Robust Streaming PCA

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider streaming principal component analysis (PCA) when the stochastic data-generating model is subject to perturbations. While existing models assume a fixed covariance, we adopt a robust perspective where the covariance matrix belongs to a temporal uncertainty set. Under this setting, we provide fundamental limits on any algorithm recovering principal components. We analyze the convergence of the noisy power method and Oja's algorithm, both analyzed for the stationary data generating model, and argue that the noisy power method is rate-optimal in our setting. Finally, we demonstrate the validity of our analysis through numerical experiments.

1 Introduction

Principal component analysis (PCA) is one of the most extensively studied methods for obtaining the low-dimensional representation of observed data [21]. However, classical algorithms for PCA store all the observations and use cubic-time complexity, thereby imposing prohibitively large computation-time and space requirements.

Recently, several works on PCA have focused on the design and analysis of single-pass (streaming) algorithms with near-optimal memory and storage complexity [40, 43, 24, 28]. These algorithms assume that all the observations belong to the same low-dimensional space. However, this situation is unlikely when the unknown/unexplored alterations corrupt a system's observations. For instance, it is well known that the typical data attacks on power grids can significantly change the estimated covariance matrix of the data observed from sensors [8, 22, 9]. Similarly, PCA can be used to explain stock returns in terms of the macroeconomic factors [26], product pricing taking into account cross product elasticity and demands [45], etc. In all these scenarios the underlying data generating model changes at every time instant and the decisions are based on identifying the changed model.

Recent work considers perturbations of the data lying in a *fixed* low-dimensional space [49]. They determine the worst-case position of the adversarial data-point to incur the maximum error in the subspace estimated through PCA and measure the distance between the two subspaces using the notion of the principal angle between them. Another line of work considers PCA through the lens of stochastic optimization [51, 52, 42, 4]. Our work differs from these approaches since we assume that the data generating model changes at every time instant. Further, we propose near-optimal algorithms for recovering principal components under this framework.

We assume a system relies on the time-series of p-dimensional vectors sampled from the time-variant model. The available observations are the vectors $\{\mathbf{x}_t\}_{t=1}^{T}$. We assume that $T \gg p$; justifying the use of streaming (single-pass) algorithms. The noisy observation $\mathbf{x}_t \in \mathbb{R}^p$ is a vector lying in the column space of an unobserved full-rank matrix $\mathbf{A}_t \in \mathbb{R}^{p\times k}$. Precisely, from the standard spiked covariance model [35], we consider the time-dependent environment:

$$\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}_{p\times 1}\,,\, \mathbf{A}_t\mathbf{A}_t^\top + \sigma^2\mathbf{I}_{p\times p}), \tag{1}$$

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

where $\mathbf{A}_t$ can vary with time. The parameter $k \ll p$ is the desired number of principal components. Assuming $\mathbf{A}_t$ belongs to a temporal uncertainty set (defined in (2)), our goal is to recover the top-k principal components of the terminal subspace $\mathbf{A}_T$. In financial applications, our model captures the market evolution, in terms of the changing $\mathbf{A}_T$ with the ultimate goal of explaining the market conditions for an appropriately chosen T.

Previous works on the stationary environment assume $\mathbf{A}_t = \mathbf{A}$ for all t and have focused on computing a basis for the column space of the matrix $\mathbf{A}$ using streaming algorithms [43, 28, 38, 32, 20, 36]. The accurate reconstruction of the principal components for the standard streaming PCA problem depends on the magnitude of observation noise σ, the dimension of observations p, the number of principal components k of the matrix $\mathbf{A}$, and the spectral gap δ between k^{th} and $k{+}1^{\text{th}}$ singular value of $\mathbf{A}$. In a marked departure from previous work, we study the case when the column space of $\mathbf{A}$ varies across time, possibly perturbed by Γ. This paper explores these avenues and proposes a tractable analysis framework for the streaming PCA problem, robust to perturbations in the stochastic data-generating model. Our contributions can be summarized as:

(a) (Lower Bound; Section 4) Our first contribution is establishing a fundamental lower bound for estimating the principal components when the covariance matrix belongs to a temporal uncertainty set. In Theorem 1, we derive the minimax bounds of the expected error for recovering the top singular vectors for any algorithm in the class Φ. A key component of this analysis involves carefully constructed instances to elicit the worst-case behavior of any algorithm in Φ. We show that for $T = \mathcal{O}(\Gamma^{-2/3})$, the minimax estimation error decreases as $\mathcal{O}(p^{1/2}/\delta\sqrt{T})$. On the other hand, for $T = \Omega(\Gamma^{-2/3})$, the estimation error stagnates to $\mathcal{O}(p^{1/3}\Gamma^{1/3}/\delta)$ and does not decrease upon collecting more observations[1]. Since observations can belong to potentially different spaces, the estimation of principal components associated with $\mathbf{A}_T$ should be determined by a subset of samples. This is in sharp contrast to the standard spiked covariance model (where data is generated from the fixed model), observations from the far past become less important when the covariance matrix is time-variant, and it becomes imperative to find the block size B that can be used to recover the principal components.

(b) (Algorithm Analysis; Section 5) We then analyze two algorithms to recover the principal components extensively used in the standard streaming PCA setting; the noisy power method [28] and Oja's algorithm [48]. These algorithms represent two very different design principles for computing principal components from data; processing data in blocks vis-à-vis single observations (However, note that both algorithms perform computations in a streaming fashion, i.e., have $\mathcal{O}(kp)$ storage requirements). We first determine the optimal choice for critical parameters: the block size in the noisy power method on Lemma 1 and the learning rate in Oja's algorithm on Lemma 2. Next, we leverage these results in order to obtain an upper bound on the convergence error for these algorithms in Theorem 2 and Theorem 3respectively. From these results, we have that: (i) the block size B for the noisy power method and inverse of the learning rate, $1/\zeta$ for Oja's algorithm play a similar role even in the non-stationary environment; (ii) In either case, the optimal parameters scale with $\sim\Gamma^{-2/3}$, where Γ is perturbation budget which will be defined later, and (iii) the derived upper bound on the estimation error for noisy power method matches the minimax error in terms of p, δ, and Γ and becomes rate-optimal if it satisfies mild condition.

Notation We fix notation throughout the main body of the paper. Matrices are denoted by bold uppercase letters ($\mathbf{M}$) and vectors are denoted by bold lowercase letters ($\mathbf{x}$). Let $\mathbf{M} \in \mathbb{R}^{p\times k}$. Then $\mathbf{M}_i$ denote the ith column of $\mathbf{M}$, $\mathbf{M}_{i:j}$ denote the submatrix consists with $i^{\text{th}} \sim j^{\text{th}}$ column of $\mathbf{M}$ and $\mathbf{M}_{i,j}$ be the (i,j)-element of $\mathbf{M}$. $\|\cdot\|$ denote matrix 2-norm or equivalently operator norm for matrices and standard 2-norm for vectors. We use $\mathrm{St}_k(\mathbb{R}^p)$ for the class of orthogonal matrices in $\mathbb{R}^{p\times k}$. $b(\mathbf{M})$ is the orthogonal matrix where the columns form basis for $\mathrm{ran}(\mathbf{M})$. $\lambda_i(\mathbf{M})$ represents the ith largest eigenvalue of the matrix $\mathbf{M}$ and and $s_i(\mathbf{M})$ be the ith largest singular value. The singular value decomposition of $\mathbf{M}$ is defined as $\mathrm{SVD}(\mathbf{M})$=$\mathbf{U}\mathbf{D}\mathbf{V}^\top$, where $\mathbf{U}\in\mathrm{St}_p(\mathbb{R}^p)$, $\mathbf{V}\in\mathrm{St}_k(\mathbb{R}^k)$, and $\mathbf{D} \in \mathbb{R}^{p\times k}$ is a diagonal matrix whose i^{th} diagonal element equals $s_i(\mathbf{M})$. Therefore, we assume without loss of generality that the singular values and respective singular vectors are ordered from largest to smallest. We denote by $\mathbf{M}_\perp$ the orthogonal projection onto the orthogonal complement of

[1] For precise explanation for δ and Γ, please refer to the Section 3.

the range of $\mathbf{M}$. Therefore, if $\mathrm{rk}(\mathbf{M})$=k and $\mathrm{SVD}(\mathbf{M})$= $\mathbf{UDV}^\top$, $\mathbf{M}_\perp$ given by $\mathbf{M}_\perp$=$\mathbf{I} - \mathbf{U}_{1:k}\mathbf{U}_{1:k}^\top$. Moreover, when $\mathrm{rk}(\mathbf{M})$=$\mathrm{rk}(\tilde{\mathbf{M}})$ and $\mathrm{SVD}(\tilde{\mathbf{M}})$= $\tilde{\mathbf{U}}\tilde{\mathbf{D}}\tilde{\mathbf{V}}^\top$, the distance between $\mathrm{ran}(\mathbf{M})$ and $\mathrm{ran}(\tilde{\mathbf{M}})$ is defined by:

$$d(\mathrm{ran}(\mathbf{M}), \mathrm{ran}(\tilde{\mathbf{M}})) = \| \, b(\mathbf{M})b(\mathbf{M})^\top - b(\tilde{\mathbf{M}})b(\tilde{\mathbf{M}})^\top \, \| = \| \, \mathbf{U}_{1:k}\mathbf{U}_{1:k}^\top - \tilde{\mathbf{U}}_{1:k}\tilde{\mathbf{U}}_{1:k}^\top \, \| \, .$$

We denote $d(\mathrm{ran}(\mathbf{M}), \mathrm{ran}(\tilde{\mathbf{M}}))$ by $d(\mathbf{M}, \tilde{\mathbf{M}})$ whenever clear from the context. For $\mathcal{A}$=$(\mathbf{A}_1, \ldots, \mathbf{A}_T)$, we denote the expectation of $f(\mathbf{x}_1, \ldots, \mathbf{x}_T)$ over $\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{A}_t\mathbf{A}_t^\top + \sigma^2\mathbf{I})$ as $\mathbb{E}_{\mathcal{X}\sim\mathcal{A}}[f]$. We denote $\tilde{\mathcal{O}}$ as the big-$\mathcal{O}$ notation with ignore the multiplicative dependency of $\log(p) + 2\log(T)$ and $1 + \Theta(\log(T/\Delta)/k)$.

2 Related Work

Principal component analysis has been extensively studied across operations research, computer science, and other communities. We highlight how our work differs from existing literature. **Robust PCA** Robust PCA deals with the problem of retrieving the principal components robust to the presence of outliers in the data. The cornerstone work in this direction is the Principal Component Pursuit framework wherein they assume that the matrix of observations $\mathbf{M}$ can be decomposed in terms of a low-rank matrix $\mathbf{L}$ and a sparse matrix (with entries of arbitrarily large magnitude) $\mathbf{S}$ [14].

Several works [23, 25, 16, 46] consider the robust PCA problem and propose algorithms in the offline, batch, and online setting. Our work differs from this line of literature in two aspects: assumptions about the data generating model and convex optimization techniques. First, our data generation model is unrelated to those considered in the robust PCA literature. Further, rich theories from convex optimization can be used in the PCA framework to design efficient algorithms, but our problem is not amenable to such techniques.

Streaming PCA Streaming algorithms for PCA have been proposed among other works on PCA [43, 34, 28, 59, 2]. Algorithms analyzed in this work, such as the noisy power method [28] and Oja's algorithm [48], are iterative methods for estimating the principal components. These iterative schemes are instances of stochastic approximation-based solutions for the optimization version of the PCA problem [4]. The stochastic approximation is a root-finding framework extensively used for stochastic optimization [37, 10]. Oja's algorithm, originally proposed by [48], was the first such scheme.

This framework is also used to analyze gradient-type and proximal-type incremental methods akin to algorithms for convex optimization. Stochastic gradient descent-based algorithms for the streaming PCA problem, where a single observation is used at every point in time to update the principal components' estimate, have been extensively studied. Along this line, [29] propose GRASTA, an incremental online gradient method for learning over different subspaces. Similarly, [5] proposes GROUSE, based on the idea of gradient updates over the Grassmannian manifold. In these and other related works [53, 15, 60, 30, 30, 1, 58] consider Oja's algorithm for the standard streaming PCA problem. However, all these works are based on a completely different modeling assumption than this paper and do not provide theoretical guarantees for our setting.

Streaming and robust PCA algorithms are used in presence of outliers or data with a lot of missing entries [17, 54]. From this literature, closest to our work is the work on robust subspace tracking [2, 59, 29]. However, the robustness considered their is against erasures or sparse outliers. While these algorithms provide theoretical guarantees in that setting, those guarantees cannot be extended to our model. Considerations of erasures and outliers under the model proposed in this paper is beyond the scope of this work and remains an interesting future direction.

3 Mathematical Framework

Observations from the standard spiked covariance model belong to a fixed k-dimensional column space $\mathbf{A} \in \mathbb{R}^p$. Previous work has focused on reconstructing this space from the observed time series. Under our framework, we assume that the sequence of observations $(\mathbf{x}_t)_{t=1}^T$ follows the time-dependent spiked covariance model (1). We consider the problem of computing top-k singular vectors of $\mathbf{A}_T$. We formulate this model by addressing a minimax optimization as a robust optimization problem.

When the adversary is allowed to select a completely arbitrary sequence of matrices $\mathbf{A}_1, \ldots, \mathbf{A}_T$, it is impossible to accurately recover the column space of $\mathbf{A}_T$ since only the single vector $\mathbf{x}_T$ provides information about this space. Instead, we define temporal uncertainty sets to restrict the power of the adversary.

Definition 1. Let $\Gamma, \delta \geq 0$. We only allow the sequence of matrices $\mathbf{A}_1, \ldots, \mathbf{A}_T$ that lie in an temporal uncertainty set $\mathrm{Tu}(\delta, \Gamma)$ defined as:

$$\mathrm{Tu}(\delta, \Gamma) := \left\{ (\mathbf{A}_t)_{t=1}^T \ : \ s_k(\mathbf{A}_t \mathbf{A}_t^\top) \geq \delta \ , \ \max_{2 \leq t \leq T} \|\mathbf{A}_t \mathbf{A}_t^\top - \mathbf{A}_{t-1} \mathbf{A}_{t-1}^\top\| \leq \Gamma \right\} . \tag{2}$$

That is, we constrain the difference between any two consecutive covariance matrices of the underlying (noise-free) process. The assumption $s_k(\mathbf{A}_t \mathbf{A}_t^\top) \geq \delta$ is crucial for establishing bounds of estimation error. It is also justified in any setting where the underlying phenomenology guarantees covariance of rank k; the stated lower bound excludes pathological cases of near-rank $k-1$ or smaller.

Having described the constraints on the perturbation power, we now present the family of algorithms and the performance metric of interest. In particular, we consider algorithms that are (a) streaming, i.e., make a single pass through the time series in chronological order; only the previous samples are stored at any point in the past. These characteristics model the behavior of an algorithm that receives data in real-time and only stores samples in limited memory. Among those streaming algorithms, we consider the algorithms whose (b) output is a set of k orthonormal vectors in $\mathbb{R}^p$. The first requirement ensures that the storage and computational memory requirements are independent of the number of samples T, and hence, the assumption $T \gg p$ is not prohibitive. Let us denote by Φ the family of algorithms just described. Let $\mathcal{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_T) \in \mathrm{Tu}(\delta, \Gamma)$ be given. We formalize this definition as follows:

Definition 2. $\Sigma(\mathcal{A})$ is the time-dependent spiked covariance model represented by the $\mathcal{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_T)$. Furthermore, $\mathcal{X} := (\mathbf{x}_1, \ldots, \mathbf{x}_T) \sim \Sigma(\mathcal{A})$ is a sequence of observation vectors drawn from $\Sigma(\mathcal{A})$.

The output of algorithm ϕ for a given choice of the observations $\mathbf{x_t}$ sampled using model (2) is a set of k orthonormal vectors which we denote by $\phi_{\mathcal{X}}$; abusing notation we will also view $\phi_{\mathcal{X}}$ as a matrix in $\mathbb{R}^{p \times k}$ or the subspace generated by $\phi_{\mathcal{X}}$.

Definition 3. Let $\phi \in \Phi$, $\mathcal{A} \in \mathrm{Tu}(\delta, \Gamma)$ and $\mathcal{X} \sim \Sigma(\mathcal{A})$.

(a) Given $\mathcal{A}$ and $\mathcal{X}$, the estimation error of ϕ given $\mathcal{X}$ is the distance between the space spanned by $\phi_{\mathcal{X}}$ and the column space of $\mathbf{A}_T$ (=$d(\mathrm{ran}(\mathbf{A}_T), \phi_{\mathcal{X}})$). The metric can be easily extended to the cumulative error. Please see the Appendix A for the discussion.

(b) $\mathcal{R}^\phi$ is the maximum expected estimation error of ϕ under $\mathcal{A}$ over all $\mathcal{A} \in \mathrm{Tu}(\delta, \Gamma)$. Precisely,

$$\mathcal{R}^\phi \ := \sup_{\mathcal{A} \in \mathrm{Tu}(\delta, \Gamma)} \mathbb{E}_{\mathcal{X} \sim \mathcal{A}} \left[d\big(\mathrm{ran}(\mathbf{A}_T), \phi_{\mathcal{X}}\big) \right] .$$

(c) $\mathcal{R}^*$ is the minimax estimation error defined as the minimum of the largest expected estimation error incurred by $\phi \in \Phi$:

$$\mathcal{R}^* := \inf_\phi \mathcal{R}^\phi = \inf_\phi \sup_{\mathcal{A} \in \mathrm{Tu}(\delta, \Gamma)} \mathbb{E}_{\mathcal{X} \sim \mathcal{A}} \left[d\big(\mathrm{ran}(\mathbf{A}_T), \phi_{\mathcal{X}}\big) \right] .$$

We note that minimax estimation problems have been widely studied [56, 13, 57].

(d) An algorithm $\phi \in \Phi$ is **rate-optimal** if $\mathcal{R}^\phi \leq C \cdot \mathcal{R}^*$, where constant $C > 0$ is independent of the problem parameters Γ, T, δ, p, k. In this paper we establish the minimax estimation error $\mathcal{R}^*$ and propose rate-optimal sublinear-time, single-pass algorithms for robust streaming PCA.

4 Minimax Lower Bound

When $\mathbf{A}_t$ belongs to the temporal uncertainty set $\mathrm{Tu}(\delta, \Gamma)$ (Definition 1) an algorithm designed to recover the principal components of $\mathbf{A}_T$ from the observations cannot guarantee zero estimation error. Our first goal is to obtain a lower bound on the estimation error in terms of the problem parameters

δ, Γ, p and the number of observations T. We now establish a lower bound on the expected error in recovering the top singular vector of $\mathbf{A}_T$ from observations $\mathcal{X} = (\mathbf{x}_t)_{t=1}^T$.

In order to establish the lower bound, we leverage the fundamental limit of hypothesis tests [56]. The crux of the proof lies in constructing the set of worst-case hypotheses and establishing a lower bound on the probability of error in distinguishing between these hypotheses using observed data. The complete proof is provided in Appendix C.

Theorem 1 (Lower Bound)**.** *Assume $0 \leq \Gamma < \delta$ and $T \geq 1$. For any algorithm $\phi \in \Phi$, there exists a sequence $\mathcal{A} \in \mathrm{Tu}(\delta, \Gamma)$ and constant $C > 0$ which is independent with Γ, δ, T, p, k such that:*

$$\mathbb{E}_{\mathcal{X}\sim\mathcal{A}}\big[d\big(\mathrm{ran}(\mathbf{A}_T), \phi_{\mathcal{X}}\big)\big] \geq \min\left\{1, C\Big(\left(\tfrac{\Gamma}{\delta}\right)^{1/3}\left(\tfrac{p\sigma^2(\sigma^2+\delta)}{\delta^2}\right)^{1/3} + \tfrac{1}{\sqrt{T}}\left(\tfrac{p\sigma^2(\sigma^2+\delta)}{\delta^2}\right)^{1/2}\Big)\right\}. \quad (3)$$

For the standard streaming PCA problem (Theorem 1 with the case of $\Gamma = 0$), the fundamental limit is $\Theta((\sigma/\delta)(p(\sigma^2+\delta)/T)^{1/2})$ which has the expected $\Theta(1/\sqrt{T})$ dependence [13, 57]. On the other hand, in presence of perturbations ($\Gamma > 0$), the lower bound exhibits a phase transition phenomenon, with the first term representing the effect of model ambiguity. To this end, define the critical time T_c as

$$T_c := \left(\frac{\Gamma}{\delta}\right)^{-2/3}\left(\frac{p\sigma^2(\sigma^2+\delta)}{\delta^2}\right)^{1/3}. \quad (4)$$

For $T = \mathcal{O}(T_c)$, the lower bound decreases with the rate of $1/\sqrt{T}$. However, when $T = \Omega(T_c)$, the first term dominates the second term and $\mathcal{R}^*$ becomes independent to the number of observations T. In this regime, the error stagnates to $\mathcal{O}((\Gamma/\delta)^{1/3}\left(p\sigma^2(\sigma^2+\delta)/\delta^2\right)^{1/3})$. This validates our intuition that in a dynamic environment past information quickly becoming stale.

In Theorem 2 and 3, we will prove that the noisy power method and Oja's algorithm attain a near-optimal bound on the convergence guarantee. Theorem 2 guarantees that if $s_1(\mathbf{A}_t\mathbf{A}_t^\top) = \delta$, the upper bound for estimation error on the noisy power method is of the following order:

$$\tilde{\mathcal{O}}\left(\left(\frac{\Gamma}{\delta}\right)^{1/3}\Big(\frac{(p\sigma^2+k\delta)(\sigma^2+\delta)}{\delta^2}\Big)^{1/3}\right). \quad (5)$$

That is, if $p\sigma^2$ dominates $k\delta$, the noisy power method becomes rate-optimal under the controlled uncertainty with $\mathrm{Tu}(\delta, \Gamma)$. This regime is the case of practical noisy situations, with $\sigma^2 \not\ll \delta$. Observe also that we have $p > k$ from the setting.

5 Convergence Analysis

In this section, we analyze two algorithms for the robust streaming PCA problem. A generic template for algorithms $\phi \in \Phi$ of interest to us is as follows: (i) ϕ is initialized with a random matrix with orthonormal columns $\hat{\mathbf{U}} \in \mathbb{R}^{p\times k}$; (ii) A running estimate of the principal components is maintained as the columns of $\hat{\mathbf{U}}$; (iii) Observations are projected onto the column space of $\hat{\mathbf{U}}$ to update this estimate.

We consider two algorithms: robust version of the noisy power method (Algorithm 1) and Oja's algorithm (Algorithm 2). The key difference between the noisy power method and Oja's algorithm is in the data used to estimate the principal components. In the noisy power method, the estimates are updated after a batch of observations, whereas in Oja's algorithm, the estimates are updated after scaling every observation with the learning rate. Therefore, the parameters determining the performance of these algorithms are the batch size B for the robust power method and the learning rate ζ for Oja's algorithm.

The analysis of these algorithms cannot be readily established with existing techniques when the covariance matrix belongs to a temporal uncertainty set since they rely on showing that the estimates improve every iteration. Further, the application of many concentration results requires random matrices to be bounded which is not the case when the observations are sampled from (1). Therefore, in order to simplify the analysis on both algorithms, we introduce Assumption 1, adapted from [34].

Assumption 1. Let $(\mathbf{A}_t)_{t=1}^T \in \mathrm{Tu}(\delta, \Gamma)$. For $\tilde{\delta} \geq \delta$ and $\mathcal{M}, \mathcal{V} > 0$, we consider the observations $\mathbf{x}_t$ for $t \in [T]$ satisfy the following:

(a) $\mathbb{E}[\mathbf{x}_t\mathbf{x}_t^\top] = \mathbf{A}_t\mathbf{A}_t^\top + \sigma^2\mathbf{I}_{p\times p}$ while $\|\mathbf{A}_t\mathbf{A}_t^\top\| \le \tilde{\delta}$,

(b) $\|\mathbf{x}_t\mathbf{x}_t^\top - (\mathbf{A}_t\mathbf{A}_t^\top + \sigma^2\mathbf{I}_{p\times p})\| \le \mathcal{M}$ almost surely, and

(c) $\left\|\mathbb{E}\left[(\mathbf{x}_t\mathbf{x}_t^\top - (\mathbf{A}_t\mathbf{A}_t^\top + \sigma^2\mathbf{I}_{p\times p}))^2\right]\right\| \le \mathcal{V}$.

When the observations follow equation (1), we condition our analysis on event $\mathfrak{E}_\Delta$.

Definition 4. For $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}_{k\times 1}\,,\,\mathbf{I}_{k\times k})$ and $\mathbf{w}_t \in \mathcal{N}(\mathbf{0}_{p\times 1}\,,\,\mathbf{I}_{p\times p})$, the event $\mathfrak{E}_\Delta$ is defined as:

$$\mathfrak{E}_\Delta \;:\; \begin{cases} \|\mathbf{z}_t\|^2 \le k + 2\sqrt{k\log(2T/\Delta)} + 2\log(2T/\Delta)\,, \text{ and} \\ \|\mathbf{w}_t\|^2 \le p + 2\sqrt{p\log(2T/\Delta)} + 2\log(2T/\Delta) \end{cases} \quad \forall t \in [T]\,. \tag{6}$$

The observations from model (1) satisfy Assumption 1 with $\mathcal{M} = 2(k\tilde{\delta}+p\sigma^2)(1+\Theta(\log(T/\Delta)/k))$, $\mathcal{V} = 2\mathcal{M}(\tilde{\delta}+\sigma^2)$ with probability $\mathbb{P}[\mathfrak{E}_\Delta] \ge 1-\Delta$. Although after conditioning $\mathbb{E}[x_t x_t^\top|\mathfrak{E}_\Delta] \ne \mathbf{A}_t\mathbf{A}_t^\top + \sigma^2\mathbf{I}_{p\times p}$, we can use all the results in Section 5.1 and 5.2 with a sufficiently small Δ. Please refer to Appendix H for the details.

5.1 Noisy Power Method

Algorithm 1 Noisy Power Method with block size B [28]

Input: $\{\mathbf{x}_1\}_{t=1}^T$, Block Size: B, Dimensions: p, k & **Output:** $\hat{\mathbf{U}}(L)$
$\hat{\mathbf{U}}(0) \leftarrow p \times k$ random matrix
for $\ell = 1 : L = T/B$ **do**
  $\mathbf{Y}(\ell) \leftarrow \mathbf{0} \in \mathbb{R}^{p\times k}$
  for $t = (\ell-1)B+1 : \ell B$ **do**
    $\mathbf{Y}(\ell) \leftarrow \mathbf{Y}(\ell) + \frac{1}{B}\mathbf{x}_t\mathbf{x}_t^\top\hat{\mathbf{U}}(\ell-1)$
  end for
  $\hat{\mathbf{U}}(\ell) \leftarrow$ Gram-Schmidt$(\mathbf{Y}(\ell))$
end for

The noisy power method is an iterative algorithm for computing the top-k principal components of given matrix. Starting from the random matrix $\hat{\mathbf{U}}(0)$ in $\mathbb{R}^{p\times k}$, the algorithm runs for L iterations, each processing B samples. By repeating this procedure, we expect the algorithm to reconstruct the covariance matrix if the observations are derived from the fixed distribution.

When observations are drawn from the model 1 under Assumption 1, the observations belong to different underlying subspace. Unlike the standard case, for any algorithm ϕ, the presence of perturbation prevents the convergence of the columns of $\hat{\mathbf{U}}(\ell)$ to the singular vectors of the final covariance matrix $\mathbf{U}(\ell)$. The main difficulty here is that the columns of $\hat{\mathbf{U}}(\ell)$ do not converge towards a fixed set of vectors but keep tracking the time-varying principal components. Recall that our ultimate objective is to recover the principal components associated with the *terminal* observation. Hence, we decompose the covariance matrix in terms of the last observation and the remaining samples. For ℓ-th block, we denote the covariance matrix for ℓB-th observation as $\mathbf{M}(\ell)$ and have:

$$\frac{1}{B}\sum_{t=(\ell-1)B+1}^{\ell B} \mathbf{x}_t\mathbf{x}_t^\top = \mathbb{E}[\mathbf{x}_{\ell B}\mathbf{x}_{\ell B}^\top] + \mathcal{E}(\ell) = \underbrace{\mathbf{A}_{\ell B}\mathbf{A}_{\ell B}^\top + \sigma^2\mathbf{I}}_{\mathbf{M}(\ell)} + \mathcal{E}(\ell)\,. \tag{7}$$

Due to presence of perturbations in the robust model, $\mathcal{E}(\ell)$ is a non-zero mean random variable. We first decompose $\mathcal{E}(\ell)$ in terms of the contribution due to inherent noise and the perturbations allowed by the robust model. In Lemma 1, we decompose the error $\|\mathcal{E}(\ell)\|$ with respect to the block size B and the allowed perturbations in the robust model. The complete proof is provided in Appendix D.

Lemma 1 (Spectral norm of noise)**.** *Assume that the observations $(\mathbf{x}_t)_{t=1}^T$ generated according to the Assumption 1. For probability greater than $1-\frac{1}{T}$, the matrices $\mathcal{E}(\ell)$ in the equation* (7) *are bounded by:*

$$\max_{1\le\ell\le L} \|\mathcal{E}(\ell)\| \le \frac{1+3\sqrt{2}}{3}\sqrt{\frac{\mathcal{V}\log(2pT^2)}{B}} + \frac{B\Gamma}{2}. \tag{8}$$

when the block size B is larger than $\mathcal{M}^2 \log(2pT^2)/\mathcal{V}$.

Lemma 1 highlights the effect of allowing perturbations in the data generation model. From classical results in statistics, our intuition tells that the effect of noise washes out as the block size increases, i.e., the error decays with the 'inverse square root' of the block size. Hence, barring memory and data issues, a larger block size is better when $\Gamma = 0$. In contrast, covariance perturbations add error proportional to the block size. Consequently, we have a trade-off between two terms in this case, and there exists an optimal block size depending on the Γ.

We establish the convergence guarantee of the robust power method in Theorem 2. In the proof of Theorem 2 (Appendix E), we bound the distance, $d\big(\mathbf{U}(L), \hat{\mathbf{U}}(L)\big)$ between the output of Algorithm 1, $\hat{\mathbf{U}}_{1:k}(L)$ and k-orthonormal vectors spanning the column space of $\mathbf{A}_T, \mathbf{U}_{1:k}(L)$. We identify the optimal block size B, which is the unique parameter for noisy power method and establish an upper bound on the estimation error of the noisy power method.

Theorem 2 (Robust power method)**.** *Assume that $\delta \geq 0.71\sigma^2$ and $\Gamma = \mathcal{O}(\delta^3/(\mathcal{V}\log(2pT^2)))$. When the observations $\{\mathbf{x}_t\}_{t=1}^T$ satisfies the assumption 1, for $B = \Theta(\mathcal{V}^{1/3}\log(2pT^2)^{1/3}/\Gamma^{2/3})$ we have:*

$$\mathcal{R}^{\mathrm{NPM}} = \mathcal{O}\left(\frac{(\mathcal{V}\Gamma\log(2pT^2))^{1/3}}{\delta} + 0.7^{T/B}\right), \tag{9}$$

with probability $1 - 1/T - c^{\Omega(p-k+1)} - \exp(-\Omega(p))$.

When $\Gamma = \Omega(\delta^3/(\mathcal{V}\log(2pT^2)))$ we have that $\frac{(\mathcal{V}\Gamma\log(2pT^2))^{1/3}}{\delta} = \Omega(1)$. Therefore, the condition on Γ in Theorem 2 is necessary to avoid a trivial upper bound (≤ 1) on $\mathcal{R}^{\mathrm{NPM}}$. This condition encompasses several applications of interest alluded to earlier. For example, in the financial applications alluded to earlier, individual market changes of interest happen on millisecond time-scale and it is of significant interest to terminally detect cumulative market changes. Our results hold for sufficiently large value of cumulative changes (allowing for seconds time-scale) allowing us to study them. They further imply that the noisy power method is rate-optimal for non-trivial values of Γ. When the observations follow model (1), under the event $\mathfrak{E}_\Delta$, Theorem 2 shows that the robust power method can achieve a estimation error of:

$$\tilde{\mathcal{O}}\left(\left(\frac{\Gamma}{\delta}\right)^{1/3}\left(\frac{(p\sigma^2 + k\delta)(\sigma^2+\delta)}{\delta^2}\right)^{1/3}\right), \tag{10}$$

with probability $1 - \Delta - 1/T - c^{\Omega(p-k+1)} - \exp(-\Omega(p))$ if $\tilde{\delta}$ equals to δ. This is order-wise identical to the fundamental limit established in Theorem 1 when $p\sigma^2$ dominates $k\delta$.

Establishing bounds on the estimation error when the underlying singular vectors change is intricate since the subspace to which consecutive observations belong are potentially different. Conventional proofs that analyze noisy power methods or Oja's algorithm show that under a variety of assumptions at every iteration ℓ, the distance between the estimated and true subspace, $d(\mathbf{U}(\ell), \hat{\mathbf{U}}(\ell))$ decreases. For instance, the proof in [28] requires $\|\mathcal{E}(\ell)\mathbf{U}(\ell)\| = O(\sqrt{k/p})$ which does not hold under our model since $\|\mathbf{M}(\ell) - \mathbf{M}(\ell-1)\|$ is in general greater than $\sqrt{k/p}$. Similarly, the concentration approach in [32] can be used only when the covariance matrix is time-invariant. We briefly describe our proof technique to establish Theorem 2, deferring details to Appendix E. Let $\mathcal{M}^{(L)}$ denote the product $\prod_{\ell=1}^{L}\big(\mathbf{M}(\ell) + \mathcal{E}(\ell)\big)$. Then, the output of the algorithm $\hat{\mathbf{U}}_{1:k}(L)$ is an orthonormal basis of $\mathcal{M}^{(L)}\hat{\mathbf{U}}(0)$, which estimates the first k principal components $\mathbf{U}_{1:k}$ of the $\mathbf{M}(L)$. To address this, we construct sequences of k and $(p-k)$-dimensional subspaces of $\mathbb{R}^p$ from observations $\{(\mathbf{x}_t)_{t=(\ell-1)B}^{\ell B}\}_{\ell=1}^L$, denoted by $\{\mathbf{N}^{(\ell)}\}_{\ell=1}^L \in \mathrm{St}_{p-k}(\mathbb{R}^p)$ and $\{\mathbf{W}^{(\ell)}\}_{\ell=1}^L \in \mathrm{St}_k(\mathbb{R}^p)$ respectively, such that for all iterations ℓ:

(a) $\mathrm{ran}\big((\mathbf{M}(\ell)+\mathcal{E}(\ell))\mathbf{N}^{(\ell)}\big) \subseteq \mathrm{ran}\big(\mathbf{N}^{(\ell+1)}\big)$ and $\mathrm{ran}\big((\mathbf{M}(\ell)+\mathcal{E}(\ell))\mathbf{W}^{(\ell)}\big) \subseteq \mathrm{ran}\big(\mathbf{W}^{(\ell+1)}\big)$,

(b) $\|(\mathbf{M}(\ell) + \mathcal{E}(\ell))\mathbf{N}^{(\ell)}\|\|((\mathbf{M}(\ell) + \mathcal{E}(\ell))\mathbf{W}^{(\ell)})^{-1}\| < 1$,

(c) $d(\mathbf{U}_{1:k}(\ell), \mathbf{W}^{(\ell+1)}) = O(\|\mathcal{E}(\ell)\|)$ and $d(\mathbf{U}_{k+1:p}(\ell), \mathbf{N}^{(\ell+1)}) = O(\|\mathcal{E}(\ell)\|)$.

The initial random matrix $\hat{\mathbf{U}}_{1:k}(0)$ consists of both $\mathbf{N}^{(1)}$ and $\mathbf{W}^{(1)}$ with high probability. From the first two properties, at every iteration ℓ, the projection of $\hat{\mathbf{U}}_{1:k}(\ell-1)$ in $\mathbf{W}^{(\ell)}$ is amplified more than that on $\mathbf{N}^{(\ell)}$ and thus $\hat{\mathbf{U}}_{1:k}(L)$ becomes very close to $\mathbf{W}^{(L)}$ after sufficiently large L. From the last property, we can conclude that $\mathbf{W}^{(L)}$ is close to $\mathbf{U}_{1:k}(L)$, where the distance between $\mathbf{W}^{(L)}$ and $\mathbf{U}_{1:k}(L)$ is proportional to $\|\mathcal{E}(L)\|$. Combining these ideas with Lemma 1 establishes the results.

5.2 Oja's Algorithm

Algorithm 2 Oja's Algorithm with learning rate ζ [48]

1: **Input:** $\{\mathbf{x}_1\}_{t=1}^T$, Learning rate: ζ, dimensions: p, k & **Output:** $\hat{\mathbf{U}}(T)$
2: $\hat{\mathbf{U}}(0) \leftarrow p \times k$ random matrix
3: **for** $t = 1 : T$ **do**
4: $\quad \hat{\mathbf{U}}(t) \leftarrow$ Gram-Schmidt$((\mathbf{I} + \zeta \mathbf{x}_t \mathbf{x}_t^\top)\hat{\mathbf{U}}(t-1))$
5: **end for**

We now establish the convergence guarantee for Oja's Algorithm (Algorithm 2) when observations follow (1). Unlike the noisy power method, Oja's Algorithm is multiplicative in its construction of the estimated subspace. We extend the existing analysis for Oja's algorithm [34, 32, 3] by considering a virtual block with $B = B_\zeta = \zeta^{-1}$ observations. Building upon the analysis framework for the noisy power method and intuition from binomial approximation $(1+\zeta x)^{1/\zeta} \simeq (1+x)$ we establish the convergence guarantees for Oja's algorithm. Like the previous section, we decompose the block with target $\mathbf{M}(\ell)$ and error matrix $\mathcal{E}'(\ell)$ as:

$$\prod_{t=(\ell-1)B+1}^{\ell B} (\mathbf{I} + \zeta\, \mathbf{x}_t \mathbf{x}_t^\top) = \prod_{t=(\ell-1)B+1}^{\ell B} (\mathbf{I} + \zeta\, \mathbb{E}\, \mathbf{x}_{\ell B} \mathbf{x}_{\ell B}^\top) + \mathcal{E}(\ell) = \mathbf{M}_{\text{Oja}}(\ell) + e^{\tilde{\delta}+\sigma^2}\, \mathcal{E}'(\ell)\,. \tag{11}$$

where $\mathcal{E}'$ is scaled error matrix. In Lemma 2, we provide bound for scaled error matrix with respect to the learning parameter ζ, or the virtual block size B_ζ=$1/\zeta$ similar to Lemma 1. The proof is provided in Appendix F.

Lemma 2 (Spectral norm of noise, Oja's algorithm case)**.** *Assume that the observations* $(\mathbf{x}_t)_{t=1}^T$ *generated according to the Assumption 1. For probability greater than* $1 - \frac{1}{T}$*, the matrices* $\mathcal{E}'(\ell)$ *on the equation* (11) *are bounded by:*

$$\max_{1 \le \ell \le L} \|\mathcal{E}'(\ell)\| \le \sqrt{\frac{2e^2 \mathcal{M}^2 \log(pT^2)}{B_\zeta}} + \frac{B_\zeta \Gamma}{2} + \mathcal{O}(B_\zeta \Gamma)\,, \tag{12}$$

when the virtual block size $B_\zeta = \zeta^{-1}$ *is larger than* $2\mathcal{M}^2 \log(pT^2)$.

The difference in parameter $\mathcal{M}$, rather than $\mathcal{V}$ as in the case of Lemma 1 arises due to use of a different concentration inequality. The estimator of the noisy power method averages the outer product of vectors (equation (11)) making it straightforward to use Bernstein's inequality. On the other hand, Oja's algorithm averages the product of random matrices rather than the sum of random matrices. Therefore, we introduce the multiplicative concentration inequality [31], which requires the (probabilistic) norm bound for matrices.Combining multiplicative concentration inequalities with Lemma 2 and the techniques developed for noisy power method, we obtain a convergence guarantees for Oja's algorithm in Theorem 3. The complete proof is provided in Appendix G.

Theorem 3 (Oja's algorithm)**.** *Assume that* $\delta \ge 0.71$ *and* $\Gamma = \mathcal{O}(\delta^3/(e^{3(\tilde{\delta}-\delta)}\mathcal{M}^2 \log(pT^2)))$*. When the observations* $\{\mathbf{x}_t\}_{t=1}^T$ *satisfies the assumption 1, for* $1/\zeta = \Theta(\mathcal{M}^{2/3} \log(pT^2)^{1/3}/\Gamma^{2/3})$ *we have:*

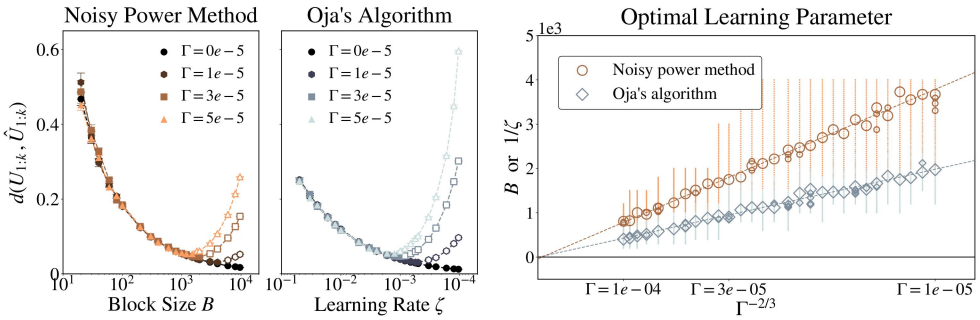
$$\mathcal{R}^{\text{Oja}} = \tilde{\mathcal{O}}\left(e^{\tilde{\delta}} \frac{(\mathcal{M}^2 \Gamma \log(pT^2))^{1/3}}{\delta} + 0.7^{\zeta T} \right)\,, \tag{13}$$

with probability $1 - 1/T - c^{\Omega(p-k+1)} - \exp(-\Omega(p))$.

Unlike the noisy power method the upper bound in Theorem 3 is $O(p^{2/3})$ rather than optimal-dependence of $O(p^{1/3})$ from Theorem 1. It is unclear whether the upper bound can be improved. Sharpening our analysis with a two-phase strategy [32, 39] (wherein the first phase identifies a good initial point and the second phase establishes convergence given an initial point) might be an excellent direction for future investigation.

6 Numerical Results

Lastly, in order to verify the established results for both algorithms, we examine the performance of algorithms for a variety of environments . Theorem 2 and 3 show the **(a)** existence of the optimal block size B and the learning rate ζ to obtain the minimum recovery error, and **(b)** $\Gamma^{-3/2}$ dependencies of those optimal B and $1/\zeta$. We demonstrate this phenomenon in Figure 1a and Figure 1b respectively. Experimental details are described in Appendix I. Our first observation on the Figure 1a is that



(a) Distance between true column space of $\mathbf{A}_T$ ($\mathbf{U}_{1:k}$) and estimated space at $t = T$ ($\hat{\mathbf{U}}_{1:k}$) as Γ varies.

(b) Empirical optimal block size B and learning rate ζ as Γ varies.

Figure 1: Numerical results. We used the setting (σ, δ, p, k)=$(0.15, 1.0, 100, 5)$.

convergence error decreases as the block size increases without any covariance perturbation ($\Gamma = 0$). This behavior is expected since in the absence of rotation of underlying space, increasing the number of past information results in better accuracy guarantees for the recovered space. However, if the covariance perturbation exists ($\Gamma > 0$), there exists the optimal learning parameter, and we have the smaller optimal block size as the adversary stronger. This phenomenon is also the expected behavior since an increase in the adversarial budget implies that the past information becomes less relevant. Our observations also corroborate our theoretical results in the Theorem 1. In Figure 1b, we focus on the optimal value of the block size B and inverse of the learning rate $1/\zeta$ and its variation with the perturbation budget Γ. We plot the empirically optimal learning rate for the case of the noisy power method and Oja's algorithm with $\Gamma^{-2/3}$. We observe that the optimal block size and the inverse learning rate is proportional to $\Gamma^{-2/3}$. This experimental dependency of $\Gamma^{-2/3}$ verifies the theoretically prescribed results in Theorem 2 and Theorem 3.

7 Conclusion

In this paper, we analyzed the fundamental limit of the minimax error and convergence of the noisy power method and Oja's algorithm on the streaming PCA settings with time-varying covariance. Under this framework, when there exists no perturbation ($\Gamma = 0$), our theoretical result on a lower limit coincides with the order of the traditional outcome. Furthermore, when perturbation exists ($\Gamma > 0$), we have a non-avoidable positive minimax recovery error, although the time horizon becomes arbitrarily long. Next, we found that the noisy power method order-wisely achieves this recovery error and becomes rate-optimal if the $k\delta \ll p\sigma^2$. Finally, we showed that optimal block size and inverse learning rate minimizing recovery error were proportional to $\Gamma^{-3/2}$ both theoretically and experimentally. All our results are based on the spectral gap assumption. However, a recent line of literature considers effective rank [11, 50] as a crucial metric. Establishing guarantees for robust streaming PCA in terms of effective rank is an interesting direction for future work.

References

[1] K. Abed-Meraim, S. Attallah, A. Chkeif, and Y. Hua. Orthogonal oja algorithm. *IEEE Signal Processing Letters*, 7(5):116–119, 2000.

[2] K. Abed-Meraim, A. Chkeif, and Y. Hua. Fast orthonormal past algorithm. *IEEE Signal Processing Letters*, 7(3):60–62, 2000.

[3] Zeyuan Allen-Zhu and Yuanzhi Li. First efficient convergence for streaming k-pca: A global, gap-free, and near-optimal rate. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 487–492, 2017.

[4] Raman Arora, Andrew Cotter, Karen Livescu, and Nathan Srebro. Stochastic optimization for pca and pls. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 861–868. IEEE, 2012.

[5] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *2010 48th Annual allerton conference on communication, control, and computing (Allerton)*, pages 704–711. IEEE, 2010.

[6] Thomas Bendokat, Ralf Zimmermann, and P. A. Absil. A grassmann manifold handbook: Basic geometry and computational aspects, 2020.

[7] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[8] Daniel Bienstock and Mauro Escobar. Stochastic defense against complex grid attacks. *IEEE Transactions on Control of Network Systems*, 7(2):842–854, 2020.

[9] Daniel Bienstock and Apurv Shukla. Variance-aware optimal power flow: Addressing the tradeoff between cost, security, and variability. *IEEE Transactions on Control of Network Systems*, 6(3):1185–1196, 2019.

[10] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[11] Florentina Bunea and Luo Xiao. On the sample covariance matrix estimator of reduced effective rank population matrices, with applications to fpca. *Bernoulli*, 21(2):1200–1230, 2015.

[12] John Burkardt. The truncated normal distribution. *Department of Scientific Computing Website, Florida State University*, 1:35, 2014.

[13] Tony Cai, Zongming Ma, and Yihong Wu. Optimal estimation and rank detection for sparse spiked covariance matrices. *Probability Theory and Related Fields*, 161(3):781–815, Apr 2015.

[14] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

[15] Minshuo Chen, Lin Yang, Mengdi Wang, and Tuo Zhao. Dimensionality reduction for stationary time series via stochastic nonconvex optimization. *arXiv preprint arXiv:1803.02312*, 2018.

[16] Yeshwanth Cherapanamjeri, Prateek Jain, and Praneeth Netrapalli. Thresholding based outlier robust pca. In *Conference on Learning Theory*, pages 593–628. PMLR, 2017.

[17] Yuejie Chi, Yonina C. Eldar, and Robert Calderbank. Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations. *IEEE Transactions on Signal Processing*, 61(23):5947–5959, 2013.

[18] John H. Conway, Ronald H. Hardin, and Neil J. A. Sloane. Packing lines, planes, etc.: packings in Grassmannian spaces. *Experimental Mathematics*, 5(2):139 – 159, 1996.

[19] Wei Dai, Youjian Liu, and B. Rider. Quantization bounds on grassmann manifolds of arbitrary dimensions and mimo communications with feedback. In *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, volume 3, pages 5 pp.–, 2005.

[20] David Donoho, Matan Gavish, and Iain Johnstone. Optimal shrinkage of eigenvalues in the spiked covariance model. *The Annals of Statistics*, 46(4):1742 – 1778, 2018.

[21] G.H. Dunteman. *Principal Components Analysis*. Number No. 69 in A Sage Publications. SAGE Publications, 1989.

[22] Mauro Escobar, Daniel Bienstock, and Michael Chertkov. Learning from power system data stream. In *2019 IEEE Milan PowerTech*, pages 1–6. IEEE, 2019.

[23] Jiashi Feng, Huan Xu, and Shuicheng Yan. Online robust pca via stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 404–412, 2013.

[24] Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

[25] John Goes, Teng Zhang, Raman Arora, and Gilad Lerman. Robust stochastic principal component analysis. In *Artificial Intelligence and Statistics*, pages 266–274. PMLR, 2014.

[26] Donald Goldfarb and Garud Iyengar. Robust portfolio selection problems. *Mathematics of operations research*, 28(1):1–38, 2003.

[27] Gene H Golub and Charles F Van Loan. *Matrix computations (4th ed)*, volume 3. JHU Press, 2012.

[28] Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. In *Advances in Neural Information Processing Systems*, pages 2861–2869, 2014.

[29] Jun He, Laura Balzano, and John C. S. Lui. Online Robust Subspace Tracking from Partial Information. *arXiv e-prints*, page arXiv:1109.3827, September 2011.

[30] Amelia Henriksen and Rachel Ward. Adaoja: Adaptive learning rates for streaming pca, 2019.

[31] De Huang, Jonathan Niles-Weed, Joel A Tropp, and Rachel Ward. Matrix concentration for products. *arXiv preprint arXiv:2003.05437*, 2020.

[32] De Huang, Jonathan Niles-Weed, and Rachel Ward. Streaming k-pca: Efficient guarantees for oja's algorithm, beyond rank-one updates. In Mikhail Belkin and Samory Kpotufe, editors, *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pages 2463–2498. PMLR, 2021.

[33] T Inglot. Inequalities for quantiles of the chi-square distribution. *Probability and Mathematical Statistics*, 30:339—-351, 2010.

[34] Prateek Jain, Chi Jin, Sham M Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming pca: Matching matrix bernstein and near-optimal finite sample guarantees for oja's algorithm. In *Conference on Learning Theory*, pages 1147–1164, 2016.

[35] Iain M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295 – 327, 2001.

[36] Iain M Johnstone and Debashis Paul. Pca in high dimensions: An orientation. *Proceedings of the IEEE*, 106(8):1277–1292, 2018.

[37] Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

[38] Chris Junchi Li, Mengdi Wang, Han Liu, and Tong Zhang. Near-optimal stochastic approximation for online principal component estimation. *Mathematical Programming*, 167(1):75–97, 2018.

[39] Xin Liang. On the optimality of the oja's algorithm for online PCA. *CoRR*, abs/2104.00512, 2021.

[40] Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 581–588, New York, NY, USA, 2013. Association for Computing Machinery.

[41] André L. G. Mandolesi. Grassmann angles between real or complex subspaces, 2021.

[42] Poorya Mianjy and Raman Arora. Stochastic pca with ℓ_2 and ℓ_1 regularization. In *ICML*, 2018.

[43] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory limited, streaming pca. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2886–2894, Red Hook, NY, USA, 2013. Curran Associates Inc.

[44] Paul Mooney. Stock Market Data (NASDAQ, NYSE, S&P500). `https://www.kaggle.com/datasets/paultimothymooney/stock-market-data`, Version 64.

[45] Jonas W Mueller, Vasilis Syrgkanis, and Matt Taddy. Low-rank bandit methods for high-dimensional dynamic pricing. *Advances in Neural Information Processing Systems*, 32, 2019.

[46] Praneeth Narayanamurthy and Namrata Vaswani. Provable dynamic robust pca or robust subspace tracking. *IEEE Transactions on Information Theory*, 65(3):1547–1577, 2018.

[47] Yuri Neretin. On jordan angles and the triangle inequality in grassmann manifolds. *Geometriae Dedicata*, 86:81–91, 06 2001.

[48] Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of mathematical analysis and applications*, 106(1):69–84, 1985.

[49] Daniel L Pimentel-Alarcón, Aritra Biswas, and Claudia R Solís-Lemus. Adversarial principal component analysis. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2363–2367. IEEE, 2017.

[50] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007.

[51] Ohad Shamir. Convergence of stochastic gradient descent for pca. In *International Conference on Machine Learning*, pages 257–265, 2016.

[52] Ohad Shamir. Fast stochastic algorithms for svd and pca: Convergence properties and convexity. In *International Conference on Machine Learning*, pages 248–256, 2016.

[53] Wenjie Song, Jianke Zhu, Yang Li, and Chun Chen. Image alignment by online robust pca via stochastic gradient descent. *IEEE Transactions on Circuits and Systems for video Technology*, 26(7):1241–1250, 2015.

[54] Le Trung Thanh, Nguyen Viet Dung, Nguyen Linh Trung, and Karim Abed-Meraim. Robust subspace tracking with missing data and outliers: Novel algorithm with convergence guarantee. *IEEE Transactions on Signal Processing*, 69:2070–2085, 2021.

[55] Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[56] Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

[57] Vincent Q. Vu and Jing Lei. Minimax sparse principal subspace estimation in high dimensions. *The Annals of Statistics*, 41(6):2905 – 2947, 2013.

[58] Sissi Xiaoxiao Wu, Hoi-To Wai, Anna Scaglione, and Neil A. Jacklin. The power-oja method for decentralized subspace estimation/tracking. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3524–3528, 2017.

[59] Bin Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43(1):95–107, 1995.

[60] Puyudi Yang, Cho-Jui Hsieh, and Jane-Ling Wang. History pca: A new algorithm for streaming pca. *arXiv preprint arXiv:1802.05447*, 2018.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] The datasets are public datasets and the code is MIT license

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We explain our new framework and contributions in both abstract and introduction.
   (b) Did you describe the limitations of your work? [Yes] There is a result which is partially suboptimal compare to the edge works. We also describe and study the reason for that.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We provide our assumptions on the Section 3 and Assumption 1
   (b) Did you include complete proofs of all theoretical results? [Yes] Detailed Proofs are in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Those are included in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Those are included in the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Yes, we repeated every experiment 10 times and reported error bars or uncertainty markers.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Those are included in the Appendix.
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We only used synthesized random matrix data.
4. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]