Analyze Gauss: Optimal Bounds for Privacy-Preserving Principal Component Analysis

Cynthia Dwork
Microsoft Research

Kunal Talwar
Microsoft Research

Abhradeep Thakurta*
Stanford University
Microsoft Research

Li Zhang
Microsoft Research

ABSTRACT

We consider the problem of privately releasing a low dimensional approximation to a set of data records, represented as a matrix A in which each row corresponds to an individual and each column to an attribute. Our goal is to compute a subspace that captures the covariance of A as much as possible, classically known as principal component analysis (PCA). We assume that each row of A has ℓ_2 norm bounded by one, and the privacy guarantee is defined with respect to addition or removal of any single row. We show that the well-known, but misnamed, randomized response algorithm, with properly tuned parameters, provides nearly optimal additive quality gap compared to the best possible singular subspace of A. We further show that when A^TA has a large eigenvalue gap – a reason often cited for PCA – the quality improves significantly. Optimality (up to logarithmic factors) is proved using techniques inspired by the recent work of Bun, Ullman, and Vadhan on applying Tardos's fingerprinting codes to the construction of hard instances for private mechanisms for 1-way marginal queries. Along the way we define a *list culling game* which may be of independent interest.

By combining the randomized response mechanism with the well-known *following the perturbed leader* algorithm of Kalai and Vempala we obtain a private online algorithm with nearly optimal regret. The regret of our algorithm even outperforms all the previously known online *non-private* algorithms of this type. We achieve this better bound by, satisfyingly, borrowing insights and tools from differential privacy!

1. INTRODUCTION

In areas as diverse as machine learning, statistics, information retrieval, earth sciences, archaeology, and image processing, given a data set represented by a matrix $A \in \mathbb{R}^{m \times n}$, it is often desirable to find a good approximation to A that has low rank. Working with low-rank approximations improves space and time efficiency. Other benefits include removal of noise and extraction of correlations, useful, for example, in (approximate) matrix completion from a small set of observations – an impossible task if A is arbitrary but potentially feasible if A enjoys a good low rank approximation. The problem of low-rank approximation has also received substantial attention in the differential privacy literature [4, 13, 24, 21, 9, 17, 18]. If we think of the matrix $A \in \mathbb{R}^{m \times n}$ as containing information about n attributes of m individuals, the goal is to learn "about" A (we intentionally remain vague, for now) without compromising the privacy of any individual. That is, the literature focuses on being able to do, in a differentially private way, whatever is achieved by low-rank approximation in the non-private literature. Our work continues this line of research.

Existing differentially private algorithms can have errors with an unfortunate dependence on the ambient dimension n of the data. This bad dependence may sometimes be due to the suboptimality of our algorithms, sometimes due to the inherent difficulty of the problem. A driving motivation for our work is to extract better performance from these algorithms when the inherent dimensionality of the input is much lower than the ambient dimension. For example, the data may be generated according to a low dimensional model and the measurements may be noisy.

The standard method of the principal component analysis (PCA) for low rank approximation is to compute a best low-dimensional eigen-subspace B of the matrix $A^TA = \sum_{i=1}^{m} a_i^T a_i$ (recall that the a_i are row vectors). The underlying intuition is that the projection onto B preserves the important features of the data rows while projecting away the noise. We will focus on a private mechanism for computing B. By (1) privately finding a low-rank subspace B capturing most of the variance in A, and then (2) running the existing differentially private algorithm on the projection of A onto B, the hope is that poor dependence on the dimension in the second step is mitigated by the dimension reduction obtained in the first.

Because it was found in a privacy-preserving fashion, B can safely be made public. A key point is that the two-step procedure just described does not require publication of the projection. This, then, will be our approach: the *projector* (Π_B) will be public, the *projection* ($\Pi_B(A)$) will not be released. [1].

The literature sometimes focuses on the case of $m \gg n$, and at other times assumes $m \ll n$. In the first case, the rows of the data matrix are often assumed to be normalized to have norm at most 1, as is done here; when $m \ll n$ the row norms may be unbounded [17, 18]. The literature also varies in terms of granu-

*Supported in part by the Sloan Foundation

STOC'14 May 31 - June 03 2014, New York, NY, USA

[1] This was exploited by McSherry and Mironov in their work on differentially private recommendation systems [24]: in many non-private recommendation systems, recommendations made to individual i depend only on the item covariance information and the individual's own item ratings. In our terms, the recommendations to user i depend only on row i of the input matrix A and on A^TA. It makes no sense to hide the user's own ratings from himself, so it is sufficient that A^TA be approximated in a privacy-protective fashion.

larity of the privacy guarantee, protecting, variously, the privacy of each row *in its entirety* [4, 13, 21, 9], which is what we do here, or individual entries [24, 18], or norm 1 changes to any row [17]. Finally, the literature varies on the nature of differential privacy offered: so-called *pure*, or $(\epsilon, 0)$-differential privacy [13, 21, 9] and *approximate*, or (ϵ, δ), differential privacy [4, 24, 17, 18], which is the notion used in our work.

Refined Randomization: Blum *et al.* were the first to suggest privately releasing $A^T A$ by adding independent noise to each of the n^2 entries of this matrix [4]. The data analyst is then free to compute best rank k approximations to the privacy preserving, noisy, $\widehat{A^T A}$ for any and all k. This naïve noising approach, which has somewhat erroneously become known as *randomized response*, was refined in [13] to add less noise; our main algorithmic result is a careful analysis of a version of this refinement. Specifically, we will use the Gaussian mechanism [11], which adds independently chosen Gaussian noise to each entry of $A^T A$. When there is a gap in the singular values of A, or even a gap between singular values whose indices are not adjacent (formally $\sigma_k^2 - \sigma_{k'}^2 \in \omega(\sqrt{n}/(k + k'))$), we see a clear improvement, in captured variance, over previously published results. In this case, the analysis further shows, the space spanned by the top k right singular vectors of the (refined) noisy version of $A^T A$ is very close to the space spanned by the top k right singular vectors of A, with the spectral norm of the difference in projector matrices actually independent of k.

When there is no gap the algorithm performs no worse than the best in the literature; when $m \gg n$ we do expect such a gap: the more data, the better the algorithm's utility. The algorithm approaches the correct subspace of $A^T A$ at a rate faster than $1/m$, meaning that as we increase the number of samples the total error decreases.

Optimality: Our version of the refined noisy release of $A^T A$ is, up to logarithmic factors, optimal for approximate differential privacy. Pursuing a connection between differentially private algorithms and cryptographic traitor-tracing schemes [15], Bun, Ullman, and Vadhan [6] established lower bounds on errors for approximately differentially private release of a class of counting queries that are tight to within logarithmic factors. Their query class is based on a class of *fingerprinting codes* [5] due to Tardos [32]. We show that their result translates fairly easily to a lower bound for private approximation of the top singular vector. We also extend this to obtain lower bounds for rank k subspace estimation even for $k \in \Omega(n)$, a much more challenging task. Intuitively, for $k > 1$, we construct k "clusters" of fingerprinting codes. We have to overcome some difficulties to show that these clusters do not interfere much and to identify a "privacy-violating" vector hidden in a subspace. For the first we prove a stronger property of Tardos's codes, and for the second we introduce a game, called the list culling game, in which one player, using "planted questions", has to identify a good answer promised in a large set of answers provided by the other player. We propose a strategy for discovering the good answer with high success probability and apply it to constructing the privacy lower bound. Both results might be of independent interest.

Online Algorithms: Our third contribution merges two lines of research: differentially private regret minimization in online algorithms [14, 28] inspired by the Follow the Perturbed Leader (FPL) algorithm of Kalai and Vempala [20], and non-private online algorithms for principal components analysis [33]. A folk theorem says that differential privacy provides stability and hence reduces generalization error. We make this connection explicit in the online setting.

In the online model, computation proceeds in steps. At each time step t a rank k subspace V_t is output, a single data row A_t of A is received, and a reward is earned equal to $||A_t V_t||_2^2$. Regret is the difference between the sum of the earned rewards and the corresponding quantity for the best rank k matrix V chosen in hindsight (call it OPT). It is known, thanks to the pioneering work of [22], that the stability of an online algorithm is useful for achieving the low regret bound[2]. In [20], the FPL algorithm achieves stability by the addition of Laplace noise and is shown to have low regret. This technique has been successfully applied to several online algorithms. Indeed, for the online PCA problem, the previously best known FPL algorithm [33, 19] achieves a regret bound of $\widetilde{O}(\sqrt{kn\mathsf{OPT}})$. Our main observation is that a differentially private algorithm achieves similar stability to that of the FPL algorithm. With this insight, and borrowing tools from differential privacy, we show that, rather than adding Laplace noise, which might be unnecessarily large, one can instead add Gaussian noise, leading to an improved regret bound of only $\widetilde{O}(\sqrt{k\mathsf{OPT}}n^{1/4})$. In addition, by adding carefully correlated noise as in [14], we can make the entire algorithm private by incurring only a polylogarithmic factor in regret.

Granularity of Privacy: Two works of Hardt and Roth aim to exploit *low coherence* of the data matrix, a phenomenon of substantial interest in the (non-private) compressed sensing and matrix completion literature [7, 8, 26, 30, 25], to (privately) obtain good low rank approximations to the data matrix [17, 18]. There are several definitions of matrix coherence; roughly speaking coherence measures the extent to which the singular vectors are correlated with the standard basis. In the case of matrix completion, where the samples are intimately tied to the basis in which the data matrix is naturally represented, low coherence says that information is holographically embedded throughout the rows. The two definitions in [17] deal with row norms, either of the data matrix A or of U, when expressing $A = U\Sigma V^T$ in its singular value decomposition. There is an interplay between the granularity of the privacy guarantee and the specific coherence measure. The algorithms in [17], which are interesting when $n \geq m$, protect the rows in A up to any perturbation of Euclidean norm at most one. In this case the coherence conditions and the privacy granularity are rotationally invariant. In contrast, in [18] the coherence notion deals with the maximum entries of U and V, and the privacy granularity is for changes of magnitude at most one to a single entry of the data matrix. In this case neither the coherence condition nor the privacy granularity is rotationally invariant.

In our privacy definition, we protect the privacy against any individual row change. This is a natural choice for us as in many applications of PCA, each row corresponds to an individual. But for such a strong privacy notion (compared to single entry change or change of bounded norm), it is also more challenging to provide good utility. Indeed, we cannot achieve meaningful utility if we allow arbitrary A, for instance if one row has arbitrarily large norm. But in practice, allowing such "overpowering" individuals often goes against the purpose of PCA for discovering the global structure of many data records, and row normalization is often recommended before applying PCA. For example, in face recognition each individual image (a row in A) is typically normalized to have unit variance [2, 34]. Motivated by such practical considerations,

[2]Roughly speaking, in this context stability means that the output of the online algorithm does not change significantly between adjacent steps.

we assume each row to have at most unit ℓ_2 norm[3].

2. PRELIMINARIES

2.1 Notations and definitions

We treat vectors as column vectors (unless explicitly mentioned). For a given matrix $A \in \Re^{m\times n}$, we denote the i-th row of A by A_i, which in this case is a row vector. For a vector $x \in \Re^n$, $\|x\|$ denotes the ℓ_2 norm. For a matrix $A \in \Re^{m\times n}$, the spectral norm is defined as $\|A\|_2 = \max_{x\in\Re^n, \|x\|_2=1} \|Ax\|_2$; the Frobenius norm is defined as $\|A\|_F = \sqrt{\sum_{i\in[m],j\in[n]} a_{ij}^2}$, where a_{ij} are the entries of the matrix A. For a square matrix, the trace $\mathsf{tr}(\cdot)$ is defined as the sum of its diagonal elements. So $\|A\|_F^2 = \mathsf{tr}(A^TA) = \mathsf{tr}(AA^T)$. Slightly abusing terminology we will refer to A^TA as the covariance matrix of A.

For a matrix A, the singular value decomposition of A is defined as $A = U\Sigma V^T$, where $U \in \Re^{m\times m}$ and $V \in \Re^{n\times n}$ are unitary matrices and called the *left* and *right* singular subspaces, respectively. The matrix $\Sigma \in \Re^{m\times n}$ is a diagonal matrix with non-negative entries $\sigma_1, \ldots, \sigma_{\min(m,n)}$ along the diagonal, called the singular values. In this paper, we assume they are ordered decreasingly, i.e $\sigma_1 \geq \sigma_2 \geq \ldots$. Suppose that $V = (v_1, \ldots, v_n)$. We define $V_k = (v_1, \ldots, v_k)$ and call it the principal (or top) k right singular subspace. It is well known that $\|A\|_2 = \sigma_1$, $\|A\|_F^2 = \sum_i \sigma_i^2$, and $\|AV_k\|_F^2 = \sum_{i=1}^k \sigma_i^2 = \max_{P\in\mathbb{P}_k} \|AP\|_F^2$.

Each row $a_i \in \Re^n$, $1 \leq i \leq m$, of the data matrix $A \in \Re^{m\times n}$ represents the attributes of a single user. As discussed above, we assume each row has at most unit ℓ_2 norm. The set of all such matrices is denoted $\mathcal{A}$.

Given the data matrix A, our objective is to output a subspace that preserves privacy and captures the variance of A as much as possible. To define privacy, we call two matrices $A, A' \in \mathcal{A}$ *neighbors* if they differ in exactly one row, as each row in A corresponds to an individual user. We will ensure (ϵ, δ)-differential privacy.

DEFINITION 1 (DIFFERENTIAL PRIVACY [13, 11]). *A randomized mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if for every two neighboring matrices $A, A' \in \mathcal{A}$ and for all events $\mathcal{O} \subseteq Range(\mathcal{M})$, $\Pr[\mathcal{M}(A) \in \mathcal{O}] \leq e^\epsilon \Pr[\mathcal{M}(A') \in \mathcal{O}] + \delta$.*

Let $f : \mathcal{A} \to \Re^p$ be a vector-valued function operating on databases. The ℓ_2-*sensitivity* of f, denoted Δf, is the maximum over all pairs A, A' of neighboring datasets of $||f(A) - f(A')||_2$. The *Gaussian mechanism* adds independent noise drawn from a Gaussian with mean zero and standard deviation slightly greater than $(\Delta f)\ln(1/\delta)/\epsilon$ to each element of its output.

THEOREM 2 (GAUSSIAN MECHANISM [11, 16]). *Let $f : \mathcal{A} \to \Re^p$ be a vector-valued function. Let $\tau = \Delta f\sqrt{2\ln(1.25/\delta)}/\epsilon$. The Gaussian mechanism, which adds independently drawn random noise distributed as $\mathcal{N}(0, \tau^2)$ to each output of $f(A)$, ensures (ϵ, δ)-differential privacy.*

We are interested in the function $f(A) = A^TA$, which may be viewed as an n^2-dimensional vector. Because we ensure that $||a_i||_2 \leq 1$, the sensitivity of f is at most one.

2.2 Summary of main results

For the purposes of brevity, throughout the paper, we use $\widetilde{O}(\cdot), \widetilde{\Omega}(\cdot)$ to hide factors of $1/\epsilon$ and polynomial dependence on $\log(1/\delta), \log m$, and $\log n$, and "with high probability" means with probability $1 - 1/n^{\Omega(1)}$, under the internal randomness of the mechanism. Our first result (Main Result 1) is that the Gaussian mechanism is nearly optimal in the worst case[4]. We further show (Main Result 2) that, under natural assumptions on the data matrix A, this mechanism has even stronger utility guarantees.

MAIN RESULT 1 (THEOREMS 3 AND 18 INFORMAL VERSION).

1. *For any $\epsilon, \delta > 0$ and $1 \leq k \leq n$, the Gaussian mechanism described in Thoerem 2 ensures that for any $A \in \mathcal{A}$, with high probability over the coin tosses of the mechanism, $\|A\mathcal{M}(A)\|_F^2 \geq \|AV_k(A)\|_F^2 - \widetilde{O}(k\sqrt{n})$.*
2. *The Gaussian mechanism is nearly optimal: for any $1 \leq k \leq n$ and any $(\epsilon, 1/n^2)$-differentially private mechanism $\mathcal{M}$, there exists $A \in \mathcal{A}$ such that $\|A\mathcal{M}(A)\|_F^2 \leq \|AV_k(A)\|_F^2 - \widetilde{\Omega}(k\sqrt{n})$.*

MAIN RESULT 2 (THEOREMS 4 AND 6 INFORMAL VERSION). *Let $\sigma_1 \geq \cdots \geq \sigma_n$ be the singular values of $A \in \mathcal{A}$. Assuming $\sigma_k^2 - \sigma_{k'+1}^2 = \omega(\sqrt{n})$, with high probability the Gaussian mechanism $\mathcal{M}$ satifies*

$$\|A\mathcal{M}(A)\|_F^2 \geq \|AV_k\|_F^2 - \widetilde{O}\left(\frac{k'n}{\sigma_k^2 - \sigma_{k'+1}^2}\right).$$

Additionally, when $k' = k$,

$$\|\mathcal{M}(A)\mathcal{M}(A)^T - V_kV_k^T\|_2 = \widetilde{O}\left(\frac{\sqrt{n}}{\sigma_k^2 - \sigma_{k+1}^2}\right).$$

Finally, we consider the online version in which a_t arrives in a stream for $t = 1, \cdots, m$, and the mechanism $\mathcal{M}$ is required to compute a k-dimensional subspace $\mathcal{M}_t = \mathcal{M}(a_1, \cdots, a_{t-1})$ before seeing a_t. Define $\mathsf{OPT} = \max_{P\in\mathbb{P}_k} \sum_{t=1}^m \|P^Ta_t\|_2^2$. The regret of $\mathcal{M}$ is defined as $\mathsf{Regret}(\mathcal{M}) = \mathsf{OPT} - \sum_{t=1}^m \|\mathcal{M}_t^Ta_t\|_2^2$. We show (in main Theorem 3) that by adding carefully calibrated noise, the Following the Perturbed Leader algorithm in [20] can be made both private and with low regret. And the regret bound is nearly optimal for any online private PCA algorithm.

MAIN RESULT 3 (THEOREM 8 INFORMAL VERSION). *When $\mathsf{OPT} = \widetilde{\Omega}(k\sqrt{n}/\epsilon^2)$, we can obtain an (ϵ, δ)-differentially private online mechanism $\mathcal{M}$ such that $\mathbb{E}[\mathsf{Regret}(\mathcal{M})] = \widetilde{O}(\sqrt{k\mathsf{OPT}}n^{1/4})$. This bound is nearly optimal for $\mathsf{OPT} = \widetilde{O}(k\sqrt{n})$.*

Due to the space limit, we will omit from this abstract many proof details. They can be found in the full version.

3. PRIVATE SINGULAR SUBSPACE COMPUTATION VIA THE GAUSSIAN MECHANISM

The Gaussian mechanism (with symmetric noise matrix) is straightforward: just release $\widehat{C} = A^TA + E$ where E is a symmetric noise matrix, with each (upper-triangle) entry drawn i.i.d. from Gaussian distribution with sufficiently high variance. Algorithm 1 describes such a mechanism, a variation of those in [4, 13] that enjoys smaller noise, and in which the noise matrix is symmetric. Set $\Delta_{\epsilon,\delta} = \sqrt{2\ln(1.25/\delta)}/\epsilon$.

[3]To enforce this condition, an offending row can be divided by its own norm; this will not affect privacy.

[4]We will tweak the mechanism slightly by ensuring that the matrix of noise values added to A^TA is symmetric. We abuse notation by referring to this symmetric version simply as the Gaussian mechanism.

Algorithm 1 The Gaussian Mechanism: releasing the covariance matrix privately

Input: matrix $A \in \Re^{m \times n}$, and privacy parameters $\epsilon, \delta > 0$.

1: $E \in \Re^{n \times n}$ be a symmetric matrix where the upper triangle (including the diagonal) is i.i.d. samples from $\mathcal{N}\left(0, {\Delta_{\epsilon,\delta}}^2\right)$, and each lower triangle entry is copied from its upper triangle counterpart.

2: Output $\widehat{C} \leftarrow A^T A + E$.

That Algorithm 1 provides (ϵ, δ)-differential privacy is immediate from Theorem 2, using the fact that the ℓ_2 sensitivity of $f(A) = A^T A$, when viewed as an n^2-dimensional vector, is 1.

Differential privacy is closed under post-processing, so the data analyst can run any post-processing algorithm on $\widehat{C}$ with no further erosion of privacy. In particular, the analyst can compute the singular decomposition of $\widehat{C}$ to obtain any k-dimensional principal singular subspace $\widehat{V}_k$ of $\widehat{C}$. But how useful is such a $\widehat{V}_k$? In this section, we will show that $\widehat{V}_k$ can actually be a quite good approximation to the principal rank-k right singular subspace V_k of A (or equivalently the principal singular subspace of $A^T A$.) In particular, we consider three measures: *1) How well does $\widehat{V}_k$ capture the variance of A compared to V_k? 2) How close is $\widehat{V}_k$ to V_k? and 3) How well does the best rank-k approximation of $\widehat{C}$ approximate $A^T A$?*

Our analyses come in two flavors. One is on the worst case guarantee, where no assumption is made on A. Most of these results follow relatively easily from random matrix theory. As we will show later by our lower bound, one cannot expect to outperform these bounds in the worst case. The other set of results depend on the spectrum of $A^T A$. We show, by using tools from matrix perturbation theory, that when the spectrum of $A^T A$ has large drop in its eigenvalues, $\widehat{V}_{k'}$ can be a much better approximation to V_k when $k' \geq k$. For example, when the data are drawn from a distribution with an eigengap, the error will go to 0 as the number of samples $m \to \infty$! Since the presence of such drop is one of the rationales for principal components analysis, these results are probably more interesting in practice. We emphasize that this improved data dependent bound holds for the same algorithm; the gain comes entirely from the analysis.

3.1 Variance guarantee

We now consider how well $\widehat{V}_k$ captures the variance of A. We first provide a worst case bound.

THEOREM 3 (WORST CASE UTILITY GUARANTEE). *Let V_k be the principal rank-k right singular subspace of A and let $\widehat{V}_k$ be the principal rank-k subspace of the matrix $\widehat{C}$ (output by Algorithm 1). Then with high probability,*

$$\|A\widehat{V}_k\|_F^2 \geq \|AV_k\|_F^2 - O\left(k\sqrt{n}\Delta_{\epsilon,\delta}\right) .$$

As we will see in Section 5, the above bound is nearly tight in the worst case. Now, suppose there is a large eigengap, so that $\sigma_k - \sigma_{k+1} \in \omega(\sqrt{n})$. In this case we will see that $\widehat{V}_k$ can provide utility that beats the worst-case lower bound. Moreover, an analagous claim holds even if there is not a precipitous drop between adjacent eigenvalues.

THEOREM 4 (SPECTRUM SEPARATION GUARANTEE). *Let $\sigma_1 \geq \cdots \geq \sigma_n$ be the singular values of the data matrix A. Let V_k be the principal rank-k right singular subspace of A. Let $\widehat{V}_{k'}$ be the principal $k' \geq k$-dimensional subspace of the matrix $\widehat{C}$ (output by Algorithm 1). Assuming $\sigma_k^2 - \sigma_{k'+1}^2 = \omega(\sqrt{n}\Delta_{\epsilon,\delta})$, with high probability,*

$$\|A\widehat{V}_{k'}\|_F^2 \geq \|AV_k\|_F^2 - O\left(\frac{k' n {\Delta_{\epsilon,\delta}}^2}{\sigma_k^2 - \sigma_{k'+1}^2}\right) .$$

PROOF. The basic tool in our analysis is a sin-θ theorem, which is a generalization of the classic *Davis-Kahan sin-θ* theorem [10]. By the optimality of $\widehat{V}_{k'}$ and using $k' \geq k$, we have,

$$\begin{aligned}\mathsf{tr}(\widehat{V}_{k'}^T (A^T A)\widehat{V}_{k'}) &\geq \mathsf{tr}(V_k^T (A^T A) V_k) + \mathsf{tr}(V_k^T E V_k) - \mathsf{tr}(\widehat{V}_{k'}^T E \widehat{V}_{k'}) \\ &= \mathsf{tr}(V_k^T (A^T A) V_k) + \mathsf{tr}\left(\left(V_k V_k^T - \widehat{V}_{k'}\widehat{V}_{k'}^T\right) E\right) .\end{aligned}$$

For the ease of notation, let $\Pi = V_k V_k^T$ and $\widehat{\Pi} = \widehat{V}_{k'}\widehat{V}_{k'}^T$. To bound $\mathsf{tr}\left(\left(\Pi - \widehat{\Pi}\right) E\right)$, we use Von Neumann's trace inequality: For two matrices $X \in \Re^{n\times n}$ and $Y \in \Re^{n \times n}$, let $\sigma_i(X), \sigma_i(Y)$ be the decreasingly ordered singular values of X, Y, respectively. Then $|\mathsf{tr}(XY)| \leq \sum_{i=1}^n \sigma_i(X)\sigma_i(Y)$. Hence, we have

$$|\mathsf{tr}\left(\left(\Pi - \widehat{\Pi}\right) E\right)| \leq \sum_{i=1}^n \sigma_i\left(\Pi - \widehat{\Pi}\right) \cdot \sigma_i(E) .$$

Since $\left(\Pi - \widehat{\Pi}\right)$ is of rank at most $k + k' \leq 2k'$, at most $2k'$ of the σ_i are non-zero. So we further have

$$\begin{aligned}|\mathsf{tr}\left(\left(\Pi - \widehat{\Pi}\right) E\right)| &\leq \|E\|_2 \sum_{i=1}^{2k'} \sigma_i\left(\Pi - \widehat{\Pi}\right) \\ &\leq \sqrt{2k'}\|E\|_2 \left\|\Pi - \widehat{\Pi}\right\|_F \qquad (1)\end{aligned}$$

We now have the following.

$$\Pi - \widehat{\Pi} = \Pi(\mathbb{I} - \widehat{\Pi}) - (\mathbb{I} - \Pi)\widehat{\Pi} = \Pi\widehat{\Pi}^\perp - \Pi^\perp\widehat{\Pi} . \qquad (2)$$

Plugging (2) in (1), we have the following.

$$\begin{aligned}|\mathsf{tr}\left(\left(\Pi - \widehat{\Pi}\right) E\right)| &\leq \sqrt{2k'}\|E\|_2 \left\|\Pi\widehat{\Pi}^\perp - \Pi^\perp\widehat{\Pi}\right\|_F \\ &\leq \sqrt{2k'}\|E\|_2 \left(\left\|\Pi\widehat{\Pi}^\perp\right\|_F + \left\|\Pi^\perp\widehat{\Pi}\right\|_F\right) \\ &= \sqrt{2k'}\|E\|_2 \left(\left\|\Pi\widehat{\Pi}^\perp\right\|_F + \left\|\widehat{\Pi}\Pi^\perp\right\|_F\right) \qquad (3) \\ &\leq \sqrt{2}k'\|E\|_2 \left(\|\Pi\widehat{\Pi}^\perp\|_2 + \|\widehat{\Pi}\Pi^\perp\|_2\right) , \qquad (4)\end{aligned}$$

where (3) follows because $\Pi^\perp, \widehat{\Pi}$ are symmetric matrices (since they are projectors), and for symmetric E, F, $\|EF\|_F = \|FE\|_F$.

Let $X, Y \in \Re^{n\times n}$ be two symmetric matrices, and let $\lambda_1(X) \geq \cdots$ and $\lambda_1(Y) \geq \cdots$ be the corresponding eigenvalues of X and Y. Let $\Pi_X^{(i)}$ be the projector to the subspace spanned by the top i singular vectors of X, where $i \leq n$. To bound $\|\Pi\widehat{\Pi}^\perp\|_2$ and $\|\widehat{\Pi}\Pi^\perp\|_2$, we will use the following result from matrix perturbation theory, which generalizes [10]:

THEOREM 5 (SIN-Θ THEOREM [23] (COROLLARY 8)). *For any $1 \leq i, j \leq n$,*

$$(\lambda_i(X) - \lambda_{j+1}(Y))\|\Pi_X^{(i)}(\mathbb{I} - \Pi_Y^{(j)})\|_2 \leq \|X - Y\|_2 .$$

Now to bound $\|\Pi\widehat{\Pi}^\perp\|_2$ in (4), we use Theorem 5 with $X = A^T A$ and $Y = A^T A + E$. Notice that $\|Y - X\|_2 = \|E\|_2$, and since E is a symmetric Gaussian ensemble, by Corollary 2.3.6 from [31], with high probability, $\|E\|_2 = O\left(\sqrt{n}\Delta_{\epsilon,\delta}\right)$. Also by Weyl's inequality it follows that $\lambda_{j+1}(Y) \leq \lambda_{j+1}(X) + \|E\|_2$. Plugging these

bounds in Theorem 5 and recalling that $\sigma_k^2 - \sigma_{k'+1}^2 = \omega(\sqrt{n}\Delta_{\epsilon,\delta})$ (by assumption), we get $\|\Pi\widehat{\Pi}^{\perp}\|_2 = O\left(\frac{\sqrt{n}\Delta_{\epsilon,\delta}}{\sigma_k^2 - \sigma_{k'+1}^2}\right)$.

Using the same argument as above, and selecting $X = A^TA + E$ and $Y = A^TA$ in Theorem 5, we get $\|\widehat{\Pi}\Pi^{\perp}\|_2 = O\left(\frac{\sqrt{n}\Delta_{\epsilon,\delta}}{\sigma_k^2 - \sigma_{k'+1}^2}\right)$. Theorem 4 follows now follows from these bounds. □

While the bound in Theorem 3 may not be useful when $\sigma_k^2 - \sigma_{k'+1}^2$ is small, in many cases (even for $k' = k$) the gap is quite large, especially when the number of samples m is large. Here we give two examples. In the first example, suppose that a_i's are drawn i.i.d. from some distribution with a specturm gap, say α, between σ_k^2 and σ_{k+1}^2. Then by the matrix concentration bound, it is easy to see that when $m \gg \sqrt{n \log n}/\alpha$, the gap is $\Omega(\alpha m)$ with high probability. In this case, Theorem 4 provides a better bound than Theorem 3. In the second example the a_i's are random Gaussian vectors, where there is no eigengap (in this case the usefulness of PCA is problematic but we use it as an illustration). For m random samples, the gap between two consecutive eigenvalues is expected to be $\Omega(\sqrt{m}/n^2)$, so in this case, Theorem 4 provides a better upper bound whenever $m = \Omega(n^5)$. In both cases, the error gap of Algorithm 1 goes to 0 when $m \to \infty$!

Bounds on residual variance. We observe that by Pythagorean theorem, $\|A - A(V_kV_k^T)\|_F^2 = \|A\|_F^2 - \|AV_k\|_F^2$. Since the bounds in Theorem 3 and 4 are additive, the same error guarantees hold if we are to minimize the total variance projected in the residual space.

3.2 Closeness to the right singular subspace

Another consequence of Theorem 4 is that when there is a spectrum gap in A^TA, $\widehat{V}_k$ not only captures large amount of variance, but is also close to the top k right singular subspace V_k of A. In Theorem 6, we provide the closeness between them, measured by the $\|\cdot\|_2$ norm. We note that the spectrum gap is necessary for such a bound as otherwise the top k-singular space is not uniquely defined.

THEOREM 6 (SUBSPACE CLOSENESS). *Let $\sigma_1 \geq \cdots \geq \sigma_n$ be the singular values of the data matrix A. Assuming $\sigma_k^2 - \sigma_{k+1}^2 = \omega(\sqrt{n}\Delta_{\epsilon,\delta})$, then with high probability,*

$$\left\|V_kV_k^T - \widehat{V}_k\widehat{V}_k^T\right\|_2 = O\left(\frac{\sqrt{n}\Delta_{\epsilon,\delta}}{\sigma_k^2 - \sigma_{k+1}^2}\right).$$

We note that the above bound implies an upper bound in terms of the Frobenius norm

$$\left\|V_kV_k^T - \widehat{V}_k\widehat{V}_k^T\right\|_F = O\left(\frac{\sqrt{kn}\Delta_{\epsilon,\delta}}{\sigma_k^2 - \sigma_{k+1}^2}\right).$$

3.3 Low-rank approximation to the covariance matrix

$\widehat{C}$ also provides a good low rank approximation to A^TA in the settings considered in seveal earlier works [17, 21, 18] (see Section 1 for comparison).

THEOREM 7 (LOW RANK APPROXIMATION). *Let $A \in \Re^{m\times n}$ be the input data matrix and let C_k be the best rank-k approximation to A^TA. Let $\widehat{C}_k$ be the rank-k approximation to $\widehat{C}$ (output by Algorithm 1). Then with high probability,*

- $\|A^TA - \widehat{C}_k\|_F \leq \|A^TA - C_k\|_F + O\left(\Delta_{\epsilon,\delta}k\sqrt{n}\right)$.
- $\|A^TA - \widehat{C}_k\|_2 \leq \|A^TA - C_k\|_2 + O\left(\Delta_{\epsilon,\delta}\sqrt{n}\right)$.

In the above, the spectral norm bound can be derived immediately from [1] (Lemma 1.1). For the Frobenius norm bound, compared to the bound there, we need to prove a strengthened version with a better dependence on the spectrum of $C = A^TA$.

Data-dependent Noise Addition algorithm. While releasing a perturbed covariance matrix allows great flexbility for the data analyst, it requires releasing an $n \times n$ matrix, which can be computationally expensive.This motivates another noise addition algorithm: apply SVD first to A and then release the top singular subspace by adding noise calibrated to its gap-dependent sensitivity. This yields improved running time when the matrix A is sparse. But we need to be careful not to violate the data privacy in the process. This can done by employing a variant of the *propose-test-and-release* (PTR) framework of [12]. We present such an algorithm in the full version of this paper.

4. PRIVATE ONLINE SINGULAR SUBSPACE COMPUTATION

The design of the private online algorithm turns out to be closedly related to the class of *follow the perturbed leader* (FPL) algorithms of [20]. Such algorithms add regularization noise to the problem to reduce generalization error and hence the regret. This noise also reduces the dependence of the algorithm outcome on individual data items and therefore is aligned with the goal of providing privacy. Recall that $\mathbb{P}_k$ denotes the set of k-dimensional orthogonal projectors in $\Re^n$. Define $\mathsf{OPT} = \max_{P\in\mathbb{P}_k} \sum_{t=1}^m \|Pa_t\|^2$. We show that by using the Gaussian noise for the regularization noise, we can achieve the regret bound (defined in Section 2.2) of $\widetilde{O}(\sqrt{k\mathsf{OPT}}n^{1/4})$.

THEOREM 8 (REGRET GUARANTEE). *If $\delta < 1/m^2$, $\epsilon < 0.1$, $m = O(\mathsf{poly}\, n)$ and $\mathsf{OPT} > \frac{k\sqrt{n}\log^2(m/\delta)}{\epsilon^2}$, then there exists an (ϵ,δ)-differentially private online learning algorithm whose regret guarantee is the following.*

$$\mathbb{E}[\mathsf{Regret}] = O\left(\sqrt{k\mathsf{OPT}}n^{1/4}\log^2(m/\delta)\right).$$

As can be shown from the lower bound for the offline problem, the $\widetilde{O}(\sqrt{k\mathsf{OPT}}n^{1/4})$ bound is nearly optimal for $\mathsf{OPT} = O(k\sqrt{n})$. Therefore the $n^{1/4}$ gap between the regret of the non-private and private algorithms is essentially tight.

The above theorem is proved in two steps. We first present an FPL algorithm that gives regret of $\widetilde{O}(\sqrt{k\mathsf{OPT}}n^{1/4})$. The analysis of the algorithm borrows techniques from differential privacy but the algorithm itself is not private. We then show that, by using the *tree based aggregation technique* [3, 14], we can obtain a private online mechanism with similar regret bound, at a cost of only a $\log^{O(1)}(m/\delta))$ factor.

Algorithm 2 is the formal description of our basic FPL algorithm. The algorithm is simple: at each step $t \in [m]$ it executes Algorithm 1 and output the result. One can provide the following regret guarantee (Theorem 9) for Algorithm 2.

THEOREM 9 (REGRET GUARANTEE). *For $\epsilon < 1, \delta < 1/m$, the regret guarantee for Algorithm 2 is the following.*

$$\mathbb{E}[\mathsf{Regret}] = O\left(\frac{k\sqrt{n}\log m}{\epsilon} + \epsilon\mathsf{OPT} + 1\right).$$

While each step in Algorithm 2 is (ϵ,δ)-differentially private, overall it is not. This is easily remedied with a classical *tree-based* technique [3] applied in [14] for ensuring differential privacy under continual observation.

Algorithm 2 Online singular subspace computation.

Input: Vectors $a_1, \ldots, a_m \in \Re^n$ where $\|a_t\| \leq 1$, rank parameter: k, regularization parameter: ϵ, δ.

Output: k-dimensional subspaces $\widehat{V}_1, \ldots, \widehat{V}_m$.

1: Choose an arbitrary rank k subspace $\widehat{V}_1$.
2: **for** $t \leftarrow 1$ to m **do**
3: Get a *reward* $R_t = \|\widehat{V}_t^T a_t\|_2^2 = \mathtt{tr}(a_t^T \widehat{V}_t \widehat{V}_t^T a_t)$ and receive input a_t.
4: Compute $C_t = \sum_{\tau=1}^{t} a_\tau a_\tau^T$
5: Compute $\widehat{C}_t = C_t + E_t$, where E_t is sampled as in Algorithm 1 using the parameters ϵ, δ.
6: Compute $\widehat{V}_{t+1}$ as the top k singular subspace of $\widehat{C}_t$.
7: **end for**

The technique is simple. Assume m is a power of 2. We will divide the input stream into the natural set of intervals corresponding to the labels on a complete binary tree with m leaves, where the leaves are labeled from left to right, with the intervals $[0,0], \ldots, [m-1, m-1]$ and each parent is labeled with the interval that is the union of the intervals labeling its children. (Note that we have indexed the inputs starting with zero.) The idea is to run and release the results of Algorithm 1 for each label $[s,t]$; that is, the released result corresponding to the label $[s;t]$ is the output of Algorithm 1 on the data items $a_s, \ldots, a_t$, that is, an approximation to $\sum_{i=s}^{t} a_i^T a_i$ (recall that the a_i are row vectors). Let M_t denote the matrix whose rows are (row vectors) $a_1, \ldots, a_t$ for $t \in [0, m-1]$. To obtain an approximation to $M_t^T M_t = \sum_{i=1}^{m} a_i^T a_i$, the analyst uses the binary representation of t to determine a set of at most $\log_2 m$ disjoint intervals whose union is $[0,t]$. These intervals correspond to internal nodes in the tree. The outputs corresponding to these nodes are summed to obtain the desired approximation. The key point for privacy is that each element in the stream affects only $1 + \log_2 m$ invocations of Algorithm 1.

At each step t we obtain $\widehat{C}_t = C_t + E'_t$, where E'_t is a symmetric matrix whose upper triangular entries are i.i.d. with variance $\log^3(m/\delta)\Delta_{\epsilon,\delta}{}^2$. (Note that $\widehat{C}_t$ denotes the output at step t, and not a rank t subspace of anything!) Plugging this into Algorithm 2 and using the standard doubling trick from the online learning literature (see Section 2.3.1. in [27]), we obtain the bound as claimed in Theorem 8.

5. LOWER BOUNDS

Bun, Ullman and Vadhan [6] recently showed that the existence of fingerprinting codes can be used to prove lower bounds on the error of (ϵ, δ)-differentially private mechanisms. We next show that using some of their tools, with some extra effort, one can derive a lower bound for private subspace estimation that nearly matches our upper bounds.

Fingerprinting codes were introduced by Boneh and Shaw [5] for watermarking. Informally, a fingerprinting code is a (distribution over) collection of codewords, one to each agent which has the property that no coalition of agents with access only to its own codewords will be able to produce a valid-looking codeword without at least one coalition member being identified. Formally, we have a pair of (randomized) algorithms Gen and $Trace$. Gen outputs a codebook C consisting of t vector $c_1, \ldots c_t \in \{-1,1\}^n$ with c_i representing the codeword given to user i. Given a subset $S \subseteq [t]$ of agents, let $c_S \in \{-1,0,1\}^n$ be defined as

$$c_{Sj} = \begin{cases} +1 & \text{if } c_{ij} = +1 \forall i \in S \\ -1 & \text{if } c_{ij} = -1 \forall i \in S \\ 0 & \text{otherwise} \end{cases}$$

Let $F_+(S) = \{j \in [n] : c_{Sj} = 1\}$ and similarly $F_-(S) = \{j \in [n] : c_{Sj} = -1\}$. Let $F(S) = F_+(S) \cup F_-(S)$ denote the set of unanimous coordinates in S where all codewords in S agree. We say that a vector $c' \in \{-1,1\}^n$ is β-valid for S if c'_j agrees with c_{Sj} in at least a $(1-\beta)|F(S)|$ of the locations in $F(S)$. In other words, $\Pr_{j \sim F(S)}[c'_j = c_{Sj}] \geq 1 - \beta$. (Robust) Fingerprinting codes have the property that given a c' that is β-valid for a coalition S, the tracing algorithm $Trace$ outputs a member of the coalition with high probability. We use the following definition (essentially) from [6].

DEFINITION 10 (WEAKLY ROBUST FINGERPRINTING CODES). *Let t, n, f be integers and let $\xi, \beta \in [0,1]$. A pair of algorithms $(Gen, Trace)$ is a (t, n, f, β, ξ)-fingerprinting code if Gen outputs a codebook $C = \{c_1, \ldots, c_t\} \subseteq \{-1,1\}^n$ and for every possible (possibly randomized) adversary A_{pirate}, and for every coalition $S \subseteq [t]$,*

1. $\Pr[Trace(C, c') \in S \mid c' \text{ is } \beta\text{-valid for } S] \geq 1 - \xi$.
2. $\Pr[Trace(C, c') \in [t] \setminus S] \leq \xi$.
3. $\Pr[|F(S)| \geq f] \geq 1 - \xi$.

where $c' = A_{pirate}(c_i : i \in S)$ and the probability is taken over the coins of Gen,$Trace$ and A_{pirate}.

Bun et al. [6] show that the fingerprinting codes construction of Tardos [32] is weakly robust.

THEOREM 11. *For every $n \in \mathbb{N}$ and $\xi \in [0,1]$, the construction of [32] gives an $(t, n, f, \frac{1}{20}, \xi)$ fingerprinting code such that*

$$t = \Omega(\sqrt{n/\log(n/\xi)}) \qquad f = \Omega(t^{\frac{3}{2}})$$

Finally, the following theorem, essentially from [6] shows how fingerprinting codes lead to lower bounds for differentially private mechanisms (by setting $\xi, \delta = O(1/n^2)$).

THEOREM 12. *Let $\mathcal{M} : D^m \to D$ be an (ϵ, δ)-DP mechanism with $D = \{-1,1\}^n$. If $(m+1, n, f, \beta, \xi)$-weakly robust fingerprinting codes exist with security $\xi \leq \frac{1}{2}$, then*

$$\Pr[\mathcal{M}(C|_S) \text{ is } \beta\text{-valid for } S] \leq m(2\xi \exp(\epsilon) + \delta).$$

PROOF. Let $\mathcal{M}'(C|_S) = Trace(\mathcal{M}(C|_S))$, and let p denote $\Pr[\mathcal{M}(C|_S)$ is β-valid for S]. Then by the first property of fingerprinting codes, $\Pr[\mathcal{M}'(C|_{[m]}) \in [m]] \geq p(1-\xi)$. Thus there exists an $i \in [m]$ such that $\Pr[\mathcal{M}'(C|_{[m]}) = i] \geq \frac{p(1-\xi)}{m}$.

Let $S' = [m+1] \setminus \{i\}$. Then by the second property of fingerprinting codes, $\Pr[\mathcal{M}'(C|_{S'}) = i] \leq \xi]$. Since $\mathcal{M}'$ satisfies (ϵ, δ)-DP, it follows that

$$\frac{p(1-\xi)}{m} \leq \exp(\epsilon)\xi + \delta.$$

Rearranging gives the result. □

Because Differential Privacy is closed under post-processing, this says that a differentially private mechanism $\mathcal{M}$ cannot even produce a vector in $\Re^n$ whose sign agree with $C|_S$ in a $(1-\beta)$ fraction of the locations in $F(S)$ (or else we could round this vector and contradict the theorem).

5.1 Lower bound for eigenvector computation

We say a unit vector v is an α-useful eigenvector for a matrix A if $\|Av\|_2^2 \geq \|Av'\|_2^2 - \alpha$ for every unit vector v'. The main result of this section says that no differentially private mechanism can output a v that is $o(m)$-useful on any $m \times n$ matrix, if (m, n, f, β, ξ)-fingerprinting codes exist for appropriate f, β, ξ. At a high level, we construct a hard matrix by taking a fingerprinting codes matrix, padding it with many 1s, and suitably scaling to make rows norm 1. For the top eigenvector v_1 of this matrix, either v_1 or $-v_1$ must agree with $C|_S$ in sign on all the consensus locations, and we can use the padding bits to pick between v_1 and $-v_1$. Lemma 14 is a robust version of this statement. The padding also ensures a large gap between the first and the second eigenvalue (Lemma 15), so that any $o(m)$-good vector must be very close to v_1. Thus we can use any $o(m)$-good vector to construct a β-valid vector for appropriate β. We next give the details.

THEOREM 13. *There is a universal constant K such that the following holds. Suppose there is an (ϵ, δ)-DP mechanism that for any matrix $A \in \Re^{m \times 16n}$ with each row having norm at most 1 outputs an γm-useful eigenvector of A with probability p. Then there is an (ϵ, δ)-DP mechanism that on input $S = \{c_1, \ldots, c_m\}$ from a $(m+1, n, f, \beta_0, \xi)$-fingerprinting code outputs a c' that is $K\gamma$-valid for S with probability $p - \xi - \exp(-\Omega(\gamma^2 f))$.*

The proof of this result uses a padding approach similar to the strongly robust fingerprinting codes construction in [6].

Algorithm 3 Pirate algorithm A_{pirate}

Input: Set of codewords $S = \{c_1, \ldots, c_m\}$ with $c_i \in \{-1, 1\}^n$. Oracle access to Mechanism $\mathcal{M}$ for privately computing top right singular vector.

1: Let $pad \leftarrow 1^{15n}$.
2: **for** $i = 1, \ldots, m$ **do**
3: Let $c_i^{(1)} \leftarrow c_i \circ pad$.
4: Let $c_i^{(2)} \leftarrow c_i^{(1)}/\sqrt{16n}$.
5: **end for**
6: Let P be a random permutation matrix. Replace each 1 in P by a -1 with probability $\frac{1}{2}$.
7: Let A be the $m \times n$ matrix with the transposes of $c_i^{(2)}$'s as its rows.
8: Let $A' \leftarrow AP$.
9: Let $v \leftarrow \mathcal{M}(A')$ be the γm-useful right singular vector output by $\mathcal{M}$.
10: Let $w \leftarrow Pv$.
11: **if** $\sum_{j=n+1}^{15n} w_j \leq 0$ **then**
12: $w \leftarrow -w$.
13: **end if**
14: **for** $j = 1 \ldots n$ **do**
15: $c'_j = sgn(w_j)$.
16: **end for**
17: **return** c'

PROOF. Let $\mathcal{M}$ be a differentially private mechanism that outputs a γm-useful eigenvector for any input matrix A. We will use it as a subroutine to construct a differentially private mechanism $\mathcal{M}'$ that outputs a β-valid codeword for an appropriate β.

The mechanism $\mathcal{M}'$ works as follows. Let $c_1, \ldots, c_m$ be the input vectors to $\mathcal{M}'$. We first set pad to the vector 1^{15n} append it to each of the c_i's to get $c_i^{(1)} \in \Re^{16n}$. We then scale each $c_i^{(1)}$ to get a unit vector, by setting $c_i^{(2)} = c_i^{(1)}/\sqrt{16n}$. Let A be the matrix with rows (transpose of) $c_i^{(2)}$. Finally, we pick a random permutation matrix P and replace each 1 by -1 with probability $\frac{1}{2}$. We set $A' = AP$. Thus A' is obtained by randomly permuting the columns of A and randomizing the sign of each column. We run the mechanism $\mathcal{M}$ on A', to get a γm-useful vector v.

We then postprocess v as follows: we undo the signed permutation P and without loss of generality, assume that sum of entries of Pv on the pad locations is non-negative (if not, replace v by $-v$). We then strip off the padding and set $c'_j = sgn((Pv)_j)$ for each $j \in [n]$. This defines the output of $\mathcal{M}'$. The privacy of $\mathcal{M}'$ follows immediately from the post-processing property of differential privacy and the fact that pad and P did not depend on the data c_i's. We next argue that, conditioned on v being γ-useful, c' is β-valid.

We first establish two useful properties of the eigen-decomposition of A'. The permutation P does not change the eigen-spectrum so it suffices to prove the results for A. Let $\widehat{F}$ denote the unanimous locations in $c_i^{(1)}$ (i.e. the non-zero locations in c_S along with the padding bits). Slightly abusing notation, we extend c_S to be a vector in $\{-1, 0, 1\}^{16n}$ with $c_{Sj} = 1$ for $j \geq n$ as all $c_i^{(1)}$'s have a 1 in the padding locations. The first lemma says the the top eigenvector must agree with c_S in sign on $\widehat{F}$, and moreover must be non-negligible on these coordinates.

LEMMA 14. *Let v_1 be the top right singular vector of A such that $\sum_{j=n+1}^{16n} v_1 j \geq 0$. Then for any $j \in \widehat{F}$, $sgn(v_{1j}) = c_{Sj}$ and $|v_{1j}| \geq \frac{1}{40\sqrt{n}}$.*

PROOF. Let $a_i = c_i^{(2)} \in \Re^{16n}$. Since $\sum_i \langle a_i, v_1 \rangle^2 \geq \frac{15m}{16}$, it follows that at least for one i, it is the case that $\langle a_i, v_1 \rangle^2 \geq \frac{15}{16}$. Since $a_i|_{[n]}$ has norm $\frac{1}{4}$, it follows that the contribution to the dot product from the pad bits is at least $\frac{\sqrt{15}-1}{4}$. This in turn implies that for all i, $\langle a_i, v_1 \rangle \geq \frac{\sqrt{15}-2}{4} \geq \frac{1}{4}$.

Let $j \in \widehat{F}$ with $c_{Sj} = 1$ and suppose that $v_{1j} \leq \frac{1}{40\sqrt{n}}$. Let $e_j \in \Re^{16n}$ be a vector with one only in the j-th coordinate. We will argue that if v_1 is nearly orthogonal to e_j, then rotating v_1 slightly in the e_j direction gives a better Raleigh quotient, contradicting the optimaility of v_1. Indeed let $e'_j = e_j/100\sqrt{n}$. Thus $\langle e'_j, v_1 \rangle \leq \frac{1}{4000n}$, which implies that $\|v_1 + e'_j\|_2^2 \leq \|v_1\|_2^2 + \|e'_j\|_2^2 + 2\langle e'_j, v_1 \rangle \leq 1 + \frac{1}{10000n} + \frac{2}{4000n} \leq 1 + \frac{6}{10000n}$. On the other hand, $\langle a_i, e'_j \rangle \geq \frac{1}{400n}$ for each i, so that $(\langle a_i, (v_1 + e'_j) \rangle)^2 - \langle a_i, v_1 \rangle^2) \geq \frac{1}{400n} \cdot \frac{1}{4} \geq \frac{1}{1600n}$. In other words $\|A(v_1 + e'_j)\|_2^2 \geq \|Av_1\|_2^2(1 + \frac{1}{1600n})$, contradicting the optimality of v_1. The case of $c_{sj} = -1$ is identical. □

LEMMA 15. *For the matrix A as defined, $\sigma_1^2 \geq \frac{15m}{16}$. Thus $\sigma_1^2 - \sigma_2^2 \geq \frac{7m}{8}$.*

PROOF. The vector v_{pad} that is zero of the first n coordinates, and equals $pad/\sqrt{16n}$ on the remaining coordinates has norm less than 1 and gives $\|Av_{pad}\|_2^2 = \frac{15nm}{16n}$. This implies the first part of the lemma. The second part follows from noting that the sum of all σ_i^2 is m. □

Let v be a γ-useful vector output by the Algorithm $\mathcal{M}$ and let $w = Pv$. Let v_1 be the top right singular vector of A. From Lemma 15, it follows that, $\langle w, v_1 \rangle^2 \geq (1 - 4\gamma/3)$ so that $\|w - v_1\|_2^2 \leq 8\gamma/3$. By Lemma 14, every coordinate in $\widehat{F}$ such that $sgn(w)_j$ is different from c_{Sj} contributes $\frac{1}{1600n}$ to the squared distance $\|w - v_1\|^2$. It follows that the sign is wrong on at most $(1600n)(8\gamma/3) = 12800\gamma n/3$ of the $\widehat{F} \geq 15n$ coordinates.

The permutation P being random and unknown to the mechanism $\mathcal{M}$ ensures that the fraction of mistakes on F is not too different from that on $\widehat{F}$. Formally, call a co-ordinate in $\widehat{F}$ bad if $sgn(w)_j \neq c_{Sj}$. Recall that P randomizes both the location and the sign of the bits in c_{Sj}. Thus from the point of view of $\mathcal{M}$, F is a random subset of $\widehat{F}$ of size $|F|$. Thus the number of bad co-ordinates in F is expected to be at most $(\frac{12800\gamma n}{3})(|F|/15n)$. Except with probability ξ, $|F| \geq f$. Moreover by concentration bounds for the hypergeometric distribution, the probability that the number of bad coordinates in F exceeds twice its expectation is at most $\exp(-\frac{1}{2}(\frac{12800\gamma}{45})^2 f)$. The claim follows.

Combining with Theorems 11 and 12, we get

COROLLARY 16. *There is a universal constant γ such that the following holds for $m = \gamma\sqrt{n/\log n}$. Let $\mathcal{M}$ be a$(1, 1/n^2)$-DP mechanism that takes as input an $m \times 16n$ matrix A with each row having norm at most 1, and outputs a unit vector v. Then the probability that $\mathcal{M}(A)$ is γm-useful is at most $\frac{1}{n}$.*

5.2 Interlude: The List Culling Game

To help understand the proof for the lower bound for the subspace estimation, we introduce the *List Culling Game*. In this game, Dave has a vector $v \in \{-1, 1\}^n$. Alice has a version $v' \in \{-1, 1, \star\}^n$ of v where f of the bits chosen at random have been replaced by $\star$; we will be interested in the setting where f is $o(n)$. Dave, without knowing which bits are erased, sends Alice a list $L = \{w_1, \ldots, w_{|L|}\}$ of $\{-1, 1\}^n$ vectors with the promise that at least one of the w_i's has Hamming distance at most βn from v for a small constant $\beta < 1/20$. Alice wins if she can fill in the $\star$'s with error rate smaller than $\frac{1}{3}$, else Dave wins. Clearly if L is allowed to be size 2^n, then Dave can send the list of all binary vector, thus leaking no information and making it very unlikely that Alice can win. We will be interested in the question: For what values of L can Alice win?

The most natural strategy for Alice is the *most-agreement-strategy*: find a w_i that has the largest agreement on the non-$\star$ locations of v' and fill in the $\star$'s using it. We next argue that this strategy fails for lists size $\binom{n}{f}$. Indeed consider the list containing all vectors at Hamming distance exactly f from v. This list contains the vector w that agrees with v' on all non-$\star$ locations, and hence will be the one picked by the most-agreement-strategy. However, this vector w is wrong everywhere on the $\star$ locations!

This most-agreement-strategy for Alice thus fails badly once $L \geq \binom{n}{f}$. One may conjecture that beyond this threshold, Alice cannot win and instead Dave has a strategy that wins with non-negligible probability. We show that this conjecture is false: there is a strategy for Alice that wins with high probability even when the list size L is $\exp(cn)$ for some constant c.

The somewhat counter-intuitive strategy for Alice is as follows: she picks a random half of the non-$\star$ locations and finds a w_i that maximizes the agreement on this subset. This *most-agreement-on-random-half* strategy thus uses only half the information that Alice has about v! Consider a specific w_i that has Hamming distance more than $2\beta n$ from v, and let w^* be the promised vector in L that has Hamming distance at most βn from v. Alice tests w_i and w^* on a random $\frac{n-f}{2}$ subset, and the probability that w_i has larger agreement than w^* on this random subset is at most $\exp(-\Omega(\beta^2 n))$. Thus the probability that any w_i with Hamming distance larger than $2\beta n$ is chosen by the most-agreement-on-random-half strategy is $L\exp(-\Omega(\beta^2 n))$. Finally, if the chosen w_i has Hamming distance less than $2\beta n$ from v, the probability that it has disagreement more than $5\beta f$ on the $\star$ locations is at most $\exp(-\Omega(\beta^2 f))$. Thus for list size up to $\exp(cn)$ for a constant c, Alice wins with high probability. We have thus argued that

THEOREM 17. *There is an absolute constant $\gamma > 0$ such that for any f and large enough n, the following holds. There is a strategy for Alice in the list culling game such that for any valid list L of size $\exp(\gamma n)$, Alice wins with probability at least $1 - \exp(-\gamma f)$.*

5.3 Lower bound for subspace estimation

We say a k-dimensional projection matrix Π_k v is an α-useful rank-k subspace for a matrix A if $\|\Pi_k A^T\|_F^2 \geq \|\Pi'_k A^T\|_F^2 - \alpha$ for any rank-k projection matrix Π'_k. The main result of this section is analagous to the result for private eigenvectors. To get this result, we combine k of the $m \times 16n$ matrices from the previous section into one $km \times 16n$ matrix. When k is small (at most n/m) we can rotate these k matrices so that their spans are all orthogonal and they do not interfere with each other and the "loss" of about m from each of them results in a total loss of km. For larger k, some interference is unavoidable, but rotating them in random directions suffices to make them nearly orthogonal; this is the content of Lemma 20. Additionally, the eigenvalue separation result of the previous section is not sufficient any more as we output a k-dimensional subspace instead of a vector. We end up needing tighter control on the second (and thus smalller) eigenvalue of A, which we obtain in Lemma 19 by using the specific construction of Tardos and results from random matrix theory. A bigger difficulty comes from the fact that the output is now a k-dimensional subspace rather than a vector, and we need to extract a vector in this subspace that we will round to a β-valid vector for S. In the vector case, we used the padding bits to pick between w and $-w$; now we use them to pick amongst an $\exp(O(k))$-sized net of the subspace. This is where the List Culling Game is useful: the usefuleness of the subspace guarantees that one of these net points, appropriately rounded is β-valid. Using half of the padding bits to pick out the correct one allows us to complete the proof. Full details follow.

THEOREM 18. *There are universal constants K, K' such that the following holds for any $k \leq n/K$. Suppose there is an (ϵ, δ)-DP mechanism that for any matrix $A \in \Re^{m\times 16n}$ with with each row having norm at most 1 outputs a γkm-useful rank-k projection matrix Π_k A with probability p. Then there is an (ϵ, δ)-DP mechanism that on a sample $S = \{c_1, \ldots, c_m\}$ from an $(m + 1, n, f, \beta_0, \xi)$-fingerprinting code outputs a c' that is $K\gamma$-valid for S with probability $K'\gamma p - \xi - \exp(-\Omega(\gamma^2 f))$.*

PROOF. Let $\mathcal{D}$ be the distribution of the fingerprinting code and let $S_1, \ldots, S_{k-1}$ be $k-1$ fresh independent samples from $\mathcal{D}$ and let $S_k = S$. Thus the S_i's are identically and independently distributed. We randomly permute the indices so that S is indistinguishable from any other sample S_j. We will show a mechanism that outputs a $K\gamma$-valid codeword for S with non-trivial probability.

Towards that goal, we transform each $S_i = \{c_{i1}, \ldots, c_{im}\}$ to a matrix $A^{(i)}$ in a manner similar to the proof of Theorem 13. We first set $pad = 1^{15n}$ and append it to each of the c_{ij}'s to get $c_{ij}^{(1)}$. We then scale each vector to get a unit vector, thus setting $c_{ij}^{(2)} = c_{ij}^{(1)}/\sqrt{16n}$. Let $A^{(i)}$ be the matrix with rows $c_{ij}^{(2)}$. Next, we pick a random permutation matrix $P^{(i)}$ with a random sign on each entry, and a random rotation matrix $R^{(i)}$ and set $B^{(i)} = A^{(i)}P^{(i)}R^{(i)}$. Thus $B^{(i)}$ is obtained by randomly permuting the columns of $A^{(i)}$, randomly flipping the sign of each column, and then randomly ro-

Algorithm 4 Pirate algorithm A_{pirate}

Input: Set of codewords $S = \{c_1, \ldots, c_m\}$ with $c_i \in \{-1, 1\}^n$. Oracle access to Mechanism $\mathcal{M}$ for privately computing top k subspace of a matrix. Sampling access to distribution $\mathcal{D}$ from which S is sampled.

for $i = 1 \ldots k$ **do**
  Sample $S_i = \{c_{i1}, \ldots, c_{im}\}$ from $\mathcal{D}$.
end for
Pick r uniformly at random from $[k]$ and set $S_r \leftarrow S$.
Let $pad \leftarrow 1^{15n}$.
for $i = 1 \ldots k$ **do**
  for $j = 1 \ldots m$ **do**
    Let $c_{ij}^{(1)} \leftarrow c_{ij} \circ pad$.
    Let $c_{ij}^{(2)} \leftarrow c_{ij}^{(1)} / \sqrt{16n}$.
  end for
  Let $P^{(i)}$ be a random permutation matrix. Replace each 1 in P by a -1 with probability $\frac{1}{2}$.
  Let $A^{(i)}$ be the $m \times n$ matrix with the transposes of $c_i^{(2)}$ as it's rows.
  Let $R^{(i)}$ be a random $n \times n$ rotation matrix.
  Let $B^{(i)} \leftarrow A^{(i)} P^{(i)} R^{(i)}$.
end for
Let B be formed by vertically concatenating $B^{(i)}$'s for $i = 1, \ldots, k$ in random order.
Let $\Pi_k \leftarrow \mathcal{M}(B)$ be the γmk-useful rank-k projection matrix output by $\mathcal{M}$.
Let $\Pi_k^{(r)} = \Pi_k (R^{(r)})^T (P^{(r)})^T$.
Let $\theta \leftarrow \frac{1}{80\sqrt{n}}$.
Let w be the vector in $Span(\Pi_k^r)$ such that $\sum_{j=8n+1}^{16n} \mathbb{1}(|w_j| \geq \theta)$ is maximized.
for $j = 1 \ldots n$ **do**
  $c'_j = sgn(w_j)$.
end for
return c'

tating the rows of the resulting matrix[5]. Finally, we set B to the $km \times 16n$ matrix formed by vertically concatenating the $B^{(i)}$'s.

Let Π_k be the γkm-useful rank-k projection matrix returned by our private mechanism on input B. We will postprocess Π_k to construct a valid pirate codeword. Let $S = S_r$ and let $\Pi_k^{(r)} = \Pi_k (R^{(r)})^T (P^{(r)})^T$. For a vector v, a parameter θ, and a location j, we say that v θ-agrees with c_S in location j if $c_{Sj} v_j \geq \theta$. Let $H = \{8n + 1, \ldots, 16n\}$ be the second half of the indices, all corresponding to padding bits. Let w be a unit vector in $Span(\Pi_k^{(r)})$ such that w $\frac{1}{80\sqrt{n}}$-agrees with c_S in the maximum number of indices in H. We strip off the padding from w and set $c'_j = sgn(v_j)$. We note that this process did not use S except through the differentially private output Π_k, and hence the mechanism that outputs c' is differentially private. We now argue that c' is $K\gamma$-valid with non-trivial probability, for a suitable constant K.

Let $v_1^{(i)}$ denote the top right singular vector of $B^{(i)}$ and recall from Lemma 15 that $\sigma_1(B^{(i)})^2 = \|B^{(i)} v_1^{(i)}\|_2^2 \geq \frac{15m}{16}$. Thus the projection matrix $\tilde{\Pi}_k$ that projects to the span of $\{v_1^{(i)}\}_{i=1}^k$ satisfies $\sum_i \|\tilde{\Pi}_k (B^{(i)})^T\|_F^2 \geq \sum_i \sigma_1(B^{(i)})^2 \geq \frac{15mk}{16}$. Let $loss_i = \sigma_1(B^{(i)})^2 - \|\Pi_k (B^{(i)})^T\|_F^2$. Then the γkm-usefuleness of Π_k implies that $\mathbb{E}_i[loss_i] \leq \gamma m$. Each $loss_i \in [-m/16, m]$ and so it is easy to check, using arguments similar to Markov's inequality, that $\Pr_i[loss_i \geq 4\gamma m] \leq 1 - \gamma$. We omit details.

We will in fact need a stronger version of Lemma 15 to bound $\sigma_2(B^{(i)}) = O(1)$. The proof uses the particular construction of fingerprinting codes by Tardos [32] and standard results in random matrix theory.

LEMMA 19. *Let $B^{(i)}$ be constructed as above, starting with an $S_i = \{c_{i1}, \ldots, c_{im}\}$ drawn from the fingerprinting code ensemble of [32]. Then there are universal constants K_1, K_2 such that for all $s \geq C_1$, $\Pr[\sigma_2(B^{(i)})^2 \geq s^2] \leq K_1 \exp(-K_2 sn)$.*

Let us condition on the event that for all i, $\sigma_2^2(B^{(i)}) \leq K_1$; by Lemma 19, this event happens except with negligible probability. The following lemma says that on average over i, the span of $B^{(i)}$ has a small projection on Π_k; in fact it says that the average projection is small for *any* k-dimensional subspace. The proof uses the fact that the rotations $R^{(i)}$ are random and independent, and standard tail bounds along with a net argument.

LEMMA 20. *For $i = 1, \ldots k$, let $\{v_{ij}\}_{j=1}^m$ be a collection of orthogonal unit vectors in $\Re^n$ and let $R^{(i)}$'s be a independent random rotation matrices. Then for a universal constant K,*

$$\Pr[\exists \Pi_k : \sum_{i=1}^{k} \sum_{j=1}^{m} \|\Pi_k R^{(i)} v_{ij}\|_2^2 \geq Kk(1 + (km/n))] \leq e^{-\Omega(n)}.$$

Let $x_i = \|\Pi_k v_1^{(i)}\|_2^2$. Let y_i be the total squared projection of the remaining $(m-1)$ right singular vectors of $B^{(i)}$ onto Π_k, i.e. $y_i = \sum_{j=2}^m \|\Pi_k v_j^{(i)}\|^2$. Thus $\|\Pi_k (B^{(i)})^T\|_F^2 \leq \sigma_1^2 x_i + \sigma_2^2 y_i$.

Using lemma 20 with v_{ij}'s being the eigenvectors of $A^{(i)}$, we conclude that $\mathbb{E}_i[x_i + y_i] = K(1 + (km/n))$. Thus by Markov's inequality, at least a $(1 - \gamma/2)$ fraction of the i's satisfy $x_i + y_i \leq (2K/\gamma)(1 + (km/n))$. It follows that for at least a $\gamma/2$ fraction of the i's,

1. $loss_i \leq 4\gamma m$, and
2. $x_i + y_i \leq (2K/\gamma)(1 + (km/n))$.

Since $S_r = S$ has the same distribution as every other S_i, it follows that this property holds for r with probability at least $\gamma/2$. Let us condition on this event. For the rest of the proof, we will use $A, B, \sigma_1, \sigma_2, x, y, loss$, etc. to denote $A^{(r)}, B^{(r)}, \sigma_1(B^{(r)}), \sigma_2(B^{(r)})$, $x_r, y_r, loss_r$, etc. Thus as long as $k \leq \gamma^2 n / 16KK_1$, and m is at least some absolute constant,

$$\begin{aligned} 4\gamma m &\leq loss \\ &= \sigma_1^2 - \|\Pi_k B^T\|_F^2 \\ &\geq \sigma_1^2 - \sigma_1^2 x - \sigma_2^2 y \\ &\geq (1 - x)(15m/16) - K_1 (2K/\gamma)(1 + (km/n)) \\ &\geq (1 - x)(15m/16) - \gamma m/8. \end{aligned}$$

It follows that $(1 - x) \leq 5\gamma$ and thus there exists a unit vector $\tilde{v}_1 \in Span(\Pi_k)$ such that $\|v_1 - \tilde{v}_1\|_2^2 \leq 10\gamma$.

We call a vector $v \in Span(\Pi_k^{(r)})$ (θ, β)-good for a set of indices I if v θ-agrees with c_S in a $(1 - \beta)$-fraction of the indices in I.

CLAIM 21. *If v (θ, β)-agrees with c_S on I but v' does not $(\theta - \theta', \beta + \beta')$-agree with c_S on I. Then $\|v - v'\|^2 \geq \theta'^2 \beta' |I|$.*

Using Lemma 14, it follows that v_1 $(\frac{1}{40\sqrt{n}}, 0)$-agrees with c_{Sj} on H. Let $\beta = 2000^2 \gamma$. The vector w found by our mechanism must therefore be $(\frac{1}{80\sqrt{n}}, \beta)$-good for H.

[5]While this distribution is identical to that obtained by just applying R, it will be convenient in our proof to separate out the randomness in this fashion.

Let $\widehat{F}$ denote the unanimous locations in c_S (i.e. the unanimous locations $F(S)$ in $c_1, \ldots, c_m$ along with the padding bits). Recall that $H = \{8n+1, \ldots, 16n\}$ is the second half of the locations. From the point of view of the algorithm, the locations in H are indistinguishable from those in $\widehat{F} \setminus H$ and this will allow us to use arguments analagous to the list culling game. We know that w is $(\frac{1}{80\sqrt{n}}, \beta)$-good for H and we would like to argue that except with negligible probability, it is $(\frac{1}{320\sqrt{n}}, 5\beta)$-good for $\widehat{F} \setminus H$. Let N be a γ-net of the set of unit vectors in $Span(\Pi_k^{(r)})$. For any net point that is $(\frac{1}{160\sqrt{n}}, 3\beta)$-bad for $\widehat{F}$, the probability (taken over the randomness in P) that it is $(\frac{1}{160\sqrt{n}}, 2\beta)$-good for H is no larger than $\exp(-\Omega(\beta^2 n))$. Taking a union bound over a $\exp(O(k\log(1/\gamma)))$ points in N, we conclude that except with negligible probability, every $v \in N$ that is $(\frac{1}{160\sqrt{n}}, 2\beta)$-good for H is also $(\frac{1}{160\sqrt{n}}, 4\beta)$-good for $\widehat{F} \setminus H$.

Since w is $(\frac{1}{80\sqrt{n}}, \beta)$-good for H, then its nearest net point w' is $(\frac{1}{160\sqrt{n}}, 2\beta)$-good for H. Thus w' is $(\frac{1}{160\sqrt{n}}, 4\beta)$-good for $\widehat{F} \setminus H$, which in turn implies that w itself is $(\frac{1}{320\sqrt{n}}, 5\beta)$-good for $\widehat{F} \setminus H$.

Finally, since F is itself a random subset of $\widehat{F} \setminus H$, w is $(\frac{1}{320\sqrt{n}}, 6\beta)$-good for F except with probability $\exp(-\Omega(\beta^2 f))$. This then implies that the output of the mechanism is 6β-valid with probability $\Omega(\gamma) - \exp(-\Omega(f)) - \exp(-\Omega(n))$, completing the proof of Theorem 18.[6] $\square$

COROLLARY 22. *There is a universal constant γ such that the following holds for any $k \leq \gamma n$ and for $m = \gamma\sqrt{n/\log n}$. Let $\mathcal{M}$ be a $(1, 1/n^2)$-DP mechanism that takes as input an $km \times 16n$ matrix A with each row having norm at most 1, and outputs a rank k projection matrix. Then the probability that $\mathcal{M}(A)$ is γkm-useful is at most $\frac{1}{n}$.*

Acknowledgements

We are grateful to many colleagues for the generosity in sharing their insights and time: Moritz Hardt, Prateek Jain, Ravi Kannan, Frank McSherry, Sasho Nikolov, Adam Smith, and Jonathan Ullman. We would also like to thank especially the authors of [6] for sharing their manuscript.

6. REFERENCES

[1] D. Achlioptas and F. Mcsherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54, 2007.

[2] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.

[3] J. Bentley and J. Saxe. Decomposable searching problems i: Static-to-dynamic trnasformation. *J. Algorithms*, 1, 1980.

[4] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *PODS*, 2005.

[5] D. Boneh and J. Shaw. Collusion-Secure Fingerprinting for Digital Data. *IEEE Transactions on Information Theory*, 44:1897–1905, 1998.

[6] M. Bun, J. Ullman, and S. Vadhan. Fingerprinting codes and the price of approximate differential privacy. Manuscript, 2013.

[7] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 2009.

[6] The mechanism can be easily modified so that it either outputs a c', or FAIL and is correct except with negligible probability when it does not output FAIL. This can be done by testing (privately) whether the conditions 1 and 2 above hold approximately for r, so that it outputs FAIL with probability at most $(1-\gamma)$.

[8] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 2010.

[9] K. Chaudhuri, A. D. Sarwate, and K. Sinha. Near-optimal differentially private principal components. In *NIPS*, 2012.

[10] C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 1970.

[11] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[12] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Symp. Theory of Computing (STOC)*, pages 371–380, 2009.

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

[14] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 715–724. ACM, 2010.

[15] C. Dwork, M. Naor, O. Reingold, G. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390, 2009.

[16] C. Dwork and A. Roth. Algorithmic foundations of differential privacy, 2014. Monograph in preparation.

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

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

[19] E. Hazan, S. Kale, and M. K. Warmuth. Corrigendum to "learning rotations with little regret" september 7, 2010. 2010.

[20] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 2005.

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

[22] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

[23] F. McSherry. Spectral methods for data analysis. 2004.

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

[25] M. Mohri and A. Talwalkar. Can matrix coherence be efficiently and accurately estimated? In *International Conference on Artificial Intelligence and Statistics*, 2011.

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

[27] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 2011.

[28] A. Smith and A. Thakurta. Follow the perturbed leader is differentially private with optimal regret guarantees. *Manuscript in preparation*, 2013.

[29] G. Stewart and J. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[30] A. Talwalkar and A. Rostamizadeh. Matrix coherence and the Nystrom method. *arXiv preprint arXiv:1004.2008*, 2010.

[31] T. Tao. *Topics in random matrix theory*, volume 132. AMS Bookstore, 2012.

[32] G. Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2), 2008.

[33] M. K. Warmuth and D. Kuzmin. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 2008.

[34] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.