Private Alternating Least Squares: Practical Private Matrix Completion with Tighter Rates

Steve Chien[1] **Prateek Jain**[1] **Walid Krichene**[*1] **Steffen Rendle**[1]
Shuang Song[1] **Abhradeep Thakurta**[*1] **Li Zhang**[1]

Abstract

We study the problem of differentially private (DP) matrix completion under user-level privacy. We design a joint differentially private variant of the popular Alternating-Least-Squares (ALS) method that achieves: i) (nearly) optimal sample complexity for matrix completion (in terms of number of items, users), and ii) the best known privacy/utility trade-off both theoretically, as well as on benchmark data sets. In particular, we provide the first global convergence analysis of ALS with *noise* introduced to ensure DP, and show that, in comparison to the best known alternative (the Private Frank-Wolfe algorithm by Jain et al. (2018)), our error bounds scale significantly better with respect to the number of items and users, which is critical in practical problems. Extensive validation on standard benchmarks demonstrate that the algorithm, in combination with carefully designed sampling procedures, is significantly more accurate than existing techniques, thus promising to be the first practical DP embedding model.

1. Introduction

Given $\boldsymbol{M}_{ij}, (i,j) \in \Omega$ where $\Omega \subseteq [n] \times [m]$ is a set of observed user-item ratings, and assuming $\boldsymbol{M} \approx \boldsymbol{U}^*(\boldsymbol{V}^*)^\top \in \mathbb{R}^{n\times m}$ to be a nearly low-rank matrix, the goal of low-rank matrix completion (LRMC) is to efficiently learn $\widehat{\boldsymbol{U}} \in \mathbb{R}^{n\times r}$ and $\widehat{\boldsymbol{V}} \in \mathbb{R}^{m\times r}$, such that $\boldsymbol{M} \approx \widehat{\boldsymbol{U}}\widehat{\boldsymbol{V}}^\top$.

LRMC, a.k.a. matrix factorization, is a cornerstone technique for building recommendation systems (Koren & Bell, 2015; Hu et al., 2008), and though proposed over a decade ago, it remains highly competitive (Rendle et al., 2019). In the recommendation setting, $\boldsymbol{M}$ represents a mostly unknown user-item ratings matrix and $\widehat{\boldsymbol{U}}$ and $\widehat{\boldsymbol{V}}$ capture the user and item embeddings. Using the learned ($\widehat{\boldsymbol{U}}$, $\widehat{\boldsymbol{V}}$), the system computes rating predictions $\widehat{\boldsymbol{M}}_{ij} = (\widehat{\boldsymbol{U}}\widehat{\boldsymbol{V}}^\top)_{ij}$ to recommend items for the users. To ensure good generalization, one would set the rank $r \ll \min(m, n)$.

Such models, while highly successful in practice, have the risk of leaking users' ratings through model parameters or their recommendations. The privacy risk of similar models has been well documented, and the protection against it has been intensively studied (Dinur & Nissim, 2003; Dwork et al., 2007; Korolova, 2010; Calandrino et al., 2011; Shokri et al., 2017; Carlini et al., 2019; 2020a;b; Thakkar et al., 2020). In this paper, we focus on learning user and item embeddings, and consequently user-item recommendations, while ensuring privacy of users' ratings.

We conform to the well-established formal notion of differential privacy (DP) (Dwork et al., 2006a;b) to protect users' ratings. We operate in the setting of *user-level* privacy (Dwork & Roth, 2014; Jain et al., 2018), where we intend to protect *all the ratings by the user*, a much harder task than protecting a single rating from the user (a.k.a. *entry-level privacy*) (Hardt & Roth, 2013; Meng et al., 2018). Note that *user-level* privacy is critical in this problem, as the ratings from a single user tend to be correlated and can thus be used to fingerprint a user (Calandrino et al., 2011). As is standard in the user-level privacy literature (Jain et al., 2018), we estimate the shared item embeddings $\widehat{\boldsymbol{V}}$ while preserving privacy with respect to the users. In contrast, each user *independently* computes their embedding (a row of $\widehat{\boldsymbol{U}}$) as a function of their own ratings and the privacy preserving item embeddings $\widehat{\boldsymbol{V}}$. Formally, this setup is called *joint differential privacy* (Kearns et al., 2014), and it is well-established (Hardt & Roth, 2012; 2013) that such a relaxation is necessary to learn non-trivial recommendations while ensuring user-level privacy.

While several works have studied LRMC under joint-differential privacy (McSherry & Mironov, 2009; Liu et al., 2015; Jain et al., 2018), most of the existing techniques do not provide satisfactory empirical performance compared to the state-of-the-art (SOTA) non-private LRMC methods. Furthermore, these works either lack a rigorous performance

[1] Google Research. Correspondence to: Walid Krichene <walidk@google.com>, Abhradeep Thakurta <athakurta@google.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

analysis (McSherry & Mironov, 2009; Liu et al., 2015) or provide guarantees that are significantly weaker (Jain et al., 2018) than that of non-private LRMC algorithms. Matrix factorization can also be solved using other first-order methods such as stochastic gradient descent (Ge et al., 2016) or alternating gradient descent (Lu et al., 2019), so one may apply the differentially private SGD (DPSGD) algorithm (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016) to achieve privacy. However, applying DPSGD to LRMC is challenging as SGD typically requires many steps to converge, thus increasing privacy cost.

In this work, we design and analyze a differentially private version of the widely used alternating least squares (ALS) algorithm for LRMC (Koren et al., 2009; Jain et al., 2013). ALS alternates between optimizing over the user embeddings $\widehat{\boldsymbol{U}}$ and the item embeddings $\widehat{\boldsymbol{V}}$, each through least squares minimization. One important property of ALS is that when solving for one side, the optimization can be done independently for each user or item, which makes ALS highly scalable. Our key insight is that this decoupling of the solution is also useful for privacy-preserving computation, since there is no accumulation of noise when solving for the embeddings of different users (or items). Besides, ALS is known to require few iterations to converge in practice, making it particularly suitable for privacy preserving LRMC.

Indeed, we present a differentially private variant of ALS, which we refer to as DPALS, and demonstrate that it enjoys much tighter error rates (see Table 1) and better empirical performance than the current SOTA, the differentially private Frank-Wolfe (DPFW) method of Jain et al. (2018). Furthermore, on the large scale benchmark of MovieLens 20M, DPALS produces the first realistic DP embedding model with competitive recall metric under moderate privacy loss.

More specifically, our contributions are the following.

Private alternating least squares for matrix completion. We provide the first differentially private version of alternating least squares (DPALS) for matrix completion with user-level privacy guarantee (Section 3). The algorithm is conceptually simple, efficient, and highly scalable. We provide rigorous analysis on its privacy guarantee under the notion of Joint Rényi Differential Privacy.

Tighter privacy/utility/computation trade-offs. We prove theoretical guarantees on the sample complexity and the error bounds of DPALS under standard assumptions (Section 4). These bounds are much tighter than the current SOTA, the DPFW method (Jain et al., 2018). In particular, we show the following. First, DPALS requires only $O(\log^{O(1)} n)$ samples per user to guarantee its convergence. In contrast, DPFW requires $\sqrt{m}$ ratings per user. Second, to achieve a Frobenius norm error of ζ, DPALS requires $n = \widetilde{\Omega}\left(\frac{m\sqrt{m}}{\zeta\varepsilon} + m\right)$ users, which is nearly optimal in terms of ζ and ε. In contrast, DPFW's sample complexity is $n = \widetilde{\Omega}\left(m^{5/4}/(\zeta^5\varepsilon)\right)$; note a significant improvement in terms of ζ. Finally, Private SVD (McSherry & Mironov, 2009) is not even *consistent*, i.e., for a fixed $\varepsilon, m, |\Omega| = n\sqrt{m}$, even if we scale $n \to \infty$, the Frobenius norm error bound does not converge to 0 (see Theorem B.3 of Jain et al. (2018)).

Table 1. Sample complexity bounds for various algorithms, assuming constant Frobenius norm error. Here, n is the number of users, m is the number of items, and $\widetilde{\Omega}(\cdot)$ hides $\mathsf{polylog}(n, m, 1/\delta)$. (*) assumes additional property of $\boldsymbol{M}$ being incoherent. Private ALS requires initialization specified in Theorem 2. See Remark 2 for an initialization scheme; Private ALS with such an initialization preserves the sample complexity but requires $n = \widetilde{\Omega}(m\sqrt{m})$.

Algorithm	**Bound on n**	**Bound on $\|\Omega\|/n$**	**Iterations**
Trace Norm (*) (non-priv.) (Candès & Recht, 2009)	$\widetilde{\Omega}(m)$	$\widetilde{\Omega}(\log^2 n)$	$\mathsf{poly}(n, m)$
ALS (*) (non-priv.) (Jain et al., 2013)	$\widetilde{\Omega}(m)$	$\widetilde{\Omega}(\log^2 n)$	$\mathsf{polylog}(n, m)$
Private SVD(*) (McSherry & Mironov, 2009)	-	-	-
Private SGLD (Liu et al., 2015)	-	-	-
Private FW (Jain et al., 2018)	$\widetilde{\Omega}(m^{5/4})$	$\widetilde{\Omega}(\sqrt{m})$	$\mathsf{poly}(n, m)$
Private ALS (*) (this work)	$\widetilde{\Omega}(m)$	$\widetilde{\Omega}(\log^3 n)$	$\mathsf{polylog}(n, m)$

Practical techniques to improve accuracy. One main difficulty in applying DPALS to practical problems comes from a heavy skew in the item distribution. We propose two heuristics to reduce the skew while preserving privacy (Section 5). Experiments on real-world benchmarks show that these techniques can significantly improve model quality.

Strong empirical results using DPALS. We carry out an extensive study of DPALS both on synthetic and real-world benchmarks. Aided by the aforementioned practical techniques, DPALS achieves significant gains over the current SOTA method. In particular, on the MovieLens 10M rating prediction benchmark, DPALS achieves the same error rate as the current SOTA even when trained on a small fraction (23%) of users. When trained on the full set of users, it achieves a relative decrease in RMSE of at least 7%. DPALS also achieves a remarkably good performance on the MovieLens 20M item recommendation benchmark with modest privacy loss, and remains competitive even with non-private ALS, the first DP private embedding model to achieve such strong results.

2. Background

2.1. Notation

Let $[m]$ denote the set $\{1, 2, \cdots, m\}$. Let $\mathbb{R}^{n\times m}$ denote the set of $n \times m$ matrices. Throughout the paper, we use bold face uppercase letters to represent matrices and lowercase letters for vectors. For any matrix $\boldsymbol{A} = (\boldsymbol{A}_{ij}) \in \mathbb{R}^{n\times m}$, let $\boldsymbol{A}_i$ be the i-th *row* vector of $\boldsymbol{A}$. Denote by $\|\boldsymbol{A}\|_F, \|\boldsymbol{A}\|_\infty$ the Frobenius norm and the max norm of $\boldsymbol{A}$. For $\Omega \subseteq [n] \times$

$[m]$, define the projection $\mathsf{P}_\Omega(\boldsymbol{A}) \in \mathbb{R}^{n\times m}$ as $\mathsf{P}_\Omega(\boldsymbol{A})_{ij} = \boldsymbol{A}_{ij}$ if $(i,j) \in \Omega$ and 0 otherwise. For $i \in [n]$, define $\Omega_i := \{j : (i,j) \in \Omega\}$. Similarly, for $j \in [m]$, let $\Omega_j = \{i : (i,j) \in \Omega\}$. For $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^r$, we use $\boldsymbol{u} \cdot \boldsymbol{v} \in \mathbb{R}$ to denote their dot product, and $\boldsymbol{u} \otimes \boldsymbol{v} \in \mathbb{R}^{r\times r}$ for their outer product.

2.2. Matrix Completion, Alternating Least Squares

Let $\boldsymbol{M} \in \mathbb{R}^{n\times m}$ be a rank r matrix, such that each entry $\boldsymbol{M}_{ij}$ ($i \in [n], j \in [m]$) represents the preference/affinity of user i for item j. Given a set of observed entries $\mathsf{P}_\Omega(\boldsymbol{M})$, $\Omega \subseteq [n] \times [m]$, the goal of LRMC is to reconstruct $\boldsymbol{M}$ with minimal error. This can be achieved by finding $\widehat{\boldsymbol{U}} \in \mathbb{R}^{n\times r}$ and $\widehat{\boldsymbol{V}} \in \mathbb{R}^{m\times r}$ such that the regularized squared error $\|\mathsf{P}_\Omega(\boldsymbol{M} - \widehat{\boldsymbol{U}}\widehat{\boldsymbol{V}}^\top)\|_F^2 + \lambda\|\widehat{\boldsymbol{U}}\|_F^2 + \lambda\|\widehat{\boldsymbol{V}}\|_F^2$ is minimized. This minimization problem is NP-hard in general (Hardt et al., 2014). But the alternating least squares (ALS) algorithm has proved to work well in practice.

ALS alternatingly computes $\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{V}}$ by minimizing the above objective while assuming the other embeddings fixed. Each alternating step can be solved efficiently through the standard least squares algorithm with the following closed form solution.

$$\forall i \quad \widehat{\boldsymbol{U}}_i^t = (\lambda \boldsymbol{I} + \sum_{j\in\Omega_i} \widehat{\boldsymbol{V}}_j^t \otimes \widehat{\boldsymbol{V}}_j^t)^{-1} \sum_{j\in\Omega_i} \boldsymbol{M}_{ij}\widehat{\boldsymbol{V}}_j^t, \tag{1}$$

$$\forall j \quad \widehat{\boldsymbol{V}}_j^{t+1} = (\lambda \boldsymbol{I} + \sum_{i\in\Omega_j} \widehat{\boldsymbol{U}}_i^t \otimes \widehat{\boldsymbol{U}}_i^t)^{-1} \sum_{i\in\Omega_j} \boldsymbol{M}_{ij}\widehat{\boldsymbol{U}}_i^t. \tag{2}$$

While ALS does not guarantee convergence to the global optimum in general, it works remarkably well in practice and often produces $\widehat{\boldsymbol{U}}$ and $\widehat{\boldsymbol{V}}$ such that $\widehat{\boldsymbol{U}}\widehat{\boldsymbol{V}}^\top$ is a good approximation of $\boldsymbol{M}$. The practical success of ALS has inspired many theoretical analyses, which make the following additional assumptions on $\boldsymbol{M}$ and Ω.

Assumption 1 (μ-incoherence)**.** *Let $\boldsymbol{M} = \boldsymbol{U}^*\boldsymbol{\Sigma}^*(\boldsymbol{V}^*)^\top$ be the singular value decomposition of $\boldsymbol{M}$, i.e. $\boldsymbol{U}^* \in \mathbb{R}^{n\times r}, \boldsymbol{V}^* \in \mathbb{R}^{m\times r}$ are orthonormal matrices, and $\boldsymbol{\Sigma}^* \in \mathbb{R}^{r\times r}$ is the diagonal matrix of the singular values of $\boldsymbol{M}$. We assume that $\boldsymbol{M}$ is μ-incoherent, that is, $\forall i \in [n]$, $\|\boldsymbol{U}_i^*\|_2 \le \frac{\mu\sqrt{r}}{\sqrt{n}}$; and $\forall j \in [m]$, $\left\|\boldsymbol{V}_j^*\right\|_2 \le \frac{\mu\sqrt{r}}{\sqrt{m}}$.*

Assumption 2 (Random Ω)**.** *We assume that Ω are random observations with probability p, that is, $\Omega = \{(i,j) \in [n] \times [m] : \delta_{ij} = 1\}$, where $\delta_{ij} \in \{0,1\}$ are i.i.d. random variables with $\mathbf{Pr}[\delta_{ij} = 1] = p$.*

Jain et al. (2013); Hardt & Wootters (2014) showed that ALS converges to $\boldsymbol{M}$ with high probability if $\boldsymbol{M}$ is μ-incoherent and $p = \widetilde{\Omega}\left(\frac{\log n}{m}\right)$, where $n \ge m$ and $\widetilde{\Omega}$ hides polynomial dependence on μ, r, $\kappa := \sigma_1^*/\sigma_r^*$, with σ_1^*, σ_r^* being the maximum and minimum singular value of $\boldsymbol{M}$. In this work, we make the same assumptions on $\boldsymbol{M}$ and Ω. Our key theoretical contribution is a similar convergence result for DPALS, under the additional requirements of user-level differential privacy.

2.3. Joint Differential Privacy

Differential privacy (Dwork et al., 2006b;a) is a widely adopted privacy notion. We use the variant of *user-level* joint differential privacy (Joint DP). Intuitively, Joint DP requires any information which may cross different users to be differentially private, but allows each individual user to use her own private information to her full advantage, for example, when computing the embeddings for generating recommendations to herself. This notion was already implicit in (McSherry & Mironov, 2009) and made formal in (Kearns et al., 2014; Jain et al., 2018).

Let $D = \{d_1, \ldots, d_n\}$ be a data set of n records, where each sample d_i is drawn from a domain τ and belongs to individual i (which we also refer to as a *user*). Let $\mathcal{A} : \tau^* \to \mathcal{S}^n$ be an algorithm that produces n outputs in some space $\mathcal{S}$, one for each user i. Let D_{-i} be the data set with the i-th user removed, and let $\mathcal{A}_{-i}(D)$ be the set of outputs without that of the i-th user. Also, let $(d_i; D_{-i})$ be the data set obtained by adding d_i (for user i) to the data set D_{-i}. Joint DP and its Rényi differential privacy (Mironov, 2017) (Joint RDP) variant are defined as follows.

Definition 3 (Joint Differential Privacy (Kearns et al., 2014))**.** *An Algorithm $\mathcal{A}$ is (ε, δ)-jointly differentially private if for any user i, for any possible value of data entry $d_i, d_i' \in \tau$, for any instantiation of the data set for other users $D_{-i} \in \tau^{n-1}$, and for any set of outputs $S \subseteq \mathcal{S}^n$, the following two inequalities hold simultaneously:*

$$\mathop{\mathbf{Pr}}_{\mathcal{A}}\left[\mathcal{A}_{-i}((d_i; D_{-i})) \in S\right] \le e^\varepsilon \mathop{\mathbf{Pr}}_{\mathcal{A}}[\mathcal{A}_{-i}(D_{-i}) \in S] + \delta$$
$$\mathop{\mathbf{Pr}}_{\mathcal{A}}\left[\mathcal{A}_{-i}(D_{-i}) \in S\right] \le e^\varepsilon \mathop{\mathbf{Pr}}_{\mathcal{A}}[\mathcal{A}_{-i}((d_i; D_{-i})) \in S] + \delta.$$

An algorithm $\mathcal{A}$ is (α, ε)-joint Rényi differentially private (Joint RDP) if $D_\alpha\left(\mathcal{A}_{-i}((d_i; D_{-i}))||\mathcal{A}_{-i}(D_{-i})\right) \le \varepsilon$ and $D_\alpha\left(\mathcal{A}_{-i}(D_{-i})||\mathcal{A}_{-i}((d_i; D_{-i}))\right) \le \varepsilon$, where D_α is the Rényi divergence of order α.

If we replace $\mathcal{A}_{-i}$ with $\mathcal{A}$ in the definition, we would recover the standard definition of DP and RDP. We note that the joint DP (resp. joint RDP) enjoys the same composability properties as the notion of DP (resp. RDP).

3. DPALS: Private Alternating Least Squares

We now provide the details of the DPALS algorithm and prove its privacy guarantee in the joint DP model.

Notation. Let $\mathcal{N}(0, \sigma^2)$ be the Gaussian distribution of variance σ^2, and $\mathcal{N}_{\mathsf{sym}}(0, \sigma^2)^{r\times r}$ be the distribution of symmetric matrices where each entry in the upper triangle is drawn i.i.d. from $\mathcal{N}(0, \sigma^2)$. For a symmetric $\boldsymbol{A}$, let $\Pi_{\mathsf{PSD}}(\boldsymbol{A})$ be its projection to the positive semi-definite cone, obtained by replacing its negative eigenvalues with 0. Define

```
Algorithm 1 DPALS: Private Matrix Completion via Alternating Minimization
Required: Observed ratings: $\mathsf{P}_\Omega(\boldsymbol{M})$, $\sigma$: noise standard deviation, $\Gamma_{\boldsymbol{u}}$: row clipping parameter, $\Gamma_{\boldsymbol{M}}$: entry clipping parameter, $T$: number of steps, $\lambda$: regularization parameter, $r$: rank, $k$: maximum number of ratings per user in $\mathcal{A}_{\mathsf{item}}$, $\widehat{\boldsymbol{V}}^0$: initial $\boldsymbol{V}$.
1  Clip entries in $\mathsf{P}_\Omega(\boldsymbol{M})$ so that $\|\mathsf{P}_\Omega(M)\|_\infty \le \Gamma_{\boldsymbol{M}}$
   for $0 \le t \le T$ do
       for $1 \le i \le n$ do
2          $\widehat{\boldsymbol{U}}_i^t \leftarrow \mathcal{A}_{\mathsf{user}}\,(\widehat{\boldsymbol{V}}^t, \Omega_i, \mathsf{P}_\Omega(\boldsymbol{M})_i, T, \lambda, \Gamma_{\boldsymbol{u}})$
       end
3      $\widehat{\boldsymbol{U}}^t \leftarrow [\widehat{\boldsymbol{U}}_1^t, \cdots, \widehat{\boldsymbol{U}}_n^t]^\top$
4      if $t = T$ then
           break;
       end
5      $\widehat{\boldsymbol{V}}^{t+1} \leftarrow \mathcal{A}_{\mathsf{item}}\,(\widehat{\boldsymbol{U}}^t, \Omega, \mathsf{P}_\Omega(\boldsymbol{M}), k, \lambda, \Gamma_{\boldsymbol{u}}, \Gamma_{\boldsymbol{M}})$
   end
6  return $\widehat{\boldsymbol{U}}^T, \widehat{\boldsymbol{V}}^T$

   Procedure $\mathcal{A}_{\mathsf{item}}$ ($\boldsymbol{U}$, $\Omega$, $\mathsf{P}_\Omega(\boldsymbol{M})$, $k$, $\lambda$, $\Gamma_{\boldsymbol{u}}$, $\Gamma_{\boldsymbol{M}}$)
7      $\Omega' \leftarrow$ up to $k$ random samples of $(i,j) \in \Omega, \forall i \in [n]$.
       for $1 \le j \le m$ do
8          $\boldsymbol{G}_j \leftarrow \mathcal{N}_{\mathsf{sym}}\left(0, \Gamma_{\boldsymbol{u}}^4 \cdot \sigma^2\right)^{r\times r}$
9          $\boldsymbol{g}_j \leftarrow \mathcal{N}\left(0, \Gamma_{\boldsymbol{u}}^2 \Gamma_{\boldsymbol{M}}^2 \cdot \sigma^2\right)^r$
10         $\boldsymbol{X}_j \leftarrow \lambda \boldsymbol{I} + \sum_{i\in\Omega'_j} \boldsymbol{U}_i \otimes \boldsymbol{U}_i + \boldsymbol{G}_j$
11         $\boldsymbol{V}_j \leftarrow \Pi_{\mathsf{PSD}}\,(\boldsymbol{X}_j)^+ \left(\sum_{i\in\Omega'_j} \boldsymbol{M}_{ij} \cdot \boldsymbol{U}_i + \boldsymbol{g}_j\right)$
       end
12     $\widetilde{\boldsymbol{V}} = [\boldsymbol{V}_1, \cdots, \boldsymbol{V}_m]^\top$
13     return $\boldsymbol{V} = \widetilde{\boldsymbol{V}}(\widetilde{\boldsymbol{V}}^\top \widetilde{\boldsymbol{V}})^{-1/2}$

   Procedure $\mathcal{A}_{\mathsf{user}}$ ($\boldsymbol{V}$, $\Omega_i$, $\mathsf{P}_\Omega(\boldsymbol{M})_i$, $T$, $\lambda$, $\Gamma_{\boldsymbol{u}}$)
14     $\Omega'_i \leftarrow$ random samples of $1/T$ fraction of $j \in \Omega_i$
15     $\boldsymbol{u} \leftarrow (\lambda \boldsymbol{I} + \sum_{j\in\Omega'_i} \boldsymbol{V}_j \otimes \boldsymbol{V}_j)^{-1} \sum_{j\in\Omega'_i} \boldsymbol{M}_{ij} \boldsymbol{V}_j$
16     return clip $(\boldsymbol{u}, \Gamma_{\boldsymbol{u}})$
```

$\mathsf{clip}\,(\boldsymbol{u}, c) = \boldsymbol{u} \cdot \max(1, c/\|\boldsymbol{u}\|_2)$, i.e., the projection of $\boldsymbol{u}$ on an ℓ_2 ball of radius c. Let $\boldsymbol{A}^+$ be the pseudoinverse of $\boldsymbol{A}$.

3.1. Algorithm

The private alternating least squares algorithm, DPALS, is described in Algorithm 1. It follows the standard ALS steps, i.e. it alternatingly solves the least squares problem to obtain $\widehat{\boldsymbol{U}}$ and $\widehat{\boldsymbol{V}}$ using (1) and (2). To guarantee joint DP, we compute differentially private item embeddings $\widehat{\boldsymbol{V}}^{t+1}$ (using procedure $\mathcal{A}_{\mathsf{item}}$) by solving a private variant of (2), and compute each row of $\widehat{\boldsymbol{U}}^{t+1}$ *independently without any noise* using procedure $\mathcal{A}_{\mathsf{user}}$. A block schematic of the algorithm is presented in Figure 1.

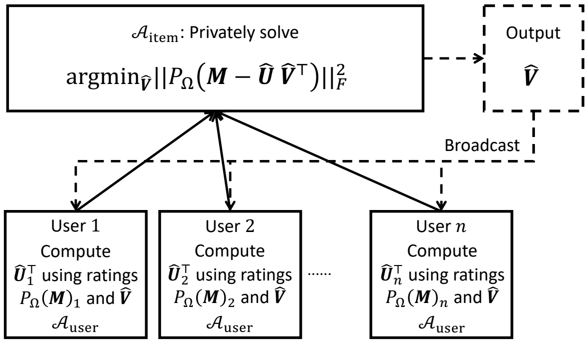

$\mathcal{A}_{\text{item}}$: Privately solve
$\text{argmin}_{\widehat{V}} \|P_\Omega(\boldsymbol{M} - \widehat{\boldsymbol{U}}\,\widehat{\boldsymbol{V}}^\top)\|_F^2$
Output
$\widehat{\boldsymbol{V}}$
Broadcast
User 1
Compute
$\widehat{\boldsymbol{U}}_1^\top$ using ratings
$P_\Omega(\boldsymbol{M})_1$ and $\widehat{\boldsymbol{V}}$
$\mathcal{A}_{\text{user}}$
User 2
Compute
$\widehat{\boldsymbol{U}}_2^\top$ using ratings
$P_\Omega(\boldsymbol{M})_2$ and $\widehat{\boldsymbol{V}}$
$\mathcal{A}_{\text{user}}$
......
User n
Compute
$\widehat{\boldsymbol{U}}_n^\top$ using ratings
$P_\Omega(\boldsymbol{M})_n$ and $\widehat{\boldsymbol{V}}$
$\mathcal{A}_{\text{user}}$

Figure 1. Block schematic of Joint differentially private alternating least squares algorithm. Solid lines and boxes represent privileged computations not visible to an adversary or other users. Dashed boxes and lines are public information accessible to anyone.

Here we describe how the privacy is guaranteed in $\mathcal{A}_{\mathsf{item}}$; see Theorem 1 for a formal statement. For a given $j \in [m]$, write $\boldsymbol{H}_j^t = \lambda \boldsymbol{I} + \sum_{i\in\Omega_j} \widehat{\boldsymbol{U}}_i^t \otimes \widehat{\boldsymbol{U}}_i^t$ and $\boldsymbol{w}_j^t = \sum_{i\in\Omega_j} \boldsymbol{M}_{ij} \widehat{\boldsymbol{U}}_i^t$. Then the non-private update (2) can be written as $\widehat{\boldsymbol{V}}_j^{t+1} = \left(\boldsymbol{H}_j^t\right)^{-1} \boldsymbol{w}_j^t$. In the private version, we need to add noise to protect both $\boldsymbol{H}_j^t$ and $\boldsymbol{w}_j^t$. To ensure sufficient noise, we limit the influence of each user by "clipping" each $\widehat{\boldsymbol{U}}_i^t$ to a bounded ℓ_2 norm $\Gamma_{\boldsymbol{u}}$ (Line 16 in $\mathcal{A}_{\mathsf{user}}$) and resampling Ω such that each user participates in at most k items' computation (Line 7 in $\mathcal{A}_{\mathsf{item}}$). We then apply the Gaussian mechanism to $\boldsymbol{H}_j^t$ and $\boldsymbol{w}_j^t$ before using them to compute $\widehat{\boldsymbol{V}}_j^{t+1}$ in $\mathcal{A}_{\mathsf{item}}$ (Lines 9–12).

While the above procedure is sufficient to guarantee privacy, we need a few additional modifications for the stability of the algorithm and the utility analysis.

Initialization. Random initialization has worked well for our empirical study. For our utility analysis, we need $\widehat{\boldsymbol{V}}^0$ to be reasonably close to $\boldsymbol{V}^*$ (in terms of spectral norm). One can compute this by estimating the rank-r top singular subspace of the matrix $\boldsymbol{A} = \mathsf{P}_\Omega(M)^T \mathsf{P}_\Omega(M)$ while preserving differential privacy.

Sampling from Ω. To ease the analysis, we require the observed values to be independent across different steps. This is achieved by resampling from Ω at the beginning of $\mathcal{A}_{\mathsf{item}}$ (Line 7) and $\mathcal{A}_{\mathsf{user}}$ (Line 14). The sampling in $\mathcal{A}_{\mathsf{item}}$ is more important as it also limits the number of items per user, for privacy purposes. In practice, we omit the sampling in $\mathcal{A}_{\mathsf{user}}$, and sample only once for $\mathcal{A}_{\mathsf{item}}$. The sampling distribution used in the latter has a significant impact in practice, as discussed in Section 5.1.

Projection to the PSD cone. In our analysis, we show that $\boldsymbol{X}_j$ (Line 10) is positive definite with high probability. In practice, we project $\boldsymbol{X}_j = \boldsymbol{H}_j + \boldsymbol{G}_j$ to the PSD cone to improve stability of the algorithm. This has a significant impact on performance in practice, as discussed in Section 6.

3.2. Computational Complexity

The computational complexity of DPALS is comparable to that of ALS, which is known to be scalable to very large matrices. More precisely, the V step of ALS involves computing $\boldsymbol{H}_j^t$ and $\boldsymbol{w}_j^t$, in $O(|\Omega'|r^2)$, then solving the m linear systems $\widehat{\boldsymbol{V}}_j^{t+1} = \left(\boldsymbol{H}_j^t\right)^{-1} \boldsymbol{w}_j^t$ in $O(mr^3)$, for a total complexity of $O(|\Omega'|r^2 + mr^3)$ (and similarly for the U step). Given that the rank r is typically a small constant, this scales linearly in the number of observations $|\Omega'|$ and the number of items m. In the private version ($\mathcal{A}_{\text{user}}$), the only additional operations are forming the noise matrices (Lines 8–9) in $O(mr^2)$, and projecting $\boldsymbol{X}_j$ on the PSD cone (Line 11), in $O(mr^3)$, so the total complexity per iteration is the same as ALS in big-O notation.

In comparison, the complexity of the DPFW method is $O(m^2 + |\Omega'|k)$, and can be reduced to $O(\Gamma(m + |\Omega'|))$ using a more efficient stochastic approximation. The per-iteration complexity also scales linearly in m and $|\Omega'|$. Even though the per-iteration complexity of DPFW and DPALS are comparable, DPALS converges in much fewer iterations (see Appendix D.4 for an example), which makes it more scalable in practice.

3.3. Privacy Guarantee

We now provide the privacy guarantee for DPALS. As each subroutine in DPALS is a variant of the Gaussian mechanism, we can apply the Rényi accounting (Mironov, 2017) and convert to (ε, δ)-DP. See Appendix A for the proof.

Theorem 1 (Privacy guarantee)**.** *With random initialization of $\widehat{\boldsymbol{V}}^0$, Algorithm 1 is $\left(\alpha, \alpha\rho^2\right)$-joint RDP with $\rho^2 = \frac{kT}{2\sigma^2}$. Hence for any $\varepsilon > 0$ and $\delta \in (0,1)$, Algorithm 1 is (ε, δ)-joint DP if we set $\sigma = \frac{\sqrt{(2kT)(\varepsilon + \ln(1/\delta))}}{\varepsilon}$.*

The guarantee holds for all values of the parameters $\Gamma_{\boldsymbol{u}}$, $\Gamma_{\boldsymbol{M}}$, T, λ, r, k. Note in particular that the scale of the noise (Lines 8–9 in Algorithm 1) is normalized so that the expression of σ in Theorem 1 does not depend on $\Gamma_{\boldsymbol{u}}, \Gamma_{\boldsymbol{M}}$.

Remark 1. Given a target (ε, δ), the parameters k, T directly determine σ, and are important parameters to tune in practice, alongside other hyper-parameters such as λ. For instance, a larger k means sampling more data per user (potentially improving model quality), but also requires more noise (potentially degrading model quality).

4. Convergence Guarantee for DPALS

We now show that under standard low-rank matrix completion assumptions (Assumptions 1 and 2), Algorithm 1, with the noisy power method initialization, solves the matrix completion problem accurately.

Theorem 2. *Suppose that $\boldsymbol{M}$ is a μ-incoherent rank-r matrix, and Ω consists of random observations with probability p. Let $\kappa := \sigma_1^*/\sigma_r^*$ be the condition number of $\boldsymbol{M}$, where $\sigma_1^* \geq \cdots \sigma_r^* > 0$ are the singular values of $\boldsymbol{M}$.*

There exists a universal constant $C > 0$, such that for all $\delta \in (0,1)$, $\varepsilon \in (0, \ln(1/\delta))$, if $p \geq \mu^6\kappa^{12}r^6 \cdot \frac{\log^3 n}{m}$ and $\sqrt{p}n \geq C\frac{\gamma \log(1/\delta)}{\varepsilon}$, where $\gamma = C\kappa^6\mu^3r^2\sqrt{m} \cdot \log^2(\kappa \cdot n)$, then DPALS, initialized with $\boldsymbol{V}^0$ s.t. $\|(I - \boldsymbol{V}^*(\boldsymbol{V}^*)^\top)\boldsymbol{V}^0\| \leq \frac{C}{\kappa^2 r^2 \ln n}$, with parameters $k = C \cdot m \cdot p \log n$, $T = \log(\mu\kappa n/\varepsilon)$, $\sigma = \frac{C\sqrt{kT\ln(1/\delta)}}{\varepsilon}$, $\Gamma_{\boldsymbol{u}} = \frac{C\mu\sigma_1^*\sqrt{r}}{\sqrt{n}}$, $\Gamma_{\boldsymbol{M}} = \frac{\mu^2 r\sigma_1^*}{\sqrt{mn}}$ and $\lambda = 0$, returns $\widehat{\boldsymbol{U}}^T$ and $\widehat{\boldsymbol{V}}^T$ such that the following holds:

- *The distribution of $(\widehat{\boldsymbol{U}}^T, \widehat{\boldsymbol{V}}^T)$ satisfies (ε, δ)-joint DP.*
- *$\|\boldsymbol{M} - \widehat{\boldsymbol{U}}^T(\widehat{\boldsymbol{V}}^T)^\top\|_F \leq C \cdot \frac{\sqrt{m}\log(1/\delta)}{\varepsilon \cdot n} \cdot \frac{\kappa\gamma}{\sqrt{p}}\|\boldsymbol{M}\|_F$, with probability $\geq 1 - 1/n^{10}$.*
- *Similarly, $\|\boldsymbol{M} - \widehat{\boldsymbol{U}}^T(\widehat{\boldsymbol{V}}^T)^\top\|_\infty \leq C \cdot \frac{m\log(1/\delta)}{\varepsilon \cdot n} \cdot \frac{\kappa\gamma}{\sqrt{p}} \cdot \frac{\mu^2 r\|M\|_2}{\sqrt{mn}}$, with probability $\geq 1 - 1/n^{10}$.*

Remark 2 (Initialization)**.** Following (Jain et al., 2013), we can use differentially private SVD of $P_\Omega(\boldsymbol{M})$ to obtain an initial estimate of $\boldsymbol{V}^{(0)}$. To this end, Algorithm 1 of (Dwork et al., 2014) can be used directly. That is, we compute top-r eigenvectors of $\boldsymbol{A} = \mathsf{P}_\Omega(\boldsymbol{M})^\top\mathsf{P}_\Omega(\boldsymbol{M}) + \boldsymbol{G}$ where $\boldsymbol{G}$ is a symmetric Gaussian matrix with standard deviation $\sigma\Gamma_{\boldsymbol{M}}^2$. However, standard analysis of Analyze Gauss requires $n = \widetilde{\Omega}(m\sqrt{m}/\varepsilon)$ to obtain an estimate within required error bound. See Appendix B for a brief analysis of the initialization scheme.

Remark 3. The choice of hyper-parameters in Theorem 2 assumes knowledge of certain quantities such as r, μ, κ. In practice, these quantities are unknown, but one can use standard DP hyper-parameter search techniques (Liu & Talwar, 2019) to search for optimal hyper-parameter values.

Remark 4. The number of samples needed per user is about $p \cdot m = O(\mu^6\kappa^{12}r^6\log^3 n)$ which is nearly optimal with respect to m and n. This represents a significant improvement over the DPFW algorithm in (Jain et al., 2018) which requires $\Omega(\sqrt{m})$ samples per user.

Remark 5. We did not optimize bounds for dependence on the rank r and condition number κ. Prior work tends to focus on the dependence on the size (m and n) and polynomial dependence on r, κ is common even in the non-private setting. For example, the dependence is $r^{4.5}$ and κ^6 in (Jain et al., 2013); r^7 and κ^6 in (Sun & Luo, 2015); r^6 and κ^{16} in (Ge et al., 2016). Our main goal is to provide a guarantee in the private setting that is competitive with the non-private setting, so we inherit the focus on the size m, n from prior work. Furthermore, dependence on κ can be removed (up to log factors) by using a *stagewise* ALS method similar to (Hardt & Wootters, 2014). However, this further complicates the proof and the practical performance of standard

ALS is comparable to such stagewise methods. Finally, there is empirical evidence that in several recommendation problems, data is close to low rank (for example, in the Netflix prize, Koren (2008) showed a high accuracy for $r = 50$) and it is common in industrial applications for m, n to be several orders of magnitude larger than academic benchmarks, which alleviates the dependence on r.

Remark 6. Our Frobenius norm error bound is significantly smaller than the bound for the DPFW algorithm, which is given by $\|\boldsymbol{M} - \widehat{\boldsymbol{U}}^T(\widehat{\boldsymbol{V}}^T)^\top\|_F \leq \left(\frac{m^{5/4}}{n\varepsilon}\right)^{1/5}\|\boldsymbol{M}\|_F$. In particular, to ensure an error $\|\boldsymbol{M} - \widehat{\boldsymbol{U}}^T(\widehat{\boldsymbol{V}}^T)^\top\|_F \leq \zeta\|\boldsymbol{M}\|_F$, DPALS requires $n \geq \frac{Cm}{\zeta \cdot \varepsilon}$, while DPFW requires $n \geq \frac{Cm^{5/4}}{\zeta^5 \cdot \varepsilon}$, which is significantly worse in terms of ζ. Furthermore, the DPFW bound is a generalization bound, i.e., there is an additional bias term which can be large, and to the best of our knowledge, existing techniques (even in the non-private setting) require incoherence to control this term.

Remark 7. Consider a set of m linear regression problems in r-dimensions: $\{\boldsymbol{y}_{(i)} = \boldsymbol{X}\theta^*_{(i)}\}_{i=1}^m$, with $\boldsymbol{X} \in \mathbb{R}^{n \times r}$. One can use a single iteration of DPALS with ($\widehat{\boldsymbol{U}} = \boldsymbol{X}$ and $\mathsf{P}_\Omega(\boldsymbol{M}) = [\boldsymbol{y}_{(1)}, \ldots, \boldsymbol{y}_{(m)}]$) to solve these linear regression problems. Assuming the conditions on $\boldsymbol{M}$ are satisfied, we can obtain an excess empirical risk of $\widetilde{O}(\sqrt{m}/(\varepsilon n))$. This matches the best known upper bound for solving a set of linear regressions with privacy (Sheffet, 2019; Smith et al., 2017). So, a better convergence rate of DPALS would lead to a tighter bound on solving a set of linear regressions with a common feature matrix. For $m = O(1)$, we know that the lower bound for private linear regression is $\widetilde{\Omega}(1/\varepsilon n)$ (Smith et al., 2017). Thus, we conjecture that the error for DPALS is tight w.r.t. m and εn.

Remark 8. Instead of using the perturbed objective function to estimate $\widehat{\boldsymbol{V}}^t$ in DPALS, one can use DPSGD (Bassily et al., 2014) to do the same (solving a least squares problem with $\widehat{\boldsymbol{U}}^t$ fixed). We leave the empirical comparison of this approach to future work. However, we know that for least-square losses, perturbing the objective is known to be theoretically optimal (Smith et al., 2017).

Proof sketch: First, we show that under the assumptions in Theorem 2, w.h.p., clipping and sampling operations in DPALS have no effect. Note, using $k \geq Cp \cdot m \log n$, w.p. $\geq 1 - 1/n^{100}, \forall i, |\Omega_i| \leq k$. Furthermore, using Lemma 3, $\|\widehat{\boldsymbol{U}}_i^t\| \leq \Gamma_{\boldsymbol{u}}$. Similarly, using Lemma 3, $\sigma_{\min}(\boldsymbol{X}) \geq p/4 - \|\boldsymbol{G}\|_2 \geq p/4 - \Gamma_{\boldsymbol{u}}^2\sigma\sqrt{r} \geq p/8$. That is, $\boldsymbol{X} \succ 0$.

The above observation implies that, under the assumptions of the theorem, Algorithm 1 is essentially performing the following iterative steps:

i) $\widehat{\boldsymbol{U}}^t = \arg\min_{\widehat{\boldsymbol{U}}} \|\mathsf{P}_\Omega(\boldsymbol{M} - \widehat{\boldsymbol{U}}(\widehat{\boldsymbol{V}}^t)^\top)\|_F^2$, and

ii) $\widehat{\boldsymbol{V}}_j^{t+1} = \Big(\boldsymbol{I} + \sum_{i \in \Omega'_j} \widehat{\boldsymbol{U}}_i^t \otimes \widehat{\boldsymbol{U}}_i^t + \boldsymbol{G}\Big)^{-1}\Big(\sum_{i \in \Omega'_j} \boldsymbol{M}_{ij}\widehat{\boldsymbol{U}}_i^t + \boldsymbol{g}\Big)$.

Let $\boldsymbol{U}^t$ (resp. $\boldsymbol{V}^t$) be the Q part in the QR decomposition of $\widehat{\boldsymbol{U}}^t$ (resp. $\widehat{\boldsymbol{V}}^t$). Using Lemma 4, we get $\mathrm{Err}(\boldsymbol{V}^*, \boldsymbol{V}^{t+1}) \leq \frac{1}{4}\mathrm{Err}(\boldsymbol{V}^*, \boldsymbol{V}^t) + \alpha$, where $\mathrm{Err}(\boldsymbol{V}^*, \boldsymbol{V}) = \|(\boldsymbol{I} - \boldsymbol{V}^*(\boldsymbol{V}^*)^\top)\boldsymbol{V}\|_F$ and $\alpha \leq \frac{C\kappa^6 \cdot \mu^3 r^2 \sqrt{\log n}}{\sqrt{p}n} \frac{\sqrt{m \log n} \cdot T \log 1/\delta}{\varepsilon}$. That is, after T iterations, $\mathrm{Err}(\boldsymbol{V}^*, \boldsymbol{V}^T) \leq 2\alpha$. The second claim of the theorem now follows from the above observation and Lemma 3. Similarly, the third claim follows by using the bound on $\mathrm{Err}(\boldsymbol{V}^*, \boldsymbol{V}^T)$ and incoherence of $\boldsymbol{U}^T$, $\boldsymbol{V}^T$ (Lemma 3). See Appendix B for a detailed proof.

Lemma 3. *Suppose the assumptions mentioned in Theorem 2 hold. Then, w.p.* $\geq 1 - 5T/n^{100}$, *we have: a) each iterate* $\widehat{\boldsymbol{U}}^t$, $\widehat{\boldsymbol{V}}^t$ *is* $16\kappa\mu$*-incoherent, b)* $1/2 \leq \sigma_q(\widehat{\boldsymbol{U}}^t(\boldsymbol{\Sigma}^*)^{-1}) \leq 2$ *for all* $q \in [r]$, *c)* $1/4 \leq \sigma_q(\frac{1}{p}\sum_{i:(i,j) \in \Omega^{v,t}} \widehat{\boldsymbol{u}}_i^t(\widehat{\boldsymbol{u}}_i^t)^\top) \leq 4$.

Lemma 4. *Suppose the assumptions mentioned in Theorem 2 hold. Also, let* $\boldsymbol{V}^t$ *be* $16\kappa\mu$*-incoherent s.t.* $\mathrm{Err}(\boldsymbol{V}^*, \boldsymbol{V}^t) \leq \frac{1}{\kappa^2 \log^2 n}$. *Then, w.p.* $\geq 1 - 5T/n^{100}$, *we have* $\mathrm{Err}(\boldsymbol{U}^*, \boldsymbol{U}^t) \leq \frac{1}{2}\mathrm{Err}(\boldsymbol{V}^*, \boldsymbol{V}^t)$, *and* $\mathrm{Err}(\boldsymbol{V}^*, \boldsymbol{V}^{t+1}) \leq \frac{1}{2}\mathrm{Err}(\boldsymbol{U}^*, \boldsymbol{U}^t) + \frac{C\kappa^6 \cdot \mu^3 r^2 \sqrt{\log n}}{\sqrt{p}n} \frac{\sqrt{m \log n} \cdot T \log 1/\delta}{\varepsilon}$, *where* $\mathrm{Err}(\boldsymbol{V}^*, \boldsymbol{V}) = \|(\boldsymbol{I} - \boldsymbol{V}^*(\boldsymbol{V}^*)^\top)\boldsymbol{V}\|_F$.

5. Heuristic Improvements to DPALS

We introduce heuristics to improve the privacy/utility tradeoff for Algorithm 1 in practice. We describe each heuristic, its motivation, and explain how we implement it differentially privately.

5.1. Reducing Distribution Skew

The first heuristics are motivated by the observation that, in practice, the elements of Ω are not sampled uniformly at random (Marlin et al., 2007). In particular, the number of observed ratings per item typically follows a power-law distribution, and is heavily skewed towards popular items. For example, Figure 2 shows the fraction of observations vs. fraction of top movies in the MovieLens 10M data set. It shows, for instance, that the top 20% of the movies account for more than 85% of the observations.

Due to this popularity bias, some items may have very few observations, and for such rare items j, the embedding $\boldsymbol{V}_j$ learned by DPALS may not be useful: The noise terms in Line 11 of Algorithm 1 do not scale with the number of observations $|\Omega'_j|$ – for otherwise we may lose the protection on users who rated rare items – thus, items with a smaller $|\Omega'_j|$ have a lower signal-to-noise ratio. In our experiments, we found that such noisy embeddings may have a further

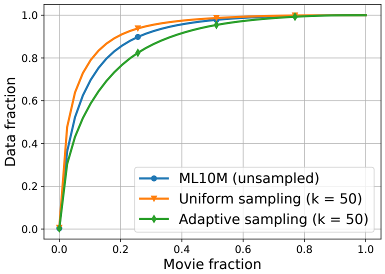


Figure 2. Fraction of observations contributed by the top movies in MovieLens 10M. Adaptive sampling reduces popularity bias.

cascading effect and lead to quality degradation in the embeddings of other movies and users. To alleviate this issue, we propose two techniques.

Learning on frequent items. The first strategy is to partition the items into two sets, based on an estimate of the item counts, which we denote by $\widetilde{\boldsymbol{c}} \in \mathbb{R}^n$. We introduce a hyper-parameter β representing the fraction of movies to train on. Define the set $\mathsf{Frequent}$ to be the $\lceil m\beta \rceil$ items with the largest $\widetilde{\boldsymbol{c}}$, and let $\mathsf{Infrequent}$ be its complement. We learn embeddings $\widehat{\boldsymbol{V}}_j$ only for $j \in \mathsf{Frequent}$, by running Algorithm $\mathcal{A}_{\mathsf{DPALS}}$ on those items. When making predictions for any missing entry $\boldsymbol{M}_{ij}$, if $j \in \mathsf{Frequent}$, we use the dot product $\widehat{\boldsymbol{U}}_i \cdot \widehat{\boldsymbol{V}}_j$, and if $j \in \mathsf{Infrequent}$ we use the average observed rating of $\mathsf{P}_\Omega(\boldsymbol{M})_i$.

To compute $\widetilde{\boldsymbol{c}}$ privately, notice that since each user contributes at most k items, the exact item count $\boldsymbol{c}$ has ℓ_2 sensitivity $\sqrt{k}$. Thus, $\widetilde{\boldsymbol{c}} := \boldsymbol{c} + \mathcal{N}(0, k\sigma^2)$ guarantees $\left(\alpha, \alpha/2\sigma^2\right)$-RDP.

Adaptive sampling. To further reduce the popularity bias, we propose to use an adaptive distribution when subsampling Ω. Recall that in Line 7 of $\mathcal{A}_{\mathsf{item}}$, we pick k items per user in Ω, in order to limit the privacy loss. We propose to sample rare items with higher probability, as follows. Given the count estimate $\widetilde{\boldsymbol{c}}$, for each user i, we pick the k items in $\Omega_i \cap \mathsf{Frequent}$ with the lowest count estimates. This heuristic effectively reduces the distribution skew and gives a significant utility gain compared to uniform sampling, see Section 6.3. Figure 2 illustrates the resulting distribution for a sample size of $k = 50$ per user. It's interesting to observe that under uniform sampling, the popularity bias is worse than in the unsampled data set, this is due to a negative correlation between user counts and item counts: conditioned on a light user, the probability to observe a rare item is lower; see Appendix C for further discussion.

5.2. Additional Heuristics

A common heuristic, used for example by (McSherry & Mironov, 2009), is to center the observed matrix $\mathsf{P}_\Omega(\boldsymbol{M})$, by subtracting an estimate of the global average, denoted by $\widetilde{m}$. To compute $\widetilde{m}$ privately, since $\|\mathsf{P}_\Omega(\boldsymbol{M})\|_\infty \le \Gamma_{\boldsymbol{M}}$ and each user contributes at most k items, publishing $\widetilde{m} = \frac{\sum_{(i,j)\in\Omega} M_{ij} + \mathcal{N}(0, k\Gamma_{\boldsymbol{M}}^2\sigma^2)}{|\Omega| + \mathcal{N}(0, k\sigma^2)}$ guarantees $\left(\alpha, \alpha/\sigma^2\right)$-RDP.

Another practice, commonly used in some benchmarks, is to modify the loss function in Section 2.2 by adding the term $\lambda_0 \|\widehat{\boldsymbol{U}}\widehat{\boldsymbol{V}}^\top\|_F^2$, where λ_0 is a hyper-parameter. This is particularly important for item recommendation tasks, such as the MovieLens 20M benchmark. This modification introduces an additional term $\boldsymbol{K} := \lambda_0 \sum_{i\in[n]} \widehat{\boldsymbol{U}}_i \otimes \widehat{\boldsymbol{U}}_i$ to $\boldsymbol{X}$ in Line 10 of $\mathcal{A}_{\mathsf{item}}$. To maintain privacy, we use a noisy version $\widetilde{\boldsymbol{K}}$ obtained by adding Gaussian noise to $\boldsymbol{K}$. Since $\boldsymbol{K}$ is independent of j, we reuse the same $\widetilde{\boldsymbol{K}}$ for all $j \in [m]$, thus limiting the additional privacy loss due to this term.

Finally, we account for the privacy cost in the computation of $\widetilde{m}$, $\widetilde{\boldsymbol{c}}$, and $\widetilde{\boldsymbol{K}}$, along with that in Theorem 1, by standard composition properties of RDP (Mironov, 2017). For completeness, the privacy accounting of the full algorithm including data pre-processing, is given in Appendix C.

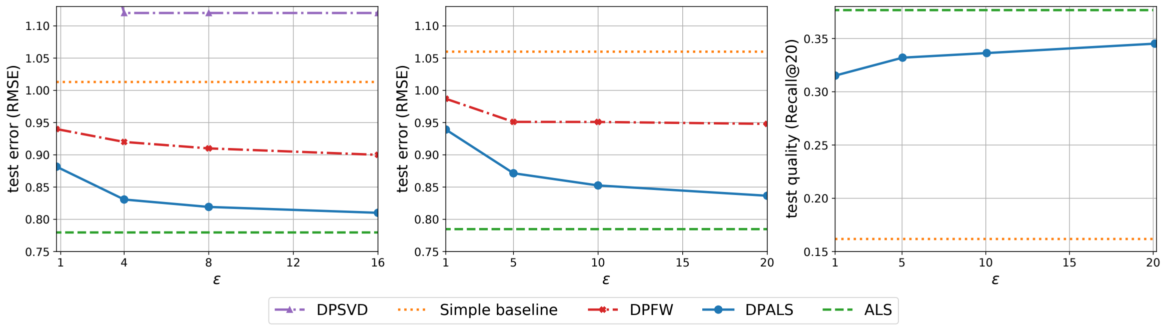
6. Empirical Evaluation

We run experiments on synthetic data and two benchmark tasks on the widely used MovieLens data sets (Harper & Konstan, 2016). The synthetic task follows the assumptions of our theoretical analysis, and serves to illustrate the guarantees of Theorem 2. The MovieLens benchmark tasks serve as an evaluation of the empirical privacy/utility trade-off on a more realistic application, and to provide some practical insights into DPALS. We use current SOTA method DPFW as the main baseline as it is already demonstrated to be more accurate than techniques like Private SVD (McSherry & Mironov, 2009). Similar to (Jain et al., 2018), we do not compare against (Liu & Talwar, 2019) as the privacy parameters are unclear, and might require (exponential time) Markov chain based sampling methods to compute them.

6.1. Metrics and Data Sets

Metrics. The quality of a learned model $(\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{V}})$ will be measured either using the RMSE or the Recall@k, depending on the benchmark. The RMSE is defined as $\mathrm{RMSE} = \|\mathsf{P}_{\Omega^{\mathsf{test}}}(\widehat{\boldsymbol{U}}\widehat{\boldsymbol{V}}^\top - \boldsymbol{M})\|_F / \sqrt{|\Omega^{\mathsf{test}}|}$, where Ω^{test} is the set of test ratings held out from Ω. Recall@k is defined as follows. For each user i, let R_i be the set of k movies with the highest scores, where the score of movie j is $\widehat{\boldsymbol{U}}_i \cdot \widehat{\boldsymbol{V}}_j$. Then Recall@k $= \frac{1}{n}\sum_{i=1}^n |R_i \cap \Omega_i^{\mathsf{test}}| / \min(k, |\Omega_i^{\mathsf{test}}|)$.

Synthetic data. We generate a rank 5 ground truth matrix as the product of two random orthogonal matrices $\boldsymbol{U}^* \in \mathbb{R}^{n\times 5}, \boldsymbol{V}^* \in \mathbb{R}^{m\times 5}$, where $m = 1000$, and $n \in \{5000, 10000, 20000, 50000\}$. We scaled the ground truth matrix such that the standard deviation of the observations is 1, in other words, a trivial model which always



(a) ML-10M (top 400 movies)

(b) ML-10M

(c) ML-20M

Figure 3. Privacy/utility trade-off of different methods. We observe that DPALS is significantly more accurate than DPFW method, and the loss in accuracy for DPALS compared to ALS is relatively small, especially for $\varepsilon \geq 10$.

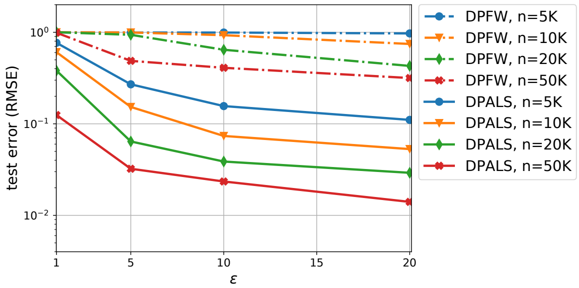


Figure 4. Comparison of DPFW and DPALS on synthetic data with different number of rows/users n.

predicts the global average has a RMSE of 1. The observed entries Ω are obtained by sampling each entry independently with probability $p = 20\log(n)/m$.

MovieLens data sets. We apply our method to two common recommender benchmarks: (i) rating prediction on MovieLens 10M (ML-10M) following Lee et al. (2013), where the task is to predict the value of a user's rating, and performance is measured using the RMSE, (ii) item recommendation on MovieLens 20M (ML-20M) following Liang et al. (2018), where the task is to select k movies for each user and performance is measured using Recall@k.

For comparison to DPFW, we use a variant of the ML-10M rating prediction task following Jain et al. (2018), where the movies are restricted to the 400 most popular movies (DPFW did not scale to the full data set with all movies, unlike DPALS).

Experimental protocol. Each data set is partitioned into training, validation and test sets. Hyper-parameters are chosen on the validation set, and the final performance is measured on the test set. The privacy loss accounting is done using RDP, then translated to (ε, δ)-DP with $\delta = 10^{-5}$ for the synthetic data and ML-10M and $\delta = 1/n$ for ML-20M. When training DPALS models on synthetic data, we use the basic version described in Algorithm 1, without heuristics. When training on MovieLens data sets, we use the heuristics described in Section 5. Note that even when training on `Frequent` items (Section 5.1), evaluation is always done on the *full set* of items, so that the reported metrics are comparable to previously published numbers. Additional details on the experimental setup are in Appendix D, including statistics of the data sets, a list of hyper-parameters and the ranges we used for each.

6.2. Privacy-Utility Trade-Off

DPALS vs. DPFW on synthetic data. On synthetic data (Figure 4) we observe: First, as expected, the trade-off of both algorithms improves as the number of users increases. Second, for $\varepsilon = 1$, the quality of the DPFW models is no better than the trivial model (RMSE equal to 1), while DPALS has a lower RMSE, which significantly improves with larger n. Third, for the largest data set ($n = 50K$), the relative improvement in RMSE between DPALS and DPFW is at least 7-fold across all values of ε. To further illustrate the difference between DPALS and DPFW, we show in Appendix D.4 the RMSE against number of iterations, both for the private and non-private variants (Figure 7).

DPALS vs. DPFW on ML10M. Next, we compare the two methods on ML-10M-top400 (Figure 3a). For DPFW and DPSVD, the numbers are taken directly from (Jain et al., 2018). For reference, we include the test RMSE of non-private ALS, and a simple baseline model that always predicts the global average rating. The performance of DPSVD is worse than that of the simple baseline. DPALS performs best, with a relative improvement in RMSE (compared to DPFW) that ranges from 7% to 11.6%, and that increases with ε. In Appendix D.4, we show that DPALS achieves performance better than DPFW even when trained on a small fraction of the users (23%).

Finally, Figure 3b shows the privacy/utility trade-off on the full ML-10M data. In order to scale DPFW to the the full data, we use the same procedure described in Sec-

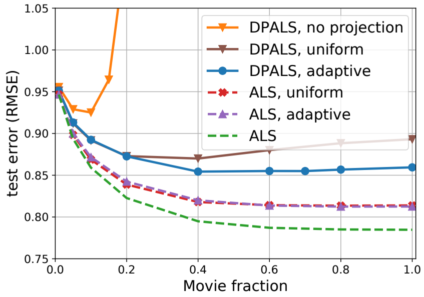


Figure 5. RMSE vs. movie fraction for $\varepsilon = 10$ on ML-10M.

tion 5: DPFW is trained on the top movies, and for remaining movies the model predicts the user's average rating. Compared to the restricted data set (ML-10M-top400), the privacy-utility trade-off is worse on the full data. This indicates that a smaller ratio between number of users and number of items makes the task harder – a result that is in line with the theory.

The results on synthetic data and ML10M suggest that DPALS exhibits a much better privacy/utility trade-off than DPFW, and a better dependence on the number of rows n, which is consistent with the theoretical analysis.

DPALS on MovieLens 20M. Figure 3c shows the privacy/utility trade-off of DPALS on the ML-20M data set. We include as a reference the non-private ALS, and a simple baseline model that always returns the k most rated movies.

On this task, the performance of the private model is remarkably good. Indeed, the best previously reported Recall@20 numbers for *non-private* models on this benchmark are 36.0% for ALS (Liang et al., 2018) and 41.4% using a sophisticated auto-encoder model (Shenbin et al., 2020). Our results show that DPALS can achieve performance comparable to the previously reported state of the art numbers for (non-private) matrix completion, and the utility does not significantly degrade, even at small ε.

6.3. Importance of Adaptive Sampling and Projection

In this section, we give additional insights into the effect of the heuristics introduced in Section 5. We run a study on ML-10M for $\varepsilon = 10$, $r = 128$ and a sample size $k = 50$ (both correspond to the best overall model); other hyper-parameters are re-tuned. According to Section 5.1, we partition the set of movies into Frequent and Infrequent and train only on Frequent . The results are reported in Figure 5, where the movie fraction is simply defined as the fraction $|\text{Frequent}|/n$. We make the following observations. First, for non-private ALS, we get the highest RMSE by training on all movies, while there is a benefit for training on a subset of the movies for the private models. Second, when training the non-private model on sub-sampled data (red and purple lines), there is a considerable increase in RMSE, from 0.785 to 0.812. This gives an indication that part of the utility loss is due to sub-sampling, and not simply due to the addition of noise. Third, the sampling strategy has a significant impact on the performance of the private DPALS model: adaptive sampling improves the RMSE from 0.870 to 0.854, in contrast, the sampling strategy appears to have little effect on non-private models (i.e. models trained without noise). Finally, training the private model without PSD projection (Π_{PSD} in Line 11 of Algorithm 1) results in a terrible performance. We find that while the projection is not technically necessary for the theoretical analysis, it is essential in practice.

Training on a subset of the movies appears to have only a marginal effect when combined with adaptive sampling in this experiment. However, as detailed in the appendix, the effect is much more significant for smaller ε, as well as on the ML-20M task.

Additional experiments are presented in Appendix D, to explore the effect of other hyper-parameters, such as the rank and the regularization of the objective function.

7. Conclusion

We presented DPALS for solving low-rank matrix completion with user-level privacy protection. We show that DPALS provably converges to high accuracy outputs under standard assumptions and, with careful implementation, significantly outperforms existing privacy preserving matrix completion methods. In fact, DPALS achieves competitive metrics on benchmark data compared to non-private models and scales well with data set size.

The efficiency of DPALS shows that by taking advantage of the structure of the problem, one can achieve a much higher utility for privacy-preserving model training. In this case, the alternating structure of ALS, along with the decoupling of the least squares solution, were essential in the design of an efficient method. These insights may be applicable to a broader class of problems and optimization algorithms.

Acknowledgments

We would like to thank Om Thakkar and the anonymous reviewers for insightful comments and discussion.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proc. of the 2014 IEEE 55th Annual Symp. on Foundations of Computer Science (FOCS)*, 2014.

Bhatia, R. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

Calandrino, J. A., Kilzer, A., Narayanan, A., Felten, E. W., and Shmatikov, V. "you might also like:" privacy risks of collaborative filtering. In *2011 IEEE symposium on security and privacy*, pp. 231–246. IEEE, 2011.

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 267–284, 2019.

Carlini, N., Deng, S., Garg, S., Jha, S., Mahloujifar, S., Mahmoody, M., Song, S., Thakurta, A., and Tramer, F. An attack on instahide: Is private learning possible with instance encoding? *arXiv preprint arXiv:2011.05315*, 2020a.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*, 2020b.

Dinur, I. and Nissim, K. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 202–210, 2003.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology—EUROCRYPT*, pp. 486–503, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Proc. of the Third Conf. on Theory of Cryptography (TCC)*, pp. 265–284, 2006b.

Dwork, C., McSherry, F., and Talwar, K. The price of privacy and the limits of lp decoding. In *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing*, pp. 85–94, 2007.

Dwork, C., Talwar, K., Thakurta, A., and Zhang, L. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 11–20, 2014.

Ge, R., Lee, J. D., and Ma, T. Matrix completion has no spurious local minimum. *Advances in Neural Information Processing Systems*, pp. 2981–2989, 2016.

Hardt, M. and Roth, A. Beating randomized response on incoherent matrices. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 1255–1268, 2012.

Hardt, M. and Roth, A. Beyond worst-case analysis in private singular vector computation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 331–340, 2013.

Hardt, M. and Wootters, M. Fast matrix completion without the condition number. In *Conference on learning theory*, pp. 638–678. PMLR, 2014.

Hardt, M., Meka, R., Raghavendra, P., and Weitz, B. Computational limits for matrix completion. In *Conference on Learning Theory*, pp. 703–725. PMLR, 2014.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *Acm Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

Hu, Y., Koren, Y., and Volinsky, C. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pp. 263–272, 2008.

Jain, P., Netrapalli, P., and Sanghavi, S. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 665–674, 2013.

Jain, P., Thakkar, O. D., and Thakurta, A. Differentially private matrix completion revisited. In *International Conference on Machine Learning*, pp. 2215–2224. PMLR, 2018.

Kearns, M., Pai, M., Roth, A., and Ullman, J. Mechanism design in large games: Incentives and privacy. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pp. 403–410, 2014.

Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *KDD*, 2008.

Koren, Y. and Bell, R. Advances in collaborative filtering. *Recommender systems handbook*, pp. 77–118, 2015.

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

Korolova, A. Privacy violations using microtargeted ads: A case study. In *2010 IEEE International Conference on Data Mining Workshops*, pp. 474–482. IEEE, 2010.

Lee, J., Kim, S., Lebanon, G., and Singer, Y. Local low-rank matrix approximation. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. II–82–II–90. JMLR.org, 2013.

Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. Variational autoencoders for collaborative filtering. WWW '18, pp. 689–698, 2018.

Liu, J. and Talwar, K. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 298–309, 2019.

Liu, Z., Wang, Y.-X., and Smola, A. Fast differentially private matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 171–178, 2015.

Lu, S., Hong, M., and Wang, Z. PA-GD: On the convergence of perturbed alternating gradient descent to second-order stationary points for structured nonconvex optimization. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 4134–4143, 2019.

Marlin, B. M., Zemel, R. S., Roweis, S., and Slaney, M. Collaborative filtering and the missing at random assumption. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, UAI'07, pp. 267–275, Arlington, Virginia, USA, 2007. AUAI Press.

McSherry, F. and 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*, pp. 627–636, 2009.

Meng, X., Wang, S., Shu, K., Li, J., Chen, B., Liu, H., and Zhang, Y. Personalized privacy-preserving social recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Mironov, I. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275. IEEE, 2017.

Rendle, S., Zhang, L., and Koren, Y. On the difficulty of evaluating baselines: A study on recommender systems. *CoRR*, abs/1905.01395, 2019.

Sheffet, O. Old techniques in differentially private linear regression. In *Algorithmic Learning Theory*, pp. 789–827. PMLR, 2019.

Shenbin, I., Alekseev, A., Tutubalina, E., Malykh, V., and Nikolenko, S. I. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pp. 528–536, 2020.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Smith, A., Thakurta, A., and Upadhyay, J. Is interaction necessary for distributed private learning? In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 58–77. IEEE, 2017.

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 245–248. IEEE, 2013.

Sun, R. and Luo, Z. Guaranteed matrix completion via nonconvex factorization. In *FOCS*, 2015.

Thakkar, O., Ramaswamy, S., Mathews, R., and Beaufays, F. Understanding unintended memorization in federated learning. *arXiv preprint arXiv:2006.07490*, 2020.

Tropp, J. A. An introduction to matrix concentration inequalities. *arXiv preprint arXiv:1501.01571*, 2015.

Vershynin, R. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.