privacy to recommender systems, and it has considered both neighborhood-based methods (using correlation as *similarity*) and latent factor model (e.g. SVD). [33] introduced a differentially private neighbor selection scheme by injecting Laplace noise to the *similarity* matrix. [10] presented a scheme to obfuscate user profiles that preserves differential privacy. [4,13] applied differential privacy to matrix factorization, and we have compared our solutions to theirs in Sect. 5. Secure multiparty computation recommender systems allow users to compute recommendation results without revealing their inputs to other parties. Many protocols have been proposed, e.g. [19,26]. Unfortunately, these protocols do not prevent information leakage from the recommendation results.

7 Conclusion

In this paper, we have proposed two different differentially private NBMs, under a probabilistic framework. We firstly introduced a way to differential-privately find the maximum a posteriori *similarity* by calibrating noise to the SGD training process. Then we built differentially private NBM by exploiting the fact that sampling from scaled posterior distribution can result in differentially private systems. While the experiment results have demonstrated that our models allow promising accuracy with a modest privacy budget in some well-known datasets, we consider it as an interesting future work to test the performances in other real world datasets.

Acknowledgments. Both authors are supported by a CORE (junior track) grant from the National Research Fund, Luxembourg.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM (2016)
2. Ahn, S., Korattikara, A., Liu, N., Rajan, S., Welling, M.: Large-scale distributed bayesian matrix factorization using stochastic gradient MCMC. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 9–18. ACM (2015)
3. Beimel, A., Brenner, H., Kasiviswanathan, S.P., Nissim, K.: Bounds on the sample complexity for private learning and private data release. Mach. Learn. **94**(3), 401–437 (2014)
4. Berlioz, A., Friedman, A., Kaafar, M.A., Boreli, R., Berkovsky, S.: Applying differential privacy to matrix factorization. In: Proceedings of the 9th ACM Conference on Recommender Systems, pp. 107–114. ACM (2015)
5. Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W., Shmatikov, V.: "You might also like:" privacy risks of collaborative filtering. In: 2011 IEEE Symposium on Security and Privacy, pp. 231–246. IEEE (2011)
6. Chen, T., Fox, E.B., Guestrin, C.: Stochastic gradient Hamiltonian Monte Carlo. In: ICML, pp. 1683–1691 (2014)
7. Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 107–144. Springer, Heidelberg (2011)
8. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). doi:10.1007/11681878_14
9. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
10. Guerraoui, R., Kermarrec, A.-M., Patra, R., Taziki, M.: D2P: distance-based differential privacy in recommenders. Proc. VLDB Endow. **8**(8), 862–873 (2015)
11. Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.: What can we learn privately? SIAM J. Comput. **40**(3), 793–826 (2011)

12. Koren, Y., Bell, R., Volinsky, C., et al.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
13. Liu, Z., Wang, Y.-X., Smola, A.: Fast differentially private matrix factorization. In: Proceedings of the 9th ACM Conference on Recommender Systems. ACM (2015)
14. McSherry, F., Mironov, I.: 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. ACM (2009)
15. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), pp. 94–103. IEEE (2007)
16. Mobasher, B., Burke, R., Bhaumik, R., Williams, C.: Toward trustworthy recommender systems: an analysis of attack models and algorithm robustness. ACM Trans. Internet Technol. (TOIT) **7**(4), 23 (2007)
17. MovieLens. MovieLens Datasets. http://grouplens.org/datasets/movielens/
18. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: 2008 IEEE Symposium on Security and Privacy. IEEE (2008)
19. Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., Boneh, D.: Privacy-preserving matrix factorization. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 801–812. ACM (2013)
20. Polat, H., Du, W.: Privacy-preserving collaborative filtering using randomized perturbation techniques. In: Third IEEE International Conference on Data Mining (ICDM 2003), pp. 625–628. IEEE (2003)
21. Polat, H., Du, W.: Achieving private recommendations using randomized response techniques. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 637–646. Springer, Heidelberg (2006). doi:10.1007/11731139_73
22. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press (2009)
23. Rossky, P., Doll, J., Friedman, H.: Brownian dynamics as smart monte carlo simulation. J. Chem. Phys. **69**(10), 4628–4633 (1978)
24. Sato, I., Nakagawa, H.: Approximation analysis of stochastic gradient langevin dynamics by using Fokker-Planck equation and Ito process. In: ICML (2014)
25. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**, 4 (2009)
26. Tang, Q., Wang, J.: Privacy-preserving context-aware recommender systems: analysis and new solutions. In: Pernul, G., Ryan, P.Y.A., Weippl, E. (eds.) ESORICS 2015. LNCS, vol. 9327, pp. 101–119. Springer, Cham (2015). doi:10.1007/978-3-319-24177-7_6
27. Töscher, A., Jahrer, M., Legenstein, R.: Improved neighborhood-based algorithms for large-scale recommender systems. In: Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems, p. 4. ACM (2008)
28. Vollmer, S.J., Zygalakis, K.C., et al.: (Non-) asymptotic properties of stochastic gradient langevin dynamics (2015). arXiv preprint arXiv:1501.00438
29. Wang, J., Tang, Q.: A probabilistic view of neighborhood-based recommendation methods (2016). https://arxiv.org/abs/1701.01250
30. Wang, Y.-X., Fienberg, S.E., Smola, A.: Privacy for free: posterior sampling and stochastic gradient monte carlo (2015)
31. Weinsberg, U., Bhagat, S., Ioannidis, S., Taft, N.: Blurme: inferring and obfuscating user gender based on ratings. In: Proceedings of the Sixth ACM Conference on Recommender Systems, pp. 195–202. ACM (2012)

32. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient langevin dynamics. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 681–688 (2011)
33. Zhu, T., Ren, Y., Zhou, W., Rong, J., Xiong, P.: An effective privacy preserving algorithm for neighborhood-based collaborative filtering. Future Gener. Comput. Syst. **36**, 142–155 (2014)