A Framework for Private Matrix Analysis in Sliding Window Model

Jalaj Upadhyay [*1] **Sarvagya Upadhyay** [*2]

Abstract

We perform a rigorous study of private matrix analysis when only the last W updates to matrices are considered useful for analysis. We show the existing framework in the non-private setting is not robust to noise required for privacy. We then propose a framework robust to noise and use it to give first efficient $o(W)$ space differentially private algorithms for spectral approximation, principal component analysis (PCA), multi-response linear regression, sparse PCA, and non-negative PCA. Prior to our work, no such result was known for sparse and non-negative differentially private PCA even in the static data setting. We also give a lower bound to demonstrate the cost of privacy.

1. Introduction

Matrix analysis manifests itself in many walks of life such as financial transactions, recommendation system, social networks, machine learning, and learning kernels. In the recent past, there has been a paradigm shift in matrix analysis in the era of big data. Two aspects that have become increasingly important are (i) protecting sensitive information and (ii) the increasing frequency with which data is being continuously updated. An example that illustrates the importance of these two aspects arises in several investment strategies in a financial firm. The strategies rely on matrix analysis (such as principal component analysis) of financial data that get continuously updated. Most of these strategies make use of "recent data" as opposed to the entire history. This heuristic is rooted in the empirical observation that recent data are better predictors of the future behavior of assets than older data (Moore et al., 2013; Tsay, 2005), a theme also found in many other applications of matrix analysis (Campos et al., 2014; Quadrana et al., 2018).

Moreover, the strategies are sensitive and have to be kept private. It is well documented that performing statistical analysis, including matrix analysis, accurately can leak private information (Narayanan & Shmatikov, 2006). As a result, privacy preserving algorithms for matrix analysis with robust privacy guarantees such as *differential privacy* are known (Amin et al., 2019; Blum et al., 2005; Dwork et al., 2014; Kapralov & Talwar, 2013; McSherry & Mironov, 2009; Hardt & Price, 2014; Hardt & Roth, 2012; Upadhyay, 2018)). However, these algorithms are not amenable to the scenario where a collection of the most recent updates on data is pertinent for analysis. In contrast, the current practical deployment of private algorithms (Erlingsson et al., 2014; Thakurta et al., 2017) favors using only recent data for a variety of reasons.

In view of this, we focus on a rigorous and comprehensive study of privacy-preserving matrix analysis in the *sliding window model of privacy* (Bolot et al., 2013; Chan et al., 2012; Upadhyay, 2019). The model is parameterized by the window size W, and assumes that the data arrive in the form of (possibly infinite) stream over time. An analyst is required to perform the analysis only on the W most recent streams of data (usually referred to as a *sliding window*) using $o(W)$ space. On the other hand, privacy is guaranteed for the entire historical data, i.e., even if the data is not in the current window, its privacy should not be compromised.

We give $o(W)$ space differentially private algorithms for several matrix analysis problems in the sliding window model (see, Table 1). Here and henceforth, $o(W)$ will ignore other factors such as matrix dimensions and privacy parameters.

A brief overview of our main contributions are as follows (and annotate each of the points below with the corresponding appendix in the supplementary material).

1. **(Limitations of known framework and algorithm).** We show that existing framework of *spectral histogram* used in the non-private setting (Braverman et al., 2020) is too stringent for privacy and algorithms in that framework are not robust to perturbation required for privacy. We show rigorously that the strict constraint imposed by spectral histogram only permits sub-optimal accurate private algorithms (Appendix B). That is, adding appropriately scaled noise to the algorithm of (Braverman et al., 2020) does not suffice. This warrants a robust framework for private matrix analysis.

[*]Equal contribution [1]Apple, USA (work done when the author was between jobs). [2]Fujitsu Research of America, USA. Correspondence to: Jalaj Upadhyay <jalaj@apple.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

	Privacy	Additive error	Space required	Reference
η-spectral approximation	(ϵ, δ)-DP	$O\left(\frac{r^2 \log^2(1/\delta)}{\epsilon^2}\right) \mathbb{1}_d$	$O\left(\frac{r^2 d}{\eta} \log W\right)$	Theorem 13
Principal component analysis (PCA)	(ϵ, δ)-DP	$O\left(\frac{\sqrt{kd}\log(1/\delta)}{\epsilon}\right)$	$O\left(\frac{dk^2}{\eta^3} \log W\right)$	Theorem 16
Sparse and Non-negative PCA	(ϵ, δ)-DP	$O\left(\frac{\sqrt{kd}\log(1/\delta)}{\epsilon}\right)$	$O\left(\frac{dk^2}{\eta^3} \log W\right)$	Theorem 17
Multiple linear regression	(ϵ, δ)-DP	$O\left(d\left(d + \frac{\log(1/\delta)}{\epsilon}\right)\right)$	$O\left(\frac{d^3}{\eta} \log W\right)$	Theorem 18
Directional variance query	(ϵ, δ)-DP	$O\left(d\left(d + \frac{\log(1/\delta)}{\epsilon}\right)\right)$	$O\left(\frac{d^3}{\eta} \log W\right)$	Theorem 11

Table 1. Results presented in this paper (W: window size, k : target rank, d : dimension of streamed row, privacy parameters (ϵ, δ), $\mathbb{1}_d$ is a $d \times d$ identity matrix, r: rank of streamed matrix).

2. **(New framework and data structure).** We introduce a relaxation of spectral histogram property on a set of positive semidefinite (PSD) matrices that is more robust to noise and call it *approximate spectral histogram property*. We also design an update time efficient data structure that maintains the approximate spectral histogram property on a set of PSD matrices while preserving differential privacy (Appendix C).

3. **(Optimal algorithms for matrix analysis).** We use approximate spectral histogram property to efficiently compute private spectral approximation. Using this, we solve several matrix analysis problems in the sliding window model while preserving privacy and optimal accuracy in Appendix D: (i) principal component analysis (PCA); (ii) directional variance queries; and (iii) multi-response linear regression. We also give algorithm for private *constrained PCA* (Cohen et al., 2015). This generalizes many variants of PCA studied in statistical machine learning such as sparse PCA and non-negative PCA.

4. **(Limitation of private sliding window algorithms).** Finally, to complete the picture, we exhibit limitations of private matrix analysis by giving a lower bound on differentially private algorithm for low-rank approximation in the sliding window model (Appendix E).

There is a known separation between what is achievable with privacy and without privacy for real-valued functions in the sliding window model (Upadhyay, 2019). Our work can be seen as extending this study to matrix-valued functions in a unified manner. Conceptually, approximate spectral histogram property can be viewed as a generalization of *subspace embedding property* (Sarlós, 2006). This allows us to use approximate spectral histogram property in the sliding window model in the same way as subspace embedding is employed in the streaming model of privacy (Upadhyay, 2018). Given the wide application of subspace embedding in streaming algorithms, we believe that the notion of approximate spectral histogram will have further applications in the sliding window model of privacy.

A natural question one may ask is why we need to introduce approximate spectral histogram property in the sliding window model of privacy. We end this section with a discussion on this (more details in Section 2). Let us consider the spectral approximation of matrices. There is one private algorithm (Blocki et al., 2012) which relies on subspace embedding. They explicitly compute the singular value decomposition of the matrix making it suitable only for static data matrix. Furthermore, we cannot just take off-the-shelf algorithm and add noise matrix to preserve privacy as well as guarantee non-trivial utility and efficiency. To begin with, standard noise mechanisms would result in a matrix that is not positive semidefinite. This is, for example, the mechanism in Dwork et al. (2014). If we instead use the projection trick of Arora & Upadhyay (2019) on top of Dwork et al. (2014), it would incur noise that scales with the dimension and have an inefficient update time. Moreover, the existing randomized space-efficient algorithm of Braverman et al. (2020) performs sampling proportional to its *leverage score*. As a result, the effect of a single row in the matrix formed by this sampling procedure can be arbitrarily large, and consequently, the sensitivity is high[1].

Notations. For a natural number n, the notation $[n]$ denotes the set $\{1, \ldots, n\}$. The Euclidean norm of a vector $v \in \mathbb{R}^d$ is denoted by $||v||_2$. For a rank-r matrix $A \in \mathbb{R}^{n \times d}$, we let the tuple $(s_1(A), s_2(A), \ldots, s_r(A))$ denote the non-zero singular values of A arranged in decreasing order, $A^\top$ to denote transpose of A, and $\|A\|_F$ to denote its Frobenius norm. The i-th row vector and the j-th column vector of a matrix A are denoted by $A[i :]$ and $A[: j]$, respectively. We use $\|A[: j]\|_2$ and $\|A[i :]\|_2$ to denote their Euclidean norms. We use $\mathbb{1}_d$ to denote identity matrix of dimension d. If all the eigenvalues of a symmetrix matrix $S \in \mathbb{R}^{d \times d}$ are non-negative, then the matrix is known as *positive semidefinite* (PSD for short) and is denoted by $S \succeq 0$. For symmetric

[1]There are counterexamples where leverage score for a row can change arbitrarily depending on whether it is in the span of the current matrix or not (see for example, (Arora & Upadhyay, 2019) in the context of graph sparsification). In fact, it is not clear if we can even use the exponential mechanism because for most natural score functions, one can construct counterexamples where the sensitivity of the score function is also large.

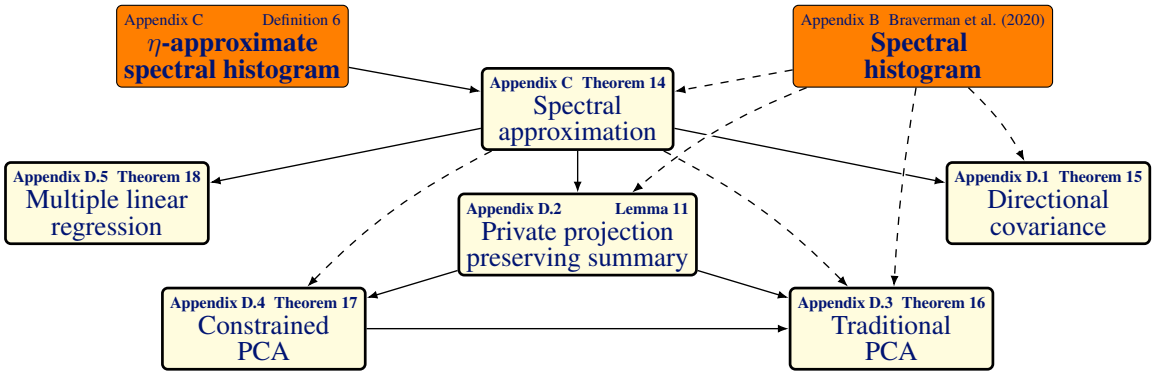

Figure 1. Dependency graph of various results (bold lines shows optimal results and dashed lines shows suboptimal results, orange boxes are datastructure). For example, a datastructure satisfying η-approximate spectral histogram implies an algorithm for spectral approximation, and so on. All our algorithms extend to the streaming model as well by setting $W = T$.

matrices $A, B \in \mathbb{R}^{d\times d}$, the notations $A \preceq B$ implies that $B - A$ is PSD and $A \not\preceq B$ implies that $B - A$ is not a PSD. For any $T, d > 0$, we use $\mathsf{N}_{T,d}$ to denote the following set of $T \times d$ matrices:

$$\mathsf{N}_{T,d} := \big\{B \in \mathbb{R}^{T\times d} : \exists i \in [T] \text{ such that } \|B[i:]\|_2 \le 1 \text{ and } \|B[j:]\|_2 = 0 \text{ for all } j \neq i\big\}.$$

A comprehensive overview of preliminaries and notations is presented in Appendix A.

1.1. Sliding window, privacy, and matrix analysis

We start by defining some additional notations pertinent to studying matrix analysis in the sliding window model. The matrix formed by d-dimensional row vectors streamed between time stamps t_1 and t_2 is denoted $A_{[t_1,t_2]}$. We define $A_W(T) := A_{[T-W+1,t]}$ for any current timestamp T where W is used to denote the window size and $A_T := A_{[0,T]}$. The matrix A_T can be obtained by setting $W = T$ and gives us the insertion only streaming model (Muthukrishnan, 2005). The matrix $A_W(T) \in \mathbb{R}^{W\times d}$ is formed incrementally through a stream of d-dimensional row vectors $\{a_i : T - W + 1 \le i \le T\}$ as follows:

$$A_W(T) := \begin{pmatrix} a_{T-W+1} \\ \vdots \\ a_{T-1} \\ a_T \end{pmatrix} \in \mathbb{R}^{W\times d}. \tag{1}$$

At start, the matrix $A_W(0)$ is an all zero matrix (with $a_i = 0^d$ if $i \le 0$). At any time T, we are interested in performing various analysis on the matrix $A_W(T)$. Our results are independent of the current time stamp T, and we will slightly abuse the notation by letting $A_W = A_W(T)$ as the matrix formed by rows streamed in the last W updates.

We now formalize the privacy model. We adhere to the neighboring relation employed in existing literature studying matrix analysis in static setting (Blocki et al., 2012; Hardt & Roth, 2012; Dwork et al., 2014; Sheffet, 2019) and streaming setting (Upadhyay, 2018).

In privacy literature, there are two well-studied levels of granularity when the data arrives in an online manner (Bolot et al., 2013; Chan et al., 2011; 2012; Dwork et al., 2010; Dwork & Roth, 2014; Upadhyay, 2018; 2019): (i) *user-level privacy*, where two streams are neighboring if they differ in a single user's data; and (ii) *event-level privacy*, where two streams are neighboring if they differ in one-time epoch. We follow previous works on private analysis in the sliding window model (Bolot et al., 2013; Chan et al., 2012; Huang et al., 2021) and consider event-level privacy. We say that two streams are *neighboring* if, at any time $T > 0$, they form matrices A_T and A'_T such that $A_T - A'_T \in \mathsf{N}_{T,d}$. We now define the privacy notion that extends the privacy notion of Bolot et al. (2013); Chan et al. (2012); Huang et al. (2021); Upadhyay et al. (2021) and Upadhyay (2019) to general matrices.

Definition 1 (Differential privacy under sliding window model)**.** *For $\epsilon \ge 0, \delta \in [0, 1]$, we say a randomized algorithm $\mathcal{M}$ with range Y is (ϵ, δ)-differentially private in the sliding window model if for all $T > 0$, for every two matrices A_T and A'_T formed by neighboring streams, and for all $\mathsf{S} \subseteq \mathsf{Y}$, $\Pr[\mathcal{M}(A_T) \in \mathsf{S}] \le \exp(\epsilon)\Pr[\mathcal{M}(A'_T) \in \mathsf{S}] + \delta$, where the probability is over the private coin tosses of M.*

Note that the privacy guarantee is for the entire stream, i.e., even if the data has expired, its privacy is not lost. However, accuracy is required only for the last W updates. This is in accordance with previous problem formulation (Bolot et al., 2013; Chan et al., 2012; Upadhyay, 2019).

The central algebraic concept underlying all analysis of interest in this paper is the spectrum of a matrix (see Figure 1). Therefore, we focus on privately computing (η, ζ)-spectral approximation, i.e., given parameters $\eta, \zeta \ge 0$ and a matrix

$A_W \in \mathbb{R}^{W\times d}$, find a matrix $C \in \mathbb{R}^{d\times d}$, such that

$$(1-\eta)A_W^\top A_W - \nu \mathbb{1}_d \preceq C \preceq (1+\eta)A_W^\top A_W + \nu \mathbb{1}_d.$$

Here the parameter $\nu \geq 0$ is the cost of privacy in the terms of distortion in the spectrum. Our goal is to keep η as small as possible so that they are useful in subsequent tasks, like PCA, multiple regression, etc. We show the following:

Theorem 1 (Informal version of Theorem 14)**.** *Let $A_W \in \mathbb{R}^{W\times d}$ be a rank-r matrix formed by the current window. Then for $\nu = \xi \log \xi$ where $\xi = O\left(\frac{r\log^2(W/\delta)}{\epsilon^2\eta}\right)$, there is an efficient (ϵ,δ)-differentially private algorithm under sliding window model that uses $O\left(\frac{dr^2}{\eta^2}\log W\right)$ space and outputs a matrix C at the end of the stream such that*

$$(1-\eta)A_W^\top A_W - \nu \mathbb{1}_d \preceq C \preceq (1+\eta)A_W^\top A_W + \nu \mathbb{1}_d.$$

A special case when the matrix is the edge-adjacency matrices was considered by Upadhyay et al. (2021). In the static setting, using the result of Sarlós (2006) and Blocki et al. (2012), we get an $O(d^2)$ space private algorithm which guarantees (η,ν,ν)-spectral approximation for $\nu = O\left(\frac{d\log(1/\delta)}{\epsilon^2\eta}\right)$. Non-privately, there is an algorithm in the sliding window model that uses $O\left(\frac{rd}{\eta}\log W\right)$ space if the matrix has a bounded condition number (Braverman et al., 2020). In many practical scenarios, the rank is constant. In this scenario, the privacy overhead is only a constant factor. Our algorithm is also flexible in the sense that we can also guarantee that the output is a PSD matrix.

Before giving a technical overview of our private algorithm, we begin by arguing why the existing private algorithms in the static setting fail in the sliding window model. Blocki et al. (2012) gave the first privacy preserving approximation of matrices. Their approach is to first compute the singular value decomposition of the given matrix $A = USV^\top$, and then output $C_{\mathsf{BBDS}} = \widehat{A}^\top \Phi^\top \Phi \widehat{A}$, where $\widehat{A} := U\sqrt{S^2+\sigma^2\mathbb{1}_d}V^\top$ for a perturbation parameter σ chosen appropriately, and Φ is a random Gaussian matrix. Since, the algorithm requires computing the SVD, one cannot extend this approach in the sliding window model. Another approach, due to Dwork et al. (2014), computes $C_{\mathsf{DTTZ}} = A^\top A + N$, where N is a symmetric Gaussian matrix with appropriate variance. In this case, we cannot revert the effect of the rows outside of the window.

Private principal component analysis has been extensively studied (Amin et al., 2019; Blum et al., 2005; Dwork et al., 2014; Hardt & Roth, 2012; Upadhyay, 2018; Dwork et al., 2014; Hardt & Price, 2014; Kapralov & Talwar, 2013; Singhal & Steinke, 2021), and matching lower and upper bounds are known on achievable accuracy in the static setting. With the exception of Arora et al. (2018); Upadhyay (2018), these algorithms perform at least two passes over the matrix. Dwork et al. (2014) gave an online algorithm for PCA using regularized follow-the-leader framework; however, online model is very different from the sliding window model[2]. Finally, the algorithm of Arora et al. (2018) and Upadhyay (2018) does not extend to the sliding window model because we cannot revert the effect of the rows that are outside of the current window.

2. Main lemma and overview of techniques

One-shot vs Continual release. In this section, we focus only on the case when the output is produced just once at the end of the stream for the ease of presentation. Such algorithms are known as *one-shot algorithm* in the literature of differential privacy and used as a building block for algorithms that continually release statistics. We cover the case of *continual release* (Dwork et al., 2010) in Appendix F, where we propose two data structures that allow continual release depending on whether space is more important or accuracy. The first approach uses the binary tree method introduced by Bentley & Saxe (1980) and used in Dwork et al. (2010) and Chan et al. (2011). However, unlike them, we build the binary tree only over the current window. This uses space linear in W but incur error that only grows polylogarithmically. In the second approach, we reduce the space requirement to be sublinear in W at the cost of increasing the error. We subdivide each window in to $\sqrt{W}$ sub-windows, each of size $\sqrt{W}$. We then run an instance of our algorithm for each of these sub-windows.

We now focus our attention to design a one-shot algorithm. Algorithmically, our approach is closest to Smith et al. (2020). They present a one-shot space-optimal algorithm for *distinct element count* in a data-stream by showing that the celebrated Flajolet-Martin sketch initiated with some random "phantom" elements (guaranteed to be not in the data set) is differentially private. Similar approaches has been used for computing low-rank approximation of a matrix formed in a streaming manner (Upadhyay, 2018).

One-shot algorithm. Our one-shot algorithms (on which the continual release algorithms is based) can be seen as a generalization of the technique of Smith et al. (2020) from real-valued functions to matrix-valued functions. We inject an appropriate random matrix to the data stream. However, this would only allows us to perform the analysis on the entire data stream and not just on the current window. That is, we need to resolve the following two related questions:

[2]The online learning model is a game between a decision-maker and adversary. The decision-maker makes decisions iteratively. After committing to a decision, it suffers a (possible adversarially) loss. The goal is to minimize the total loss in retrospect to the best decision the decision-maker should have taken.

1. **(Question 1).** How to account only for only the relevant part of the streamed data, i.e., one in the window?
2. **(Question 2).** What distribution of random matrices is to be used to inject phantom random matrices?

One naive candidate algorithm, $\mathsf{A}_{\mathsf{priv}}$, for private spectral approximation is as follows: store a set of $w = \min\{W, T\}$ positive semidefinite matrices at any time T, where the i-th matrix in this set is a sanitized version of the matrix formed by the last i updates. In this case, question 1 is answered by just removing any matrix that is out of the window, and question 2 is answered by using Wishart matrix of appropriate scale. However, $\mathsf{A}_{\mathsf{priv}}$ requires prohibitively large $O(Wd^2)$ space.

To answer question 1, while using significantly less space (as in Smith et al. (2020)) requires a conceptual contribution. To this end, we introduce *η-approximate spectral histogram property* for a set of PSD matrices and timestamps. We will occasionally refer to such a set as a data structure.

η-approximate spectral histogram property. For a matrix $S \succeq 0$, denote by $\widetilde{S}$ a matrix such that

$$\left(1 - \frac{\eta}{4}\right)\widetilde{S} \preceq S \preceq \left(1 + \frac{\eta}{4}\right)\widetilde{S}.$$

Let the current window of our matrix analysis be from timestamp $T - W + 1$ to T and $S(i)$ be the covariance matrix of the matrix formed by rows streamed between timestamps t_i and T. In other words,

$$S(i) = A_{[t_i,T]}^{\top} A_{[t_i,T]}.$$

Let $\mathfrak{D}$ be a data structure comprised of a collection of ℓ timestamps and PSD matrices $\{(t_1, \widetilde{S}(1)), \ldots, (t_\ell, \widetilde{S}(\ell))\}$ for some $\ell \in \mathbb{N}$. For all $i \in [\ell]$, $\widetilde{S}_i$ is an $(\eta/4, 0)$-spectral approximation of the matrix S_i. Roughly speaking, such a data structure $\mathfrak{D}$ satisfies *η-approximate spectral histogram property* if following two listed properties are satisfied..

1. The timestamps satisfy the following two requirements:

$$t_1 < \cdots < t_\ell = T \quad \text{and} \quad t_1 \leq T - W + 1 \leq t_2.$$

2. These two sets of matrices $\{S(i)\}_{i\in[\ell]}$ and $\{\widetilde{S}(i)\}_{i\in[\ell]}$ satisfy the following three conditions:

$$\begin{aligned} &\forall i \in [\ell - 1], S(i+1) \preceq S(i); \\ &\forall i \in [\ell - 1], (1-\eta)S(i) \preceq S(i+1); \text{ and} \\ &\forall i \in [\ell - 2], \left(1 - \frac{\eta}{2}\right)\widetilde{S}(i) \not\preceq \widetilde{S}(i+2). \end{aligned} \tag{2}$$

When it is clear from context, we call a set of matrices $\left\{\widetilde{S}_1, \cdots, \widetilde{S}_\ell\right\}$ as the one satisfying the η-approximate spectral histogram property. In contrast, spectral histogram in Braverman et al. (2020) requires $\widetilde{S}(i) = S(i)$ and uses the condition $(1-\eta)S(i) \not\preceq S(i+2)$ instead of $\left(1 - \frac{\eta}{2}\right)\widetilde{S}(i) \not\preceq \widetilde{S}(i+2)$.

Algorithm 1 PHASE 2($\mathfrak{M}'_{T+1} = \left\{\bar{A}(1), \cdots, \bar{A}(\ell+1)\right\}$)

1: **If** $t_2 < T - W + 1$, set $\bar{A}(i) = \bar{A}(i+1), t_i = t_{i+1}$ for all $i \in [\ell - 1]$. Set $\ell = \ell - 1$
2: **Define** $\bar{S}(i) = \bar{A}(i)^{\top}\bar{A}(i)$ for all $1 \leq i \leq \ell + 1$.
3: **For** $i = 1, \cdots \ell - 2$
4: Find $j = \max\left\{u > i : (1 - \frac{\eta}{2})\bar{S}(i) \preceq \bar{S}(u)\right\}$.
5: Set $\mathfrak{M}'_{T+1} \leftarrow \mathfrak{M}'_{T+1} \backslash \{\bar{A}(i+1), \cdots, \bar{A}(j-1)\}$.
6: Reorder the indices of remaining matrices.
7: Update $\ell := \ell + i - j + 1$.
8: **Output** $\mathfrak{M}_{T+1} := \mathfrak{M}'_{T+1}$.

The properties in Equation 2 are required to get the desirable space bound. Likewise, the second condition in Equation 2 and the restriction $t_1 \leq T - W + 1 \leq t_2$ are required to demonstrate the accuracy guarantee (see proof sketch of Theorem 1). Before proving the accuracy guarantee, we answer how to maintain such a set of matrices. For brevity, we introduce the following notation for matrices in the rest of this section: for any time T, we write $A(i)$ to denote the i-th matrix stored in the current data-structure.

Lemma 1. *Let $\mathfrak{M}_T := \{A(1), \ldots, A(\ell)\}$ be the set of matrices such that $\{A(1)^{\top}A(1), \ldots, A(\ell)^{\top}A(\ell)\}$ satisfies η-approximate spectral histogram property at time T. Then there is an efficient algorithm,* UPDATE*, that takes $\mathfrak{M}_T$ and a row $a_{T+1} \in \mathbb{R}^d$ as input and outputs a set of matrices $\mathfrak{M}_{T+1} = \{B(1), \ldots, B(m)\}$, such that $\{B(1)^{\top}B(1), \ldots, B(m)^{\top}B(m)\}$ satisfy the η-approximate spectral histogram property for some $m \leq \ell + 1$.*

When a new row $a_{T+1} \in \mathbb{R}^d$ is streamed, an algorithm is invoked that updates the data structure. It works in two phases: *privatization* and *maintenance*. Privatization is accomplished by (i) adding a linear sketch of a_{T+1} to all ℓ matrices in $\mathfrak{M}_T$ to obtain a new set $\mathfrak{M}'_T$, (ii) privatizing a_{T+1} to get a matrix $A(\ell+1)$, and (iii) defining $\mathfrak{M}'_{T+1} := \mathfrak{M}'_T \cup A(\ell+1)$. For privacy (or answering Question 2), adding a noise matrix that is a PSD matrix would incur additive error linear in dimension. Moreover, it will not maintain structural properties of matrices such as low-rank, which are one of the reasons why matrix analysis have such a wide array of applications. Therefore, *just adding appropriately scaled noise is not an option* (see Appendix B for details). As it turns out, a variant of Johnson-Lindenstrauss mechanism (Blocki et al., 2012) used in Upadhyay (2018) suffices for our purpose.

Now the set $\{\bar{A}^{\top}\bar{A} : \bar{A} \in \mathfrak{M}'_{T+1}\}$ may not satisfy η-approximate spectral histogram property. The *maintenance*

phase (high-level description of this phase is provided in Algorithm 1) ensures that the final set of matrices satisfies η-approximate spectral histogram property. In this phase, we greedily remove matrices if they do not satisfy any of the desired properties of η-approximate spectral histogram property (Algorithm 7 in supplementary material). The computationally expensive part in Algorithm 1 is Step 3. For this step, we can use known PSD testing algorithms (Bakshi et al., 2020).

Our greedy approach is reminiscent of the *potential barrier method* to compute spectral sparsification of a $W \times d$ matrix (Batson et al., 2012). In the potential barrier method, we remove a large subset of rank-one matrices and show that only storing $\Theta\left(d\eta^{-2}\right)$ rank-one matrices suffices for $(\eta, 0)$-spectral sparsification. This approach does not extend over to streaming matrices. In fact, two key technical features distinguish our method from theirs. In their setting, all PSD matrices are rank-one matrices corresponding to a row of the matrix; whereas we have W positive semidefinite matrices that may have different ranks (not necessarily rank-one). The second crucial point is that we aim to significantly reduce the number of matrices stored for our application. This makes maintaining our data structure much more complicated than the potential barrier method.

The proof of Lemma 1 is subtle. While it is tempting to use the analysis of the deterministic algorithm by Braverman et al. (2020) in our setting, their analysis is highly susceptible to noise. Their proof relies heavily on the fact that for all $i \in [\ell], \widetilde{S}(i) = S(i)$, i.e., matrices are exact covariance matrices corresponding to the streamed rows. In contrast, our analysis deals with the spectral approximation of the streamed matrix along with the perturbation required to preserve privacy. That is, each of the matrices $\widetilde{S}(1), \cdots, \widetilde{S}(\ell)$ is an approximation of the input matrix and has both multiplicative approximation as well as additive term. We give an arguably simpler analysis than Braverman et al. (2020) and crucially use the slack of $\left(1 - \frac{\eta}{2}\right)$ factor in the third condition of approximation spectral histogram property (Equation 2). A detail proof of Lemma 1 is presented in Appendix C.

Spectral approximation. Now that we have an algorithm to maintain η-approximate spectral histogram property, we show how to use it to compute an (η, ν)-spectral approximation of A_W. Let $\widetilde{S}(1), \ldots, \widetilde{S}(\ell)$ be the set of matrices satisfying η-approximate spectral histogram property. The algorithm outputs $S = \widetilde{S}(1) - \sigma^2 \mathbb{1}_d$, where σ^2 is the perturbation posit in the mechanism of Upadhyay (2018). Using the first condition of Equation 2 and that $t_1 < T - W + 1 < t_2$, $S(2) \preceq A_W^\top A_W \preceq S(1)$. The second condition of Equation 2 implies that $(1-\eta)S(1) \preceq S(2)$. Since $\widetilde{S}(1)$ and $\widetilde{S}(2)$ are a $(\eta/4, 0)$-spectral approximation of $S(1)$ and $S(2)$, respectively, this allows us to prove that $\widetilde{S}(1)$ is a spectral approximation of A_W.

Proof sketch of Theorem 1. For space bound, properties in equation (2) imply that there is at least one singular value that decreases by a factor of $(1 - \frac{\eta}{2})$ in every successive timestamp. We will see later that our privacy mechanism ensures that the spectrum of any matrix $\widetilde{S}_i$ is lower bounded by a constant. Since updates have bounded entries, there can be at most $\ell := O\left(r \log_{1-\frac{\eta}{2}}(W)\right) = O\left(\frac{r}{\eta}\log(W)\right)$ matrices satisfying η-approximate spectral histogram. For privacy, we use the Johnson-Lindenstrauss mechanism (Blocki et al., 2012). In this mechanism, we first perturb the matrix to raise its singular value and then multiply it with a random Gaussian matrix. The choice of perturbation used here is the one described in Upadhyay (2018) because it can account for the streamed data.

Now we give a proof sketch of the accuracy guarantee. At any time T, let $A(i)$ be the matrix formed between the time interval $[t_i, T]$. Let $\{\widetilde{A}(1), \cdots, \widetilde{A}(\ell)\}$ be the set of matrices obtained by applying Johnson Lindenstrauss mechanism on the streamed matrices $\{A(1), \cdots, A(\ell)\}$ and $\{\widehat{A}(1), \cdots, \widehat{A}(\ell)\}$ be the set of perturbed matrices before applying the Johnson-Lindenstrauss transform. Fix the following notations for covariance matrices:

$$C(j) := A(j)^\top A(j), \quad \widetilde{S}(j) := \widetilde{A}(j)^\top \widetilde{A}(j)$$
$$S(j) := \widehat{A}(j)^\top \widehat{A}(j) = C(j) + \sigma^2 \mathbb{1}_d.$$

The perturbation parameter σ is as chosen in Sheffet (2019). Since $t_1 \leq T - W + 1 \leq t_2$, we have $C(2) \preceq A_W^\top A_W \preceq C(1)$. By design of our algorithm and the second property of η-approximate spectral histogram property, we have $(1-\eta)S(1) \preceq S(2)$. We pick the dimension of the Johnson-Lindenstrauss transform so that $\widetilde{S}(j)$ is an $(\eta/4, 0)$-spectral approximation of $S(j)$ for all $j \in [\ell]$ using Sarlós (2006)'s result. Therefore, for $i \in \{1, 2\}$,

$$\left(1 - \frac{\eta}{4}\right) S(i) \preceq \widetilde{S}(i) \preceq \left(1 + \frac{\eta}{4}\right) S(i).$$

This implies that $\left(1 - \frac{\eta}{4}\right)(C(1) + \sigma^2 \mathbb{1}_d) \preceq \widetilde{S}(1)$. Since adding positive semidefinite matrices preserves the Loewner ordering and $A_W^\top A_W \preceq C(1)$, we get the following:

$$\left(1 - \frac{\eta}{4}\right)(A_W^\top A_W + \sigma^2 \mathbb{1}_d) \preceq \left(1 - \frac{\eta}{4}\right)(C(1) + \sigma^2 \mathbb{1}_d) \preceq \widetilde{S}(1).$$

Similarly, for the upper bound, we have from the definition,

$$\widetilde{S}(1) \preceq \left(1 + \frac{\eta}{4}\right) S(1) \preceq \frac{\left(1 + \frac{\eta}{4}\right)}{(1-\eta)} S(2)$$
$$= \frac{\left(1 + \frac{\eta}{4}\right)}{(1-\eta)} (C(2) + \sigma^2 \mathbb{1}_d).$$

Using the fact that $C(2) \preceq A_W^\top A_W$, scaling η and setting the value of σ completes the proof. □

3. Applications

We present three main applications of η-approximate spectral histogram property for matrix analysis.

Applications I: Principal component analysis. Principal component analysis is an extensively used subroutine in many applications like clustering (Cohen et al., 2015), recommendation systems (Drineas et al., 2002), and learning distributions (Achlioptas & McSherry, 2005). In these applications, given a matrix $A \in \mathbb{R}^{n\times d}$ and a target rank k, the goal is to output a rank-k orthonormal projection matrix $P \in \mathbb{R}^{d\times d}$ such that

$$\|A - AP\|_F \leq (1+\eta) \min_{\mathsf{rank}(X)\leq k} \|A - X\|_F + \zeta.$$

The goal here is to minimize ζ for a given k, d, and privacy parameters ϵ and δ. In many applications, instead of optimizing over all rank-k projection matrices, we are required to optimize over a smaller set of projection matrices, such as one with only non-negative entries. In particular, let Π be any set of rank-k projection matrices (not necessarily set of all rank-k projection matrices). Then the *constrained principal component analysis* is to find $P^* = \operatorname{argmin}_{P\in\Pi} \|A - AP\|_F^2$.

A naive application of approximate spectral histogram property to solve PCA leads to an additive error that depends linearly on the rank of the streamed matrix. To solve these problems with optimal accuracy, we introduce an intermediate problem that we call *private projection preserving summary* (Definition 8). This problem can be seen as a private analogue of PCP sketches (Cohen et al., 2015). Solving this problem ensures that the additive error scales with the parameter k and not with the rank of the matrix.

To remove the dependency on the rank of the streamed matrix, we consider the first k/η spectrum of the streamed matrix and show that it suffices for our purpose. That is, let $\widetilde{A}_1, \cdots, \widetilde{A}_\ell$ be matrices such that their covariance matrices $\widetilde{S}_1, \cdots, \widetilde{S}_\ell$ satisfy η-approximate spectral histogram property. We show that random projections of $\widetilde{A}_1, \cdots, \widetilde{A}_\ell$ to a k/η dimensional linear subspace suffice. Let $\pi_{k/\eta}(\widetilde{A}_1), \cdots, \pi_{k/\eta}(\widetilde{A}_\ell)$ be the projected matrices. We show that the set of covariance matrices corresponding to $\pi_{k/\eta}(\widetilde{A}_1), \cdots, \pi_{k/\eta}(\widetilde{A}_\ell)$ satisfy the approximate spectral histogram property. Using this, we show that the first matrix in this set, $\widetilde{A} := \pi_{k/\eta}(\widetilde{A}_1)$, is a *private projection preserving summary* for A_W with a small additive error. For this, we make use of the private version of one of the characterizations of projection preserving summary due to (Cohen et al., 2015). This characterization is crucial as it defines the multiplicative approximation as well as additive error.

Lemma 2 (Informal version of Lemma 11)**.** *Let k be the desired rank, η be the approximation parameter, and (ϵ, δ) be the privacy parameter. Let Π be the set of all rank-k projection matrices. Then there is an efficient (ϵ, δ)-differentially private algorithm under sliding window model that for a given matrix A_W formed by the current window, outputs a matrix $\widetilde{A}$ such that for any $P \in \Pi$,*

$$\left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F \leq (1+\eta)\left\|A_W(\mathbb{1}_d - P)\right\|_F + O\left(\frac{1}{\alpha\epsilon}\sqrt{kd\log(d)\log^2\left(\frac{W}{\delta}\right)}\right).$$

This lemma allows us to show the first result to solve constrained PCA.

Theorem 2 (Informal version of Theorem 17)**.** *Let A_W be the matrix formed by last W updates and Π be a given set of rank-k projection matrices. Then there is an (ϵ, δ)-differentially private algorithm that outputs a matrix $\widetilde{A}$ at the end of the stream, such that if $\|\widetilde{A}(\mathbb{1}_d - P)\|_F \leq \gamma \cdot \min_{X\in\Pi}\|\widetilde{A}(\mathbb{1}_d - X)\|_F$ for some $\gamma > 0$ and $P \in \Pi$, then*

$$\|A_W(\mathbb{1}_d - P)\|_F \leq (1+\eta)\,\gamma \cdot \min_{X\in\Pi}\|A_W(\mathbb{1}_d - X)\|_F + O\left(\frac{1}{\alpha\epsilon}\sqrt{kd\log(d)\log^2\left(\frac{W}{\delta}\right)}\right).$$

The matrix P in the above result can be computed by running any known non-private algorithm on $\widetilde{A}$. There are existing results for structured projection matrices, such as Asteris et al. (2014); Yuan & Zhang (2013). In particular, if Π is a set of sparse or non-negative projection matrices, then Theorem 17 gives a way to solve these problems privately. Moreover, Theorem 17 also implies a private algorithm for PCA by using any algorithm for PCA that achieves $\gamma = 1$ (Eckart & Young, 1936).

For traditional PCA, Corollary 4.5 in Hardt & Roth (2012) gives a rank-p projection matrix for $p > 2k$ with a large constant multiplicative approximation and $O(\frac{k\sqrt{d}}{\epsilon^2})$ additive error. The underlying reason for this large constant factor is because they use Markov inequality after using the expectation bound of Halko et al. (2011). We avoid this by appealing to the results that use the concentration property of random Gaussian matrices (Kane & Nelson, 2014).

We finally remark that we do not violate the lower bound of Dwork et al. (2014). Their lower bound holds when there is no multiplicative approximation. They show similar upper bound as Theorem 17 when matrices has a singular value gap of $\Omega(\sqrt{d})$. In contrast to their $O(d^2)$ space algorithm, we make use of $O\left(\frac{dk^2}{\eta^3}\log W\right)$ space in the

	Additive Error	Multiplicative	Space Required	Comments
Hardt & Roth (2012)	$\widetilde{O}(k\sqrt{d}/\epsilon^2)$	$O(1)$	$O(d^2)$	rank-$2k$, static data
Dwork et al. (2014)	$\widetilde{O}\left(\epsilon^{-1}k\sqrt{d}\right)$	–	$\widetilde{O}\left(d^2\right)$	Static data
Upadhyay (2018)	$\widetilde{O}\left(\epsilon^{-1}\sqrt{kd}\right)$	$(1+\eta)$	$\widetilde{O}\left(\eta^{-1}dk\right)$	Streaming data
Lower Bound	$\Omega\left(\sqrt{kd}\right)$	$(1+\eta)$	$\Omega\left(\eta^{-1}dk\log W\right)$	Sliding window
This Paper	$\widetilde{O}\left(\epsilon^{-1}\sqrt{kd}\right)$	$(1+\eta)$	$\widetilde{O}\left(\eta^{-3}dk^2\log W\right)$	Sliding window

Table 2. Comparison of $\left(\epsilon, \Theta\left(d^{-\log d}\right)\right)$-Differentially private PCA results (our results are in red).

sliding window setting, which is an improvement whenever $k\log(W) = o(\eta^3 d)$. We also note that Dwork et al. (2014) studied PCA in the *online learning model* (Hazan, 2019), which is incomparable to the sliding window model.

Application II: Multi-response linear regression. Another application of Theorem 1 is solving multi-response linear regression (also known as *generalized linear regression*) in the sliding window model. It is a widely studied generalization of the standard ℓ_2-regression (Woodruff, 2014). Formally, given two matrices $A \in \mathbb{R}^{n\times d}$ and $B \in \mathbb{R}^{n\times p}$ as input, the multi-response linear regression is defined as the minimization problem, $\min_{X\in\mathbb{R}^{d\times p}} \|AX - B\|_F^2$.

Theorem 3 (Informal version of Theorem 18)**.** *Let $A_W \in \mathbb{R}^{W\times d}$ and $B \in \mathbb{R}^{W\times p}$ be the matrix streamed during the window of size W formed as defined in equation (1), ϵ, δ, η be as before. Then there exists an $\tau = \left(d + \frac{14}{\epsilon^2}\log(\frac{4}{\delta})\right)\log^2(W)$ and (ϵ,δ)-differentially private algorithm in the sliding window model that output a matrix $\widetilde{X} \in \mathbb{R}^{d\times p}$ such that*

$$\left\|A_W\widetilde{X} - B_W\right\|_F^2 \leq (1+\eta)\min_{X\in\mathbb{R}^{d\times p}}\|A_WX - B\|_F^2 + O\left(\frac{(\tau+p)^2\log(\tau+p)}{\epsilon}\right).$$

This is the first result for multiple-response regression and matches the bound achieved in Sheffet (2019) when $p = 1$ even though we are in a more restrictive setting.

Application III: Directional variance queries. The directional variance queries has the following form: the analyst gives a unit-length vector $x \in \mathbb{R}^d$ and wish to know the variance of A_W along x. Theorem 1 gives an algorithm to answer directional covariance queries (and cut queries when the matrix is the edge-adjacency matrix of a graph).

Theorem 4 (Informal version of Theorem 15)**.** *Let A_W be the matrix formed by last W updates as defined in equation (1) and ϵ, δ, η be as before. Given a bound q on the number of queries that can be made, there is an efficient (ϵ,δ)-differentially private algorithm that outputs a matrix C such that for any set of q unit vector queries $x_1, \cdots, x_q \in \mathbb{R}^d$, we have for all $i \in [q]$*

$$x_i^\top A_W^\top A_W x_i - \frac{c\log q\log d}{\epsilon} \leq x_i^\top C x_i \leq (1+\eta)x_i^\top A_W^\top A_W x_i + \frac{c\log q\log d}{\epsilon}.$$

Even though we are in a more restrictive setting of sliding window, this matches the bound achieved in Blocki et al. (2012) after we apply the improvement in Sheffet (2019).

4. Concluding remarks

We believe that our approach will find applications beyond what is covered in this paper and will pave way for further research in the intersection of differential privacy and sliding window model. We focus on the model where every data in the current window is considered equally useful to explain the heuristics used in recent deployments. However, one can consider other variants of the sliding window model as far as privacy is concerned. As an example, one can consider a model where the privacy of a data decays as a monotonic function of time lapse. More so, there are more concrete questions to be asked and answered even in the model studied in this paper.

As we mentioned earlier, one can see η-approximate spectral histogram property as a generalization of subspace embedding property. We believe that any improvement in designing a more efficient data structure for maintaining a set of matrices satisfying η-approximate spectral histogram property will have a profound impact on large-scale deployment of privacy-preserving algorithms in the sliding window model. For example, we believe that space requirements can be reduced using randomization. This randomization can be either oblivious or may depend on the current set of positive semidefinite matrices. Since our set of positive semidefinite matrices are generated using a privacy mechanism, any such sampling can be viewed as post-processing and hence privacy preserving. Hence, our main conjectures are concerning the space required by any privacy-preserving algorithm. We elaborate them next.

The lower bound of $\Omega\left(d^2\right)$ space for spectral approximation is required even in the static setting. We conjecture that

there should be $\frac{1}{\eta}\log W$ factor due to the sliding window requirement. This is because, if the spectrum of a matrix is polynomially bounded, then one can construct a sequence of updates that requires at least $\frac{1}{\eta}\log W$ matrices such that successive matrices are $(1-\eta)$ apart in terms of their spectrum. For an upper bound, we believe randomization can help reduce a factor of d. This is achieved in the non-private setting using online row sampling. It was shown by Upadhyay (2018) that one can design private algorithms with space-bound comparable to a non-private algorithm in the streaming model of computation. The situation in the sliding window model is more complicated, but we believe it is possible to achieve a matching upper bound. In view of this, we conjecture the following.

Conjecture 1. *The space required for differentially private spectral approximation is* $\Theta\left(\frac{d^2}{\eta}\log W\right)$.

We believe that the bound on the additive error is optimal. A positive resolution to this conjecture would imply that the price of privacy is only in terms of the additive error.

Our second conjecture is for principal component analysis. We believe that our space-bound for principal component analysis is tight up to a factor of $\frac{k}{\eta}$. A lower bound of $\Omega(dk)$ is trivial as one requires $O(dk)$ space just to store the orthonormal matrix corresponding to the rank-k projection matrix. As before, a factor of $\frac{1}{\eta}\log W$ would be incurred due to the sliding window model. The factor of $\frac{1}{\eta}$ comes from the fact that to extract the top-k subspace, we need $\frac{k}{\eta}$ dimensional subspace.

Conjecture 2. *The space required for differentially private PCA is* $\Omega\left(\frac{dk}{\eta^2}\log W\right)$.

We believe that proving such a lower bound would require new techniques. This is because, in PCA, we only have access to an orthonormal projection matrix, while in the case of low-rank approximation, we have far more information to solve the underlying communication complexity problem.

Our work identifies another application of the Johnson-Lindenstrauss and Wishart mechanisms. Before our results, it was not even clear whether the JL mechanism can be used to compute PCA (see Section V in Blocki et al. (2012))! They consider their output matrix $\widetilde{C}$ as a "test" matrix to test if the input matrix has high directional variance along some direction $x \in \mathbb{R}^d$. However, they do not give any guarantee as to how the spectrum of C relates to that of the input covariance matrix.

5. Acknowledgement

JU's research was supported, in part, by NSF BIGDATA awards IIS-1838139 and IIS 1546482. JU would like to acknowledge Petros Drineas for useful discussion on the technique of Batson et al. (2012).

References

Achlioptas, D. and McSherry, F. On spectral learning of mixtures of distributions. In *Proceedings of the 18th Annual Conference on Learning Theory*, pp. 458–469. Springer, 2005.

Amin, K., Dick, T., Kulesza, A., Munoz, A., and Vassilvitskii, S. Differentially private covariance estimation. In *Proceedings of the 32nd Advances in Neural Information Processing Systems*, pp. 14190–14199, 2019.

Arora, R. and Upadhyay, J. On differentially private graph sparsification and applications. In *Proceedings of the 32nd Advances in Neural Information Processing Systems*, pp. 13378–13389, 2019.

Arora, R., Braverman, V., and Upadhyay, J. Differentially private robust low-rank approximation. In *Advances in Neural Information Processing Systems*, pp. 4137–4145, 2018.

Asteris, M., Papailiopoulos, D., and Dimakis, A. Nonnegative sparse PCA with provable guarantees. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1728–1736, 2014.

Bakshi, A., Chepurko, N., and Jayaram, R. Testing positive semi-definiteness via random submatrices. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science*, pp. 1191–1202, 2020.

Bar-Yossef, Z. *The complexity of massive data set computations*. PhD thesis, University of California, Berkeley, 2002.

Batson, J., Spielman, D. A., and Srivastava, N. Twice-ramanujan sparsifiers. *SIAM Journal on Computing*, 41 (6):1704–1721, 2012.

Bentley, J. L. and Saxe, J. B. Decomposable searching problems i. static-to-dynamic transformation. *Journal of Algorithms*, 1(4):301–358, 1980.

Blocki, J., Blum, A., Datta, A., and Sheffet, O. The Johnson-Lindenstrauss transform itself preserves differential privacy. In *Proceeding of the 53rd Annual IEEE Sympsium on Foundations of Computer Science*, pp. 410–419, 2012.

Blum, A., Dwork, C., McSherry, F., and Nissim, K. Practical privacy: the SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 128–138, 2005.

Bolot, J., Fawaz, N., Muthukrishnan, S., Nikolov, A., and Taft, N. Private decayed predicate sums on streams.

In *Proceedings of the 16th International Conference on Database Theory*, pp. 284–295, 2013.

Braverman, V., Drineas, P., Musco, C., Musco, C., Upadhyay, J., Woodruff, D. P., and Zhou, S. Near optimal linear algebra in the online and sliding window models. In *Proceeding of the 61st Annual Symposium on Foundations of Computer Science*, pp. 517–528, 2020.

Campos, P. G., Díez, F., and Cantador, I. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1-2):67–119, 2014.

Chan, T. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Transactions on Information and System Security*, 14(3):26:1–26:24, 2011.

Chan, T.-H. H., Li, M., Shi, E., and Xu, W. Differentially private continual monitoring of heavy hitters from distributed streams. In *Proceedings of the 12th International Privacy Enhancing Technologies Symposium*, pp. 140–159, 2012.

Clarkson, K. L. and Woodruff, D. P. Low-rank approximation and regression in input sparsity time. *Journal of the ACM*, 63(6):54, 2017.

Cohen, M. B., Elder, S., Musco, C., Musco, C., and Persu, M. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM symposium on Theory of computing*, pp. 163–172, 2015.

Drineas, P., Kerenidis, I., and Raghavan, P. Competitive recommendation systems. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, pp. 82–90, 2002.

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

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pp. 715–724, 2010.

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, 2014.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Erlingsson, Ú., Pihur, V., and Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1054–1067, 2014.

Friedland, S. and Torokhti, A. Generalized rank-constrained matrix approximations. *SIAM Journal on Matrix Analysis and Applications*, 29(2):656–659, 2007.

Halko, N., Martinsson, P.-G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

Hardt, M. and Price, E. The noisy power method: A meta algorithm with applications. In *Proceedings of the 27th Advances in Neural Information Processing Systems*, pp. 2861–2869, 2014.

Hardt, M. and Roth, A. Beating randomized response on incoherent matrices. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pp. 1255–1268, 2012.

Hazan, E. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Huang, Z., Qiu, Y., Yi, K., and Cormode, G. Frequency estimation under multiparty differential privacy: One-shot and streaming. *arXiv preprint arXiv:2104.01808*, 2021.

Kane, D. M. and Nelson, J. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):4, 2014.

Kapralov, M. and Talwar, K. On differentially private low rank approximation. In *Proceedings of the 44th Annual ACM-SIAM Symposium on Discrete algorithms*, pp. 1395–1414, 2013.

McSherry, F. and Mironov, I. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 627–636, 2009.

Miltersen, P. B., Nisan, N., Safra, S., and Wigderson, A. On data structures and asymmetric communication complexity. In *STOC*, pp. 103–111. ACM, 1995.

Moore, J. L., Chen, S., Turnbull, D., and Joachims, T. Taste over time: The temporal dynamics of user preferences. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pp. 401–406, 2013.

Muthukrishnan, S. Data streams: Algorithms and applications. *Foundations and Trends® in Theoretical Computer Science*, 1(2), 2005.

Narayanan, A. and Shmatikov, V. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

Quadrana, M., Cremonesi, P., and Jannach, D. Sequence-aware recommender systems. *ACM Computing Surveys*, 51(4):66, 2018.

Sarlós, T. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 143–152, 2006.

Sheffet, O. Old techniques in differentially private linear regression. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, pp. 789–827, 2019.

Singhal, V. and Steinke, T. Privately learning subspaces. *arXiv preprint arXiv:2106.00001*, 2021.

Smith, A., Song, S., and Thakurta, A. The Flajolet-Martin sketch itself preserves differential privacy: Private counting with minimal space. In *Proceedings of the 33rd Advances in Neural Information Processing Systems*, 2020.

Thakurta, A. G., Vyrros, A. H., Vaishampayan, U. S., Kapoor, G., Freudiger, J., Sridhar, V. R., and Davidson, D. Learning new words, March 14 2017. US Patent 9,594,741.

Tsay, R. *Analysis of Financial Time Series*. Wiley Series in Probability and Statistics. Wiley-Interscience, 2005.

Upadhyay, J. The price of privacy for low-rank factorization. In *Proceedings of the 31st Advances in Neural Information Processing Systems*, pp. 4180–4191, 2018.

Upadhyay, J. Sublinear space private algorithms under the sliding window model. In *Proceedings of the 36th International Conference of Machine Learning*, pp. 6363–6372, 2019.

Upadhyay, J., Upadhyay, S., and Arora, R. Differentially private analysis on graph streams. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pp. 1171–1179, 2021.

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1-2):1–157, 2014.

Yuan, X.-T. and Zhang, T. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14:899–925, 2013.

Zass, R. and Shashua, A. Nonnegative sparse PCA. In *Proceedings of the 19th Advances in Neural Information Processing Systems*, pp. 1561–1568, 2006.