Differentially Private Matrix Completion Revisited

Prateek Jain [1] **Om Thakkar** [2] **Abhradeep Thakurta** [3]

Abstract

We provide the first provably joint differentially private algorithm with formal utility guarantees for the problem of *user-level* privacy-preserving collaborative filtering. Our algorithm is based on the Frank-Wolfe method, and it consistently estimates the underlying preference matrix as long as the number of users m is $\omega(n^{5/4})$, where n is the number of items, and each user provides her preference for at least $\sqrt{n}$ randomly selected items. Along the way, we provide an optimal differentially private algorithm for singular vector computation, based on the celebrated Oja's method, that provides significant savings in terms of space and time while operating on sparse matrices. We also empirically evaluate our algorithm on a suite of datasets, and show that it consistently outperforms the state-of-the-art *private* algorithms.

1. Introduction

Collaborative filtering (or matrix completion) is a popular approach for modeling the recommendation system problem, where the goal is to provide personalized recommendations about certain items to a user (Koren & Bell, 2015). In other words, the objective of a personalized recommendation system is to learn the entire users-items preference matrix $Y^* \in \Re^{m \times n}$ using a small number of user-item preferences $Y^*_{ij}, (i,j) \in [m] \times [n]$, where m is the number of users and n is the number of items. Naturally, in absence of any structure in Y^*, the problem is ill-defined as the unknown entries of Y^* can be arbitrary. Hence, a popular modeling hypothesis is that the underlying preference matrix Y^* is low-rank, and thus, the collaborative filtering problem reduces to that of low-rank matrix completion (Recht, 2011; Candes & Recht, 2012). One can also enhance this formulation using side-information like user-features or item-features (Yu et al., 2014).

Naturally, personalization problems require collecting and analyzing sensitive customer data like their preferences for various items, which can lead to serious privacy breaches (Korolova, 2010; Narayanan & Shmatikov, 2010; Calandrino et al., 2011). In this work, we attempt to address this problem of privacy-preserving recommendations using collaborative filtering (McSherry & Mironov, 2009; Liu et al., 2015). We answer the following question in the **affirmative**: *Can we design a matrix completion algorithm which keeps* **all** *the ratings of a user private, i.e., guarantees* **user-level privacy** *while still providing accurate recommendations?* In particular, we provide the *first* differentially private (Dwork et al., 2006b) matrix completion algorithms with *provable* accuracy guarantees. Differential privacy (DP) is a rigorous privacy notion which formally protects the privacy of any user participating in a statistical computation by controlling her influence to the final output.

Most of the prior works on DP matrix completion (and low-rank approximation) (Blum et al., 2005; Chan et al., 2011; Hardt & Roth, 2012; 2013; Kapralov & Talwar, 2013; Dwork et al., 2014b) have provided guarantees which are non-trivial only in the *entry-level* privacy setting, i.e., they preserve privacy of only a single rating of a user. Hence, they are not suitable for preserving a user's privacy in practical recommendation systems. In fact, their trivial extension to user-level privacy leads to vacuous bounds (see Table 1). Some works (McSherry & Mironov, 2009; Liu et al., 2015) do serve as an exception, and directly address the user-level privacy problem. However, they only show empirical evidences of their effectiveness; they do not provide formal error bounds.[1] In contrast, we provide an efficient algorithm based on the classic Frank-Wolfe (FW) procedure (Frank & Wolfe, 1956), and show that it gives strong utility guarantees while preserving user-level privacy. Furthermore, we empirically demonstrate its effectiveness on various benchmark datasets.

Our private FW procedure needs to compute the top right singular vector of a sparse user preference matrix, while

[1]Microsoft Research. Email: prajain@microsoft.com [2]Department of Computer Science, Boston University. Email: omthkkr@bu.edu [3]Computer Science Department, University of California Santa Cruz. Email: aguhatha@ucsc.edu. Correspondence to: Om Thakkar <omthkkr@bu.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[1]In case of (Liu et al., 2015), the DP guarantee itself might require an exponential amount of computation.

preserving DP. For practical recommendation systems with a large number of items, this step turns out to be a significant bottleneck both in terms of space as well as time complexity. To alleviate this issue, we provide a method, based on the celebrated Oja's algorithm (Jain et al., 2016), which is nearly optimal in terms of the accuracy of the computed singular vector while still providing significant improvement in terms of space and computation. In fact, our method can be used to speed-up even the vanilla differentially private PCA computation (Dwork et al., 2013). To the best of our knowledge, this is the first algorithm for DP singular value computation with optimal utility guarantee, that also exploits the sparsity of the underlying matrix.

Notion of privacy: To measure privacy, we select *differential privacy*, which is a de-facto privacy notion for large-scale learning systems, and has been widely adopted by the academic community as well as big corporations like Google (Erlingsson et al., 2014), Apple (McMillan, 2016), etc. The underlying principle of *standard DP* is that the output of the algorithm should not change significantly due to presence or absence of any user. In the context of matrix completion, where the goal is to release the *entire preference matrix* while preserving privacy, this implies that the computed ratings/preferences for any particular user cannot depend strongly on *her own personal preferences*. Naturally, the resulting preference computation is going to be trivial and inaccurate (which also follows from the reconstruction attacks of (Dinur & Nissim, 2003) and (Hardt & Roth, 2012)).

To alleviate this concern, we consider a relaxed but natural DP notion (for recommendation systems) called *joint differential privacy* (Kearns et al., 2014). Consider an algorithm $\mathcal{A}$ that produces individual outputs Y_i for each user i, i.e., the i-th row of preference matrix Y. Joint DP ensures that for each user i, the output of $\mathcal{A}$ for all other users (denoted by Y_{-i}) does not reveal "much" about the preferences of user i. That is, the recommendations made to all the users except the i-th user do not depend significantly upon the i-th user's preferences. Although not mentioned explicitly, previous works on DP matrix completion (McSherry & Mironov, 2009; Liu et al., 2015) strive to ensure Joint DP. Formal definitions are provided in Section 2.

Granularity of privacy: DP protects the information about a user in the context of presence or absence of her data record. Prior works on DP matrix completion (McSherry & Mironov, 2009; Liu et al., 2015), and its close analogue, low-rank approximation (Blum et al., 2005; Chan et al., 2011; Hardt & Roth, 2012; Dwork et al., 2013; Hardt & Roth, 2013), have considered different variants of the notion of a data record. Some have considered a single entry in the matrix Y^* as a data record (resulting in *entry-level privacy*), whereas others have considered a more practical setting where the complete row is a data record (resulting in *user-level privacy*). In this work, we present all our results in the strictly harder user-level privacy setting. To ensure a fair comparison, we present the results of prior works in the same setting.

1.1. Problem definition: Matrix completion

The goal of a low-rank matrix completion problem is to estimate a low-rank (or a convex relaxation of bounded nuclear norm) matrix $Y^* \in \Re^{m\times n}$, having seen only a small number of entries from it. Here, m is the number of users, and n is the number of items. Let $\Omega = \{(i,j) \subseteq [m]\times[n]\}$ be the index set of the observed entries from Y^*, and let $\mathsf{P}_\Omega : \Re^{m\times n} \to \Re^{m\times n}$ be a matrix operator s.t. $\mathsf{P}_\Omega(Y)_{ij} = Y_{ij}$ if $(i,j) \in \Omega$, and 0 otherwise. Given, $\mathsf{P}_\Omega(Y^*)$, the objective is to output a matrix Y such that the following generalization error, i.e., the error in approximating a uniformly random entry from the matrix Y^*, is minimized:

$$F(Y) = \mathbb{E}_{(i,j)\sim_{\mathsf{unif}}[m]\times[n]}\left[\left(Y_{ij} - Y^*_{ij}\right)^2\right]. \qquad (1)$$

Generalization error captures the ability of an algorithm to predict unseen samples from Y^*. We would want the generalization error to be $o(1)$ in terms of the problem parameters when $\Omega = o(mn)$. Throughout the paper, we will assume that $m > n$.

1.1.1. Our contributions

In this work, we provide the first joint DP algorithm for low-rank matrix completion with formal non-trivial error bounds, which are summarized in Tables 1 and 2. At a high level, our key result can be summarized as follows:

Informal Theorem 1.1 (Corresponds to Corollary 3.1)**.** *Assume that each entry of a hidden matrix $Y^* \in \Re^{m\times n}$ is in $[-1,1]$, and there are $\sqrt{n}$ observed entries per user. Also, assume that the nuclear norm of Y^* is bounded by $O(\sqrt{mn})$, i.e., Y^* has nearly constant rank. Then, there exist (ϵ,δ)-joint differentially private algorithms that have $o(1)$ generalization error as long as $m = \omega(n^{5/4})$.*

In other words, even with $\sqrt{n}$ observed ratings per user, we obtain asymptotically the correct estimation of each entry of Y^* on average, as long as m is large enough. The sample complexity bound dependence on m can be strengthened by making additional assumptions, such as *incoherence*, on Y^*. See the supplementary material for details.

Our algorithm is based on two important ideas: a) using local and global computation, b) using the Frank-Wolfe method as a base optimization technique.

Local and global computation: The key idea that defines our algorithm, and allows us to get strong error bounds under joint DP is splitting the algorithm into two components:

global and *local*. Recall that each row of the hidden matrix Y^* belongs to an individual user. The global component of our algorithm computes statistics that are aggregate in nature (e.g., computing the correlation across columns of the revealed matrix $\mathsf{P}_\Omega(Y^*)$). On the other hand, the local component independently fine-tunes the statistics computed by the global component to generate accurate predictions for each user. Since the global component depends on the data of all users, adding noise to it (for privacy) does not significantly affect the accuracy of the predictions. (McSherry & Mironov, 2009; Liu et al., 2015) also exploit a similar idea of segregating the computation, but they do not utilize it formally to provide non-trivial error bounds.

Frank-Wolfe based method: We use the standard nuclear norm formulation (Recht, 2011; Shalev-shwartz et al., 2011; Tewari et al., 2011; Candes & Recht, 2012) for the matrix completion problem:

$$\min_{\|Y\|_{\mathsf{nuc}} \leq k} \widehat{F}(Y), \tag{2}$$

where $\widehat{F}(Y) = \frac{1}{2|\Omega|}\|\mathsf{P}_\Omega(Y - Y^*)\|_F^2$, $\|Y\|_{\mathsf{nuc}}$ is the sum of singular values of Y, and the underlying hidden matrix Y^* is assumed to have nuclear norm of at most k. Note that we denote the empirical risk of a solution Y by $\widehat{F}(Y)$ throughout the paper. We use the popular Frank-Wolfe algorithm (Frank & Wolfe, 1956; Jaggi & Sulovsky, 2010) as our algorithmic building block. At a high-level, FW computes the solution to (2) as a convex combination of rank-one matrices, each with nuclear norm at most k. These matrices are added iteratively to the solution.

Our main contribution is to design a version of the FW method that preserves Joint DP. That is, if the standard FW algorithm decides to add matrix $u \cdot v^T$ during an iteration, our private FW computes a noisy version of $v \in \Re^n$ via its global component. Then, each user computes the respective element of $u \in \Re^m$ to obtain her update. The noisy version of v suffices for the Joint DP guarantee, and allows us to provide the strong error bound in Theorem 1.1 above.

We want to emphasize that the choice of FW as the underlying matrix completion algorithm is critical for our system. FW updates via rank-one matrices in each step. Hence, the error due to noise addition in each step is small (i.e., proportional to the rank), and allows for an easy decomposition into the local-global computation model. Other standard techniques like proximal gradient descent based techniques (Cai et al., 2010b; Lin et al., 2010) can involve nearly *full-rank* updates in an iteration, and hence might incur large error, leading to arbitrary inaccurate solutions. Note that though a prior work (Talwar et al., 2015) has proposed a DP Frank-Wolfe algorithm for high-dimensional regression, it was for a completely different problem in a different setting where the segregation of computation into global and local components was not necessary.

Private singular vector of sparse matrices using Oja's method: Our private FW requires computing a noisy covariance matrix which implies $\Omega(n^2)$ space/time complexity for n items. Naturally, such an algorithm does not scale to practical recommendation systems. In fact, this drawback exists even for standard private PCA techniques (Dwork et al., 2013). Using insights from the popular Oja's method, we provide a technique (see Algorithm 2) that has a linear dependency on n as long as the number of ratings per user is small. Moreover, the performance of our private FW method isn't affected by using this technique.

SVD-based method: In the supplementary material, we also extend our technique to a singular value decomposition (SVD) based method for matrix completion/factorization. Our utility analysis shows that there are settings where this method outperforms our FW-based method, but in general it can provide a significantly worse solution. The main goal is to study the power of the simple SVD-based method, which is still a popular method for collaborative filtering.

Empirical results: Finally, we show that along with providing strong analytical guarantees, our private FW also performs well empirically. In particular, we show its efficacy on benchmark collaborative filtering datasets like Jester (Goldberg et al., 2001), MovieLens (Harper & Konstan, 2015), the Netflix prize dataset (Bennett et al., 2007), and the Yahoo! Music recommender dataset (Yahoo, 2011). Our algorithm consistently outperforms (in terms of accuracy) the existing state-of-the-art DP matrix completion methods (SVD-based method by (McSherry & Mironov, 2009), and a variant of projected gradient descent (Cai et al., 2010c; Bassily et al., 2014b; Abadi et al., 2016)).

1.2. Comparison to prior work

As discussed earlier, our results are the first to provide non-trivial error bounds for DP matrix completion. For comparing different results, we consider the following setting of the hidden matrix $Y^* \in \Re^{m \times n}$ and the set of released entries Ω: i) $|\Omega| \approx m\sqrt{n}$, ii) each row of Y^* has an ℓ_2 norm of $\sqrt{n}$, and iii) each row of $\mathsf{P}_\Omega(Y^*)$ has ℓ_2-norm at most $n^{1/4}$, *i.e.,* $\approx \sqrt{n}$ random entries are revealed for each row. Furthermore, we assume the spectral norm of Y^* is at most $O(\sqrt{mn})$, and Y^* is rank-one. Note that these conditions are satisfied by a matrix $Y^* = u \cdot v^T$ where $u_i, v_j \in [-1, 1]$ $\forall i, j$, and $\sqrt{n}$ random entries are observed *per user*.

In Table 1, we provide a comparison based on the sample complexity, i.e., the number of users m and the number observed samples $|\Omega|$ needed to attain a generalization error of $o(1)$. We compare our results with the best non-private algorithm for matrix completion based on nuclear norm minimization (Shalev-shwartz et al., 2011), and the prior work on DP matrix completion (McSherry & Mironov, 2009; Liu et al., 2015). We see that for the same $|\Omega|$, the

sample complexity on m increases from $\omega(n)$ to $\omega(n^{5/4})$ for our FW-based algorithm. While (McSherry & Mironov, 2009; Liu et al., 2015) work under the notion of Joint DP as well, they do not provide any formal accuracy guarantees.

Algorithm	Bound on m	Bound on $\lvert\Omega\rvert$
Nuclear norm min. (non-private)*	$\omega(n)$	$\omega(m\sqrt{n})$
Noisy SVD + kNN†	–	–
Noisy SGLD (Liu et al., 2015)	–	–
Private FW (This work)	$\omega(n^{5/4})$	$\omega(m\sqrt{n})$

Table 1. Sample complexity bounds for matrix completion. $m =$ no. of users, $n =$ no. of items. The bounds hide privacy parameters ϵ and $\log(1/\delta)$, and polylog factors in m, n. References: * (Shalev-shwartz et al., 2011), † (McSherry & Mironov, 2009)

Interlude: Low-rank approximation. We also compare our results with the prior work on a related problem of DP low-rank approximation. Given a matrix $Y^* \in \Re^{m\times n}$, the goal is to compute a DP low-rank approximation Y_{priv}, s.t. Y_{priv} is close to Y^* either in the spectral or Frobenius norm. Notice that this is similar to matrix completion if the set of revealed entries Ω is the complete matrix. Hence, our methods can be applied directly. To be consistent with the existing literature, we assume that Y^* is rank-one matrix, and each row of Y^* has ℓ_2-norm at most one . Table 2 compares the various results. While all the prior works provide trivial error bounds (in both Frobenius and spectral norm, as $\|Y^*\|_2 = \|Y^*\|_F \leq \sqrt{m}$), our methods provide non-trivial bounds. The key difference is that we ensure Joint DP (Definition 2.2), while existing methods ensure the stricter standard DP (Definition 2.1), with the exponential mechanism (Kapralov & Talwar, 2013) ensuring $(\epsilon, 0)$-standard DP.

Algorithm	Error
Randomized response‡	$O(\sqrt{m+n})$
Gaussian measurement$^{\mathbb{S}}$	$O\left(\sqrt{m} + \sqrt{\mu n/m}\right)$
Noisy power method¶	$O(\sqrt{\mu})$
Exponential mechanism$^{£}$	$O(m+n)$
Private FW (This work)	$O\left(m^{3/10}n^{1/10}\right)$
Private SVD (This work)	$O\left(\sqrt{\mu\left(\frac{n^2}{m} + \frac{m}{n}\right)}\right)$

Table 2. Error bounds ($\|Y - Y^*\|_F$) for low-rank approximation. $m =$ number of users, $n =$ number of items. $\mu \in [0, m]$ is the *incoherence* parameter. The bounds hide privacy parameters ϵ and $\log(1/\delta)$, and polylog factors in m an n. Rank of the output matrix Y_{priv} is $O\left(m^{2/5}/n^{1/5}\right)$ for Private FW, whereas it is $O(1)$ for the others. References: ‡(Blum et al., 2005; Chan et al., 2011; Dwork et al., 2014b), $^{\mathbb{S}}$(Hardt & Roth, 2012), ¶(Hardt & Roth, 2013),$^{£}$(Kapralov & Talwar, 2013)

2. Background: Notions of privacy

Let $D = \{d_1, \cdots, d_m\}$ be a dataset of m entries. Each entry d_i lies in a fixed domain $\mathcal{T}$, and belongs to an individual i, whom we refer to as an *agent* in this paper. Furthermore, d_i encodes potentially sensitive information about agent i. Let $\mathcal{A}$ be an algorithm that operates on dataset D, and produces a vector of m outputs, one for each agent i and from a set of possible outputs $\mathcal{S}$. Formally, let $\mathcal{A} : \mathcal{T}^m \to \mathcal{S}^m$. Let D_{-i} denote the dataset D without the entry of the i-th agent, and similarly $\mathcal{A}_{-i}(D)$ be the set of outputs without the output for the i-th agent. Also, let $(d_i; D_{-i})$ denote the dataset obtained by adding data entry d_i to the dataset D_{-i}. In the following, we define both *standard differential privacy* and *joint differential privacy* , and contrast them.

Definition 2.1 (Standard differential privacy (Dwork et al., 2006a;b))**.** *An algorithm $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy if for any agent i, any two possible values of data entry $d_i, d_i' \in \mathcal{T}$ for agent i, any tuple of data entries for all other agents, $D_{-i} \in \mathcal{T}^{m-1}$, and any output $S \in \mathcal{S}^m$, we have* $\Pr_{\mathcal{A}}[\mathcal{A}(d_i; D_{-i}) \in S] \leq e^{\epsilon} \Pr_{\mathcal{A}}[\mathcal{A}(d_i'; D_{-i}) \in S] + \delta$.

At a high-level, an algorithm $\mathcal{A}$ is (ϵ, δ)-standard DP if for any agent i and dataset D, the output $\mathcal{A}(D)$ and D_{-i} do not reveal "much" about her data entry d_i. For reasons mentioned in Section 1, our matrix completion algorithms provide privacy guarantee based on a relaxed notion of DP, called *joint differential privacy* , which was initially proposed in (Kearns et al., 2014). At a high-level, an algorithm $\mathcal{A}$ preserves (ϵ, δ)-joint DP if for any agent i and dataset D, the output of $\mathcal{A}$ for the other $(m-1)$ agents (denoted by $\mathcal{A}_{-i}(D)$) and D_{-i} do not reveal "much" about her data entry d_i. Such a relaxation is necessary for matrix completion because an accurate completion of the row of an agent can reveal a lot of information about her data entry. However, it is still a very strong privacy guarantee for an agent even if every other agent colludes against her, as long as she does not make the predictions made to her public.

Definition 2.2 (Joint differential privacy (Kearns et al., 2014))**.** *An algorithm $\mathcal{A}$ satisfies (ϵ, δ)-joint differential privacy if for any agent i, any two possible values of data entry $d_i, d_i' \in \mathcal{T}$ for agent i, any tuple of data entries for all other agents, $D_{-i} \in \mathcal{T}^{m-1}$, and any output $S \in \mathcal{S}^{m-1}$,*

$$\Pr_{\mathcal{A}}[\mathcal{A}_{-i}(d_i; D_{-i}) \in S] \leq e^{\epsilon} \Pr_{\mathcal{A}}[\mathcal{A}_{-i}(d_i'; D_{-i}) \in S] + \delta.$$

In this paper, we consider the privacy parameter ϵ to be a small constant (≈ 0.1), and $\delta < 1/m$. There are semantic reasons for such choice of parameters (Kasiviswanathan & Smith, 2008), but that is beyond the scope of this work.

3. Private matrix completion via Frank-Wolfe

Recall that the objective is to solve the matrix completion problem (defined in Section 1.1) under Joint DP. A standard modeling assumption is that Y^* is nearly low-rank, leading to the following empirical risk minimization problem (Keshavan et al., 2010; Jain et al., 2013; Jin et al., 2016): $\min_{\mathsf{rank}(Y)\le k} \underbrace{\frac{1}{2|\Omega|}\|\mathsf{P}_\Omega(Y-Y^*)\|_F^2}_{\widehat{F}(Y)}$, where $k \ll \min(m,n)$.

As this is a challenging non-convex optimization problem, a popular approach is to relax the rank constraint to a nuclear-norm constraint, i.e., $\min_{\|Y\|_{\mathsf{nuc}}\le k} \widehat{F}(Y)$.

To this end, we use the FW algorithm (see the supplementary material for more details) as our building block. FW is a popular conditional gradient algorithm in which the current iterate is updated as: $Y^{(t)} \leftarrow (1-\eta)Y^{(t-1)} + \eta \cdot G$, where η is the step size, and G is given by: $\underset{\|G\|_{\mathsf{nuc}}\le k}{\mathsf{argmin}} \left\langle G, \nabla_{Y^{(t-1)}}\widehat{F}(Y)\right\rangle$. Note that the optimal solution to the above problem is $G = -k\mathbf{u}\mathbf{v}^\top$, where $(\lambda, \mathbf{u}, \mathbf{v})$ are the top singular components of $A^{(t-1)} = \mathsf{P}_\Omega(Y^{(t-1)} - Y^*)$. Also, the optimal G is a rank-one matrix.

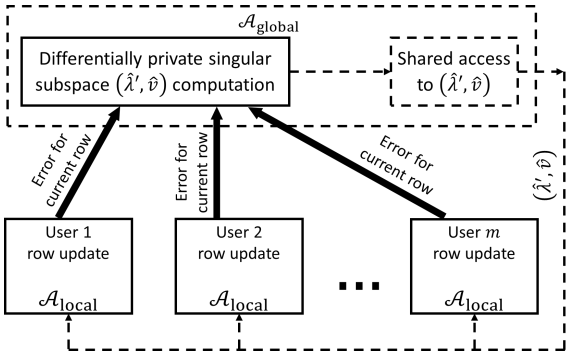

Figure 1. Block schematic describing the two functions $\mathcal{A}_{\mathsf{local}}$ and $\mathcal{A}_{\mathsf{global}}$ of Algorithm 1. The solid boxes and arrows represent computations that are privileged and without external access, and the dotted boxes and arrows represent the unprivileged computations.

Algorithmic ideas: In order ensure Joint DP and still have strong error guarantees, we develop the following ideas. These ideas have been formally compiled into Algorithm 1. Notice that both the functions $\mathcal{A}_{\mathsf{global}}$ and $\mathcal{A}_{\mathsf{local}}$ in Algorithm 1 are parts of the Private FW technique, where $\mathcal{A}_{\mathsf{global}}$ consists of the global component, and each user runs $\mathcal{A}_{\mathsf{local}}$ at her end to carry out a local update. Throughout this discussion, we assume that $\max_{i\in[m]} \|\mathsf{P}_\Omega(Y_i^*)\|_2 \le L$.

Splitting the update into global and local components: One can equivalently write the Frank-Wolfe update as follows: $Y^{(t)} \leftarrow (1-\eta)Y^{(t-1)} - \eta\cdot\frac{k}{\lambda}A^{(t-1)}\mathbf{v}\mathbf{v}^\top$, where $A^{(t-1)}, \mathbf{v}$, and λ are defined as above. Note that $\mathbf{v}$ and λ^2 can also be obtained as the top right eigenvector and eigenvalue of ${A^{(t-1)}}^\top A^{(t-1)} = \sum_{i=1}^m {A_i^{(t-1)}}^\top A_i^{(t-1)}$, where $A_i^{(t-1)} = \mathsf{P}_\Omega(Y_i^{(t-1)} - Y_i^*)$ is the i-th row of $A^{(t-1)}$. We will use the *global component* $\mathcal{A}_{\mathsf{global}}$ in Algorithm 1 to compute $\mathbf{v}$ and λ. Using the output of $\mathcal{A}_{\mathsf{global}}$, each user (row) $i \in [m]$ can compute her *local update* (using $\mathcal{A}_{\mathsf{local}}$) as follows:

$$Y_i^{(t)} = (1-\eta)Y_i^{(t-1)} - \frac{\eta k}{\lambda}\mathsf{P}_\Omega(Y^{(t-1)} - Y^*)_i\mathbf{v}\mathbf{v}^\top. \quad (3)$$

A block schematic of this idea is presented in Figure 1.

Algorithm 1 Private Frank-Wolfe algorithm

function Global Component $\mathcal{A}_{\mathsf{global}}$ **(Input-** privacy parameters: (ϵ,δ) s.t. $\epsilon \le 2\log(1/\delta)$, total number of iterations: T, bound on $\|\mathsf{P}_\Omega(Y_i^*)\|_2$: L, failure probability: β, number of users: m, number of items: n**)**
  $\sigma \leftarrow L^2\sqrt{64\cdot T\log(1/\delta)}/\epsilon$, $\widehat{\mathbf{v}} \leftarrow \{0\}^n$, $\widehat{\lambda} \leftarrow 0$
  for $t \in [T]$ **do**
    $W^{(t)} \leftarrow \{0\}^{n\times n}$, $\widehat{\lambda}' \leftarrow \widehat{\lambda} + \sqrt{\sigma\log(n/\beta)}n^{1/4}$
    for $i \in [m]$ **do** $W^{(t)} \leftarrow W^{(t)} + \mathcal{A}_{\mathsf{local}}(i, \widehat{\mathbf{v}}, \widehat{\lambda}', T, t, L)$
    $\widehat{W}^{(t)} \leftarrow W^{(t)} + N^{(t)}$, where $N^{(t)} \in \Re^{n\times n}$ is a matrix with i.i.d. entries from $\mathcal{N}(0,\sigma^2)$
    $(\widehat{\mathbf{v}}, \widehat{\lambda}^2) \leftarrow$ Top eigenvector and eigenvalue of $\widehat{W}^{(t)}$
  end for
end function
function Local Update $\mathcal{A}_{\mathsf{local}}$ **(Input-** user number: i, top right singular vector: $\widehat{\mathbf{v}}$, top singular value: $\widehat{\lambda}'$, total number of iterations: T, current iteration: t, bound on $\|\mathsf{P}_\Omega(Y_i^*)\|_2$: L, private true matrix row: $\mathsf{P}_\Omega(Y_i^*)$**)**
  $Y_i^{(0)} \leftarrow \{0\}^n$, $A_i^{(t-1)} \leftarrow \mathsf{P}_\Omega(Y_i^{(t-1)} - Y_i^*)$
  $\widehat{u}_i \leftarrow (A_i^{(t-1)}\cdot\widehat{\mathbf{v}})/\widehat{\lambda}'$
  Define $\Pi_{L,\Omega}(M)_{i,j} = \min\left\{\frac{L}{\|\mathsf{P}_\Omega(M_i)\|_2}, 1\right\}\cdot M_{i,j}$
  $Y_i^{(t)} \leftarrow \Pi_{L,\Omega}\left(\left(1-\frac{1}{T}\right)Y_i^{(t-1)} - \frac{k}{T}\widehat{u}_i(\widehat{\mathbf{v}})^T\right)$
  $A_i^{(t)} \leftarrow \mathsf{P}_\Omega\left(Y_i^{(t)} - Y_i^*\right)$
  if $t = T$, Output $Y_i^{(T)}$ as prediction to user i and **stop**
  else Return ${A_i^{(t)}}^\top A_i^{(t)}$ to $\mathcal{A}_{\mathsf{global}}$
end function

Noisy rank-one update: Observe that $\mathbf{v}$ and λ, the statistics computed in each iteration of $\mathcal{A}_{\mathsf{global}}$, are aggregate statistics that use information from all rows of Y^*. This ensures that they are noise tolerant. Hence, adding sufficient noise can ensure standard DP (Definition 2.1) for $\mathcal{A}_{\mathsf{global}}$. [2] Since

[2] The second term in computing $\widehat{\lambda}'$ in Algorithm 1 is due to a bound on the spectral norm of the Gaussian noise matrix. We use this bound to control the error introduced in the computation of $\hat{\lambda}$.

the final objective is to satisfy Joint DP (Definition 2.2), the local component $\mathcal{A}_{\mathsf{local}}$ can compute the update for each user (corresponding to (3)) without adding any noise.

Controlling norm via projection: In order to control the amount of noise needed to ensure DP, any individual data entry (here, any row of Y^*) should have a bounded effect on the aggregate statistic computed by $\mathcal{A}_{\mathsf{global}}$. However, each intermediate computation ${Y_i}^{(t)}$ in (3) can have high ℓ_2-norm even if $\|\mathsf{P}_\Omega(Y_i^*)\|_2 \leq L$. We address this by applying a projection operator $\Pi_{L,\Omega}$ (defined below) to ${Y_i}^{(t)}$, and compute the local update as $\Pi_{L,\Omega}\left({Y_i}^{(t)}\right)$ in place of (3). $\Pi_{L,\Omega}$ is defined as follows: For any matrix M, $\Pi_{L,\Omega}$ ensures that any row of the "zeroed out" matrix $\mathsf{P}_\Omega(M)$ does not have ℓ_2-norm higher than L. Formally, $\Pi_{L,\Omega}(M)_{i,j} = \min\left\{\frac{L}{\|\mathsf{P}_\Omega(M_i)\|_2}, 1\right\} \cdot M_{i,j}$ for all entries (i,j) of M. In our analysis, we show that this projection operation does not increase the error.

3.1. Privacy and utility analysis

Theorem 3.1. *Algorithm 1 satisfies (ϵ, δ)-joint DP.*

We defer the proof to the supplementary material. The proof uses standard DP properties of Gaussian noise addition from (Bun & Steinke, 2016). The requirement $\epsilon \leq 2\log(1/\delta)$ in the input of Algorithm 1 is due to a reduction of a Concentrated DP guarantee to a standard DP guarantee. We now show that the empirical risk of our algorithm is close to the optimal as long as the number of users m is "large".

Theorem 3.2 (Excess empirical risk guarantee)**.** *Let Y^* be a matrix with $\|Y^*\|_{\mathsf{nuc}} \leq k$, and $\max_{i\in[m]} \|\mathsf{P}_\Omega(Y^*)_i\|_2 \leq L$. Let $Y^{(T)}$ be a matrix, with its rows being ${Y_i}^{(T)}$ for all $i \in [m]$, computed by function $\mathcal{A}_{\mathsf{local}}$ in Algorithm 1 after T iterations. If $\epsilon \leq 2\log\left(\frac{1}{\delta}\right)$, then with probability at least 2/3 over the outcomes of Algorithm 1, the following is true:*

$$\widehat{F}\left(Y^{(T)}\right) = O\left(\frac{k^2}{|\Omega|T} + \frac{kT^{1/4}L\sqrt{n^{1/2}\log^{1/2}(1/\delta)\log n}}{|\Omega|\sqrt{\epsilon}}\right).$$

Furthermore, if $T = \tilde{O}\left(\frac{k^{4/5}\epsilon^{2/5}}{n^{1/5}L^{4/5}}\right)$, then $\widehat{F}\left(Y^{(T)}\right) = \tilde{O}\left(\frac{k^{6/5}n^{1/5}L^{4/5}}{|\Omega|\epsilon^{2/5}}\right)$ after hiding poly-logarithmic terms.

We defer the proof to the supplementary material. At a high-level, our proof combines the noisy eigenvector estimation error for Algorithm 1 with a noisy-gradient analysis of the FW algorithm. Also, note that the first term in the bound corresponds to the standard FW convergence error, while the second term can be attributed to the noise added for DP which directly depends on T. We also compute the optimal number of iterations required to minimize the empirical risk. Finally, the rank of $Y^{(T)}$ is at most T, but its nuclear-norm is bounded by k. As a result, $Y^{(T)}$ has low *generalization error* (see Section 3.1.1).

Remark 1. We further illustrate our empirical risk bound by considering a simple setting: let Y^* be a rank-one matrix with $Y^*_{ij} \in [-1,1]$ and $|\Omega| = m\sqrt{n}$. Then $k = O(\sqrt{mn})$, and $L = O(n^{1/4})$, implying an error of $\widetilde{O}\left(\sqrt{n}m^{-2/5}\right)$ hiding the privacy parameter ϵ; in contrast, a trivial solution like $Y = 0$ leads to $O(1)$ error. Naturally, the error increases with n as there is more information to be protected. However, it decreases with a larger number of users m as the presence/absence of a user has lesser effect on the solution with increasing m. We leave further investigation into the dependency of the error on m for future work.

Remark 2. Our analysis does not require an upper bound on the nuclear norm of Y^* (as stated in Theorem 3.2); we would instead incur an additional error of $\min_{\|Y\|_{\mathsf{nuc}}\leq k} \frac{1}{|\Omega|}\|\mathsf{P}_\Omega(Y^* - Y)\|_F^2$. Moreover, consider a similar scenario as in Remark 1, but $|\Omega| = mn$, i.e., all the entries of Y^* are revealed. In such a case, $L = O(\sqrt{n})$, and the problem reduces to that of standard low-rank matrix approximation of Y^*. Note that our result here leads to an error bound of $\widetilde{O}\left(n^{1/5}m^{-2/5}\right)$, while the state-of-the-art result by (Hardt & Roth, 2013) leads to an error bound of $O(1)$ due to being in the much stricter standard DP model.

3.1.1. Generalization error guarantee

We now present a generalization error (defined in Equation 1) bound which shows that our approach provides accurate prediction over *unknown* entries. For obtaining our bound, we use Theorem 1 from (Srebro & Shraibman, 2005) (provided in the supplementary material for reference). Also, the output of Private FW (Algorithm 1) has rank at most T, where T is the number of iterations. Thus, replacing T from Theorem 3.2, we get the following:

Corollary 3.1 (Generalization Error)**.** *Let $\|Y^*\|_{\mathsf{nuc}} \leq k$ for a hidden matrix Y^*, and $\|\mathsf{P}_\Omega(Y_i^*)\|_2 \leq L$ for every row i of Y^*. If we choose the number of rounds in Algorithm 1 to be $O\left(\frac{k^{4/3}}{(|\Omega|(m+n))^{1/3}}\right)$, the data samples in Ω are drawn u.a.r. from $[m]\times[n]$, and $\epsilon \leq 2\log\left(\frac{1}{\delta}\right)$, then with probability at least 2/3 over the outcomes of the algorithm and choosing Ω, the following is true for the final completed matrix Y:*

$$F(Y) = \tilde{O}\left(\frac{k^{4/3}Ln^{1/4}}{\sqrt{\epsilon|\Omega|^{13/6}(m+n)^{1/6}}} + \left(\frac{k\sqrt{m+n}}{|\Omega|}\right)^{2/3}\right).$$

The $\tilde{O}(\cdot)$ hides poly-logarithmic terms in $m, n, |\Omega|$ and δ.

Remark 3. We further illustrate our bound using a setting similar to the one considered in Remark 1. Let Y^* be a rank-one matrix with $Y^*_{ij} \in [-1,1]$ for all i,j; let $|\Omega| \geq m\sqrt{n}\cdot\mathsf{polylog}(n)$, i.e., the fraction of movies rated by each user is arbitrarily small for larger n. For this setting, our generalization error is $o(1)$ for $m = \omega(n^{5/4})$.

This is slightly higher than the bound in the non-private setting by (Shalev-shwartz et al., 2011), where $m = \omega(n)$ is sufficient to get generalization error $o(1)$. Also, as the first term in the error bound pertains to DP, it decreases with a larger number of users m, and increases with n as it has to preserve privacy of a larger number of items. In contrast, the second term is the matrix completion error decreases with n. This is intuitive, as a larger number of movies enables more sharing of information between users, thus allowing a better estimation of preferences Y^*. However, just increasing m may not always lead to a more accurate solution (for example, consider the case of $n = 1$).

Remark 4. The guarantee in Corollary 3.1 is for uniformly random Ω, but using the results of (Shamir & Shalev-Shwartz, 2011), it is straightforward to extend our results to any i.i.d. distribution over Ω. Moreover, we can extend our results to handle strongly convex and smooth loss functions instead of the squared loss considered in this paper.

3.2. Efficient PCA via Oja's Algorithm

Algorithm 1 requires computing the top eigenvector of $\widehat{W^{(t)}} = W^{(t)} + N^{(t)}$, where $W^{(t)} = \sum_i \left(A_i{}^{(t)}\right)^\top A_i{}^{(t)}$ and $N^{(t)}$ is a random noise matrix. However, this can be a bottleneck for computation as $N^{(t)}$ itself is a dense $n \times n$ matrix, implying a space complexity of $\Omega(n^2 + mk)$, where k is the maximum number of ratings provided by a user. Similarly, standard eigenvector computation algorithms will require $O(mk^2 + n^2)$ time (ignoring factors relating to rate of convergence), which can be prohibitive for practical recommendation systems with large n. We would like to stress that this issue plagues even standard DP PCA algorithms (Dwork et al., 2013), which have quadratic space-time complexity in the number of dimensions.

We tackle this by using a stochastic algorithm for the top eigenvector computation that significantly reduces both space and time complexity while preserving privacy. In particular, we use Oja's algorithm (Jain et al., 2016), which computes top eigenvectors of a matrix with a stochastic access to the matrix itself. That is, if we want to compute the top eigenvector of $W^{(t)}$, we can use the following updates:

$$\widehat{\mathbf{v}}_\tau = (I + \eta X_\tau)\widehat{\mathbf{v}}_{\tau-\mathbf{1}}, \qquad \widehat{\mathbf{v}}_\tau = \widehat{\mathbf{v}}_\tau / \|\widehat{\mathbf{v}}_\tau\|_2 \qquad (4)$$

where $\mathbb{E}[X_\tau] = W^{(t)}$. For example, we can update $\widehat{\mathbf{v}}_\tau$ using $X_\tau = W^{(t)} + N_\tau^{(t)}$ where each entry of $N_\tau^{(t)}$ is sampled i.i.d. from a Gaussian distribution calibrated to ensure DP. Even this algorithm in its current form does not decrease the space or time complexity as we need to generate a dense matrix $N_\tau{}^{(t)}$ in each iteration. However, by observing that $N_\tau{}^{(t)} v = g_\tau \sim \mathcal{N}(0, \sigma^2 \mathbf{1}^n)$ where v is independent of $N_\tau{}^{(t)}$, we can now replace the generation of $N_\tau{}^{(t)}$ by the generation of a vector g_τ, thus reducing both the space and time complexity of our algorithm. The computation of each update is significantly *cheaper* as long as $mk \ll n^2$, which is the case for practical recommendation systems as k tends to be fairly small there (typically on the order of $\sqrt{n}$).

Algorithm 2 Private Oja's algorithm

Input: An $m \times n$ matrix A s.t. each row $\|A_i\|_2 \le L$, privacy parameters: (ϵ, δ) s.t. $\epsilon \le 2\log(1/\delta)$, total number of iterations: Γ
$\sigma \leftarrow L^2\sqrt{256 \cdot \Gamma \log(2/\delta)}/\epsilon, \widehat{\mathbf{v}}_\mathbf{0} \sim \mathcal{N}(0, \sigma^2 I)$
for $\tau \in [\Gamma]$ **do**
  $\eta = \frac{1}{\Gamma\sigma\sqrt{n}}, g_\tau \sim \mathcal{N}(0, \sigma^2 \mathbf{1}^n)$
  $\widehat{\mathbf{v}}_\tau \leftarrow \widehat{\mathbf{v}}_{\tau-\mathbf{1}} + \eta\left(A^T A \widehat{\mathbf{v}}_{\tau-\mathbf{1}} + g_\tau\right), \widehat{\mathbf{v}}_\tau \leftarrow \widehat{\mathbf{v}}_\tau / \|\widehat{\mathbf{v}}_\tau\|_2$
end for
Return $\widehat{\mathbf{v}}_\mathbf{\Gamma}, \left(\widehat{\lambda}_\Gamma^2 \leftarrow ||A \cdot \widehat{\mathbf{v}}_\mathbf{\Gamma}||_2^2 + \mathcal{N}(0, \sigma^2)\right)$

Algorithm 2 provides a pseudocode of the eigenvector computation method. The computation of the approximate eigenvector $\widehat{\mathbf{v}}_\mathbf{\Gamma}$ and the eigenvalue $\widehat{\lambda}_\Gamma^2$ in it is DP (directly follows via the proof of Theorem 3.1). The next natural question is how well can $\widehat{\mathbf{v}}_\mathbf{\Gamma}$ approximate the behavior of the top eigenvector of the non-private covariance matrix $W^{(t)}$? To this end, we provide Theorem 3.3 below that analyzes Oja's algorithm, and shows that the *Rayleigh quotient* of the approximate eigenvector is close to the top eigenvalue of $W^{(t)}$. In particular, using Theorem 3.3 along with the fact that in our case, $\mathcal{V} = \sigma^2 n$, we have $\left\|A^{(t)}\right\|_2^2 \le \|A^{(t)}\widehat{\mathbf{v}}_\mathbf{\Gamma}\|_2^2 + O\left(\sigma\sqrt{n}\log(\eta/\beta)\right)$ with high probability (w.p. $\ge 1 - \beta^2$)), where $\widehat{\mathbf{v}}_\mathbf{\Gamma}$ is the output of Algorithm 2, $\Gamma = \Omega\left(\min\left\{\frac{1}{\beta}, \frac{\|A^{(t)}\|^2}{\sigma\sqrt{n}}\right\}\right)$, and $\eta = \frac{1}{\Gamma \cdot \sigma\sqrt{n}}$.

Note that the above given bound is exactly the bound required in the proof of Theorem 3.2. Hence, computing the top eigenvector privately using Algorithm 2 does not change the utility bound of Theorem 3.2.

Theorem 3.3 (Based on Theorem 3 (Allen-Zhu & Li, 2017))**.** *Let $X_1, X_2, \ldots X_\Gamma$ be sampled i.i.d. such that $\mathbb{E}X_i = W = A^T A$. Moreover, let $\mathcal{V} = \max\{\|\mathbb{E}(X_i - W)^T(X_i - W)\|, \|\mathbb{E}(X_i - W)(X_i - W)^T\|\}$, and $\eta = \frac{1}{\sqrt{\mathcal{V}\Gamma}}$. Then, the Γ-th iterate of Oja's Algorithm (Update (4)) , i.e., $\widehat{\mathbf{v}}_\mathbf{\Gamma}$, satisfies (w.p. $\ge 1 - 1/\mathsf{poly}(\Gamma)$):* $\widehat{\mathbf{v}}_\mathbf{\Gamma}^T W \widehat{\mathbf{v}}_\mathbf{\Gamma} \ge \|W\|_2 - O\left(\sqrt{\frac{\mathcal{V}}{\Gamma}} + \frac{\|W\|_2}{\Gamma}\right)$.

Comparison with Private Power Iteration (PPI) method (Hardt & Roth, 2013): Private PCA via PPI provides utility guarantees dependent on the gap between the top and the kth eigenvalue of the input matrix A for some $k > 1$, whereas private Oja's utility guarantee is gap-independent.

4. Experimental evaluation

We now present empirical results for Private FW (Algorithm 1) on several benchmark datasets, and compare its

performance to state-of-the-art methods like (McSherry & Mironov, 2009), and private as well as non-private variant of the Projected Gradient Descent (PGD) method (Cai et al., 2010c; Bassily et al., 2014a; Abadi et al., 2016). In all our experiments, we see that private FW provides accuracy very close to that of the non-private baseline, and almost always significantly outperforms both the private baselines.

Datasets: As we want to preserve privacy of every user, and the output for each user is n-dimensional, we can expect the private recommendations to be accurate only when $m \gg n$ (see Theorem 3.1). Due to this constraint, we conduct experiments on the following datasets: 1) *Synthetic:* We generate a random rank-one matrix $Y^* = uv^T$ with unit ℓ_∞-norm, $m = 500$K, and $n = 400$, 2) *Jester:* This dataset contains $n = 100$ jokes, and $m \approx 73$K users, 3) *MovieLens10M (Top 400):* We pick the $n = 400$ most rated movies from the Movielens10M dataset, resulting in $m \approx 70$K users, 4) *Netflix (Top 400):* We pick the $n = 400$ most rated movies from the Netflix prize dataset, resulting in $m \approx 474$K users, and 5) *Yahoo! Music (Top 400):* We pick the $n = 400$ most rated songs from the Yahoo! music dataset, resulting in $m \approx 995$K users.[3] We rescale the ratings to be from 0 to 5 for Jester and Yahoo! Music.

Procedure: For all datasets, we randomly sample 1% of the given ratings for measuring the test error. For experiments with privacy, for all datasets except Jester, we randomly select at most $\xi = 80$ ratings per user to get $\mathsf{P}_\Omega(Y^*)$. We vary the privacy parameter $\epsilon \in [0.1, 5]$ [4], but keep $\delta = 10^{-6}$, thus ensuring that $\delta < \frac{1}{m}$ for all datasets. Moreover, we report results averaged over 10 independent runs.

Note that the privacy guarantee is user-level, which effectively translates to an entry-level guarantee of $\epsilon_{entry} = \frac{\epsilon_{user}}{\xi}$, i.e., $\epsilon_{entry} \in [0.00125, 0.0625]$ as $\epsilon_{user} \in [0.1, 5]$.

Non-private baseline: We find that non-private FW and non-private PGD converge to the same accuracy after tuning, and hence, we use this as our baseline.

Private baselines: To the best of our knowledge, only (McSherry & Mironov, 2009) and (Liu et al., 2015) address the user-level DP matrix completion problem. While we present an empirical evaluation of the 'SVD after cleansing method' from the former, we refrain from comparing to the latter [5]. We also provide a comparison with private PGD (pseudocode provided in the supplementary material).

We elaborate on the data normalization and the parameter choices for all algorithms in the supplementary material.

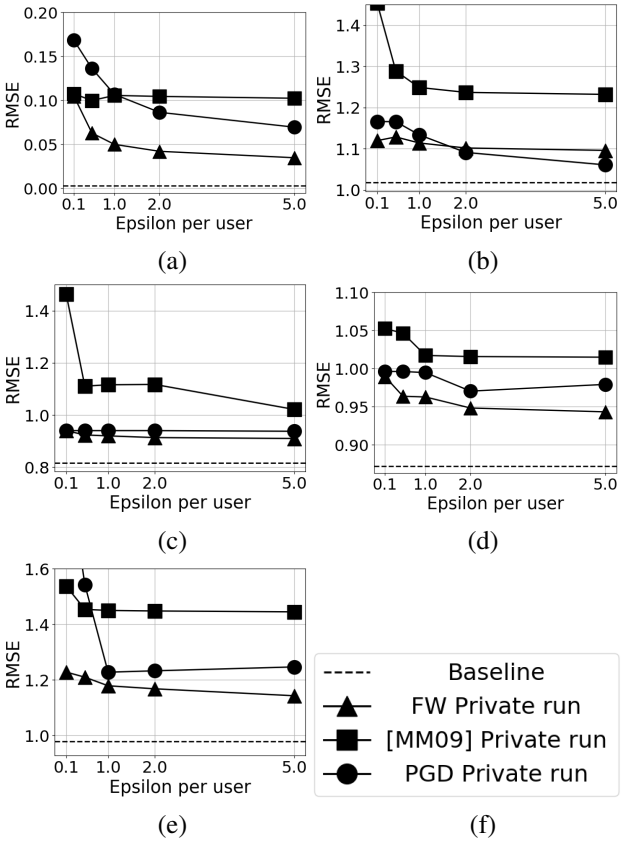

Figure 2. Root mean squared error (RMSE) vs. ϵ, on (a) synthetic, (b) Jester, (c) MovieLens10M, (d) Netflix, and (e) Yahoo! Music datasets, for $\delta = 10^{-6}$. A legend for all the plots is given in (f).

Results: Figure 2 shows the results of our experiments[6]. Even though all the considered private algorithms satisfy Joint DP, our private FW method almost always incurs a significantly lower test RMSE than the two private baselines. Note that although non-private PGD provides similar empirical accuracy as non-private FW, the difference in performance for their private versions can be attributed to the noise being calibrated to a rank-one update for our private Frank-Wolfe.

[3] For $n = 900$ with all the considered datasets (except Jester), we see that private PGD takes too long to complete; we present an evaluation for the other algorithms in the supplementary material.

[4] The requirement in Algorithm 1 that $\epsilon \leq 2\log(1/\delta)$ is satisfied by all the values of ϵ considered for the experiments.

[5] The exact privacy parameters (ϵ and δ) for the Stochastic Gradient Langevin Dynamics based algorithm in (Liu et al., 2015) (correspondigly, in (Wang et al., 2015)) are unclear. They use a Markov chain based sampling method; to obtain quantifiable (ϵ, δ), the sampled distribution is required to converge (non-asymptotically) to a DP preserving distribution in ℓ_1 distance, for which we are not aware of any analysis.

[6] In all our experiments, the implementation of private FW with Oja's method (Algorithm 2) did not suffer any perceivable loss of accuracy as compared to the variant in Algorithm 1; all the plots in Figure 2 remain identical.

Acknowledgements

The authors would like to thank Ilya Mironov, and the anonymous reviewers, for their helpful comments. This material is in part based upon work supported by NSF grants CCF-1740850 and IIS-1447700, and a grant from the Sloan foundation. The full version of this work is available at `https://arxiv.org/abs/1712.09765`.

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*, CCS '16, pp. 308–318, New York, NY, USA, 2016. ACM. doi: 10.1145/2976749.2978318.

Allen-Zhu, Z. and Li, Y. First efficient convergence for streaming k-pca: A global, gap-free, and near-optimal rate. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pp. 487–492, 2017.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pp. 464–473. IEEE, 2014a.

Bassily, R., Smith, A. D., and Thakurta, A. Private empirical risk minimization, revisited. *CoRR*, abs/1405.7085, 2014b.

Bennett, J., Lanning, S., and Netflix, N. The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*, 2007.

Blum, A., Dwork, C., McSherry, F., and Nissim, K. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 128–138. ACM, 2005.

Bun, M. and Steinke, T. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *TCC*, 2016.

Cai, J., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2010a.

Cai, J., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010b.

Cai, J.-F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2010c.

Calandrino, J. A., Kilzer, A., Narayanan, A., Felten, E. W., and Shmatikov, V. "you might also like": Privacy risks of collaborative filtering. In *IEEE Symposium on Security and Privacy*, 2011.

Candes, E. and Recht, B. Exact matrix completion via convex optimization. *Communications of the ACM*, 2012.

Chan, T.-H. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3): 26, 2011.

Clarkson, K. L. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 2010.

Dinur, I. and Nissim, K. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, pp. 202–210, 2003. doi: 10.1145/773153.773173.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pp. 265–284. Springer, 2006b.

Dwork, C., Talwar, K., Thakurta, A., and Zhang, L. Randomized response strikes back: Private singular subspace computation with (nearly) optimal error guarantees. 2013.

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

Dwork, C., Talwar, K., Thakurta, A., and Zhang, L. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pp. 11–20. ACM, 2014b.

Erlingsson, Ú., Pihur, V., and Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *CCS*, 2014.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2): 95–110, 1956.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. Eigentaste: A constant time collaborative filtering algorithm. *Inf. Retr.*, 4(2):133–151, July 2001. ISSN 1386-4564. doi: 10.1023/A:1011419012209.

Hardt, M. and Roth, A. Beating randomized response on incoherent matrices. In *STOC*, 2012.

Hardt, M. and Roth, A. Beyond worst-case analysis in private singular vector computation. In *STOC*, 2013.

Hardt, M. and Wootters, M. Fast matrix completion without the condition number. In *COLT*, 2014.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 2015.

Jaggi, M. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML*, pp. 427–435, 2013.

Jaggi, M. and Sulovsky, M. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010.

Jain, P., Meka, R., and Dhillon, I. S. Guaranteed rank minimization via singular value projection. In *NIPS*, 2010.

Jain, P., Netrapalli, P., and Sanghavi, S. Low-rank matrix completion using alternating minimization. In *STOC*, 2013.

Jain, P., Jin, C., Kakade, S. M., Netrapalli, P., and Sidford, A. Streaming pca: Matching matrix bernstein and near-optimal finite sample guarantees for ojas algorithm. In *Conference on Learning Theory*, pp. 1147–1164, 2016.

Jin, C., Kakade, S. M., and Netrapalli, P. Provable efficient online matrix completion via non-convex stochastic gradient descent. In *NIPS*, 2016.

Kapralov, M. and Talwar, K. On differentially private low rank approximation. In *SODA*, 2013.

Kasiviswanathan, S. P. and Smith, A. A note on differential privacy: Defining resistance to arbitrary side information. *CoRR*, arXiv:0803.39461 [cs.CR], 2008.

Kearns, M., Pai, M., Roth, A., and Ullman, J. Mechanism design in large games: Incentives and privacy. In *ITCS*, 2014.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 2010.

Koren, Y. and Bell, R. M. Advances in collaborative filtering. In *Recommender Systems Handbook*, pp. 77–118. Springer US, 2015.

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

Lin, Z., Chen, M., and Ma, Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *CoRR*, abs/1009.5055, 2010.

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

McMillan, R. Apple tries to peek at user habits without violating privacy. *The Wall Street Journal*, 2016.

McSherry, F. and Mironov, I. Differentially private recommender systems: building privacy into the net. In *Symp. Knowledge Discovery and Datamining (KDD)*, pp. 627–636. ACM New York, NY, USA, 2009.

Narayanan, A. and Shmatikov, V. Myths and fallacies of "personally identifiable information". *Commun. ACM*, 53(6):24–26, 2010.

Recht, B. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 2011.

Shalev-shwartz, S., Gonen, A., and Shamir, O. Large-scale convex minimization with a low-rank constraint. In Getoor, L. and Scheffer, T. (eds.), *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 329–336, New York, NY, USA, 2011. ACM.

Shamir, O. and Shalev-Shwartz, S. Collaborative filtering with the trace norm: Learning, bounding, and transducing. In *COLT*, 2011.

Srebro, N. and Shraibman, A. Rank, trace-norm and max-norm. In *International Conference on Computational Learning Theory*, 2005.

Talwar, K., Thakurta, A., and Zhang, L. Nearly optimal private lasso. In *NIPS*, 2015.

Tao, T. *Topics in random matrix theory*, volume 132. American Mathematical Society, 2012.

Tewari, A., Ravikumar, P. K., and Dhillon, I. S. Greedy algorithms for structurally constrained high dimensional problems. In *NIPS*, 2011.

Wang, Y.-X., Fienberg, S., and Smola, A. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015.

Yahoo. C15 - yahoo! music user ratings of musical tracks, albums, artists and genres, version 1.0. *Webscope*, 2011.

Yu, H.-F., Jain, P., Kar, P., and Dhillon, I. Large-scale multi-label learning with missing labels. In *ICML*, 2014.