Streaming PCA with Many Missing Entries

Ioannis Mitliagkas
The University of Texas
Austin, TX 78712 USA
ioannis@utexas.edu

Constantine Caramanis
The University of Texas
Austin, TX 78712 USA
constantine@utexas.edu

Prateek Jain
Microsoft Research India
Bangalore, INDIA
prajain@microsoft.com

ABSTRACT

We consider the streaming memory-constrained principal component analysis (PCA) problem with missing entries, where the available storage is linear in the dimensionality of the problem, and each vector has so many missing entries that matrix completion is not possible. SVD-based methods cannot work because of the memory constraint, while imputation-based updates fail when faced with too many erasures. For this problem, we propose a method based on a block power update approach introduced in [14]. We show on synthetic as well as benchmark data sets, that our approach outperforms existing approaches for streaming PCA by a significant margin for several interesting problem settings. We also consider the popular spiked covariance model with randomly missing entries, and obtain the first known global convergence guarantees for this problem. We show that our method converges to the true "spike" using a number of samples that is linear in the dimension of the data. Moreover, our memory requirement is also linear in the ambient dimension. Thus, both memory and sample complexity have optimal scaling with dimension.

Keywords

PCA, erasures, missing entries, streaming, memory-limited, data discovery

1. INTRODUCTION

Dimensionality reduction is a foundational technique in statistics and many other fields of science and engineering. Principal Component Analysis (PCA) searches for a linear manifold with maximal "explanatory power." This paper considers the problem of PCA, under *severe memory/storage constraints, and partial observation* – a setting where to the best of our knowledge, there are no known algorithms with global performance guarantees. We consider the streaming setting, where we see data points sequentially, and these are nowhere stored. We seek to use *no more total storage than required for the output*. This essentially means that while our algorithm sees each data point, it can only do so once.

Motivated by many recent applications in preference and behavior modeling, we focus on the partial observation setting: each data point is not only noisy as in traditional PCA. It also suffers some – perhaps overwhelming – number of erasures. Extreme erasures are typical of application areas, where the data is naturally sparse. Moreover, erasures may be an introduced feature, e.g. where features are withheld to protect privacy, while allowing collective learning. This distinction between extracting principle components and *completing* the data, is important in our work.

We consider the following problem: given partial observations $\dot{\mathbf{x}}_t$ of vectors $\mathbf{x}_t \in \mathbb{R}^p$, we seek to find a k-dimensional subspace U along which the variance of complete vectors $\{\mathbf{x}_i\}$ is maximized. Matrix completion techniques using either SVD [11] or Nuclear norm optimization approaches (e.g., [5, 15, 17, 12]) have formed the bulk of research into such problems, and in many settings, their algorithmic and statistical performance is now quite well understood. Yet these algorithms all have storage complexity on the order of $O(p^2)$. Moreover, most of these results focus on *matrix completion* – an objective of less value in the setting where we cannot store points.

In contrast, we focus on recovering a subspace close to U. If there are enough observations, this is equivalent to matrix completion, but this is not always the case. Recent work on memory-restricted PCA has considered this objective, including [14, 1, 2, 3]. Of these, [14] is the only one that guarantees optimal sample complexity and global convergence, yet does not consider erasures; [3, 4] is the only one that can handle erasures, though convergence guarantees there are only local. The algorithms in [3] (GROUSE) and [16] (Stochastic Approximation, finally analyzed in [2]) perform very well in general but they share two important drawbacks: i) their performance can suffer when the number of erasures is overwhelming and ii) their success critically depends on very careful parametrization individualized for every dataset.

Our contributions: We provide the first algorithm for streaming PCA with erasures, with global convergence guarantees. The main focus of our work is to obtain algorithms with *optimal sample complexity, and optimal memory complexity.* Specifically, we show:

- **Algorithm and Performance**: we provide a simple efficiently-computable algorithm that has the form of a block power method update. We simulate on synthetic and benchmark data, and demonstrate the suc-

KDD '14 New York, NY USA

cess of our algorithm. Perhaps its most salient quality is the fact that it obviates the need for "guesswork" when deploying on a new dataset. That is, unlike other streaming algorithms, the same exact parametrization, can perform well in different datasets. See Section 6 for the numbers and discussion.

- **Sample Complexity**: We consider the setting where the unerased samples $\mathbf{x}_i$ are drawn independently from a wide distribution family. Each sample entry is then observed with probability δ. See Section 2 for complete description of the model. We show that our algorithm can recover the distribution's k principal components with $\tilde{O}(p/\delta^2\epsilon^2)$ samples, which is scaling-wise optimal for any algorithm. Furthermore, we show – in theory and experiments – that we can recover U even when $\delta p < k$. In this regime, *matrix completion is generally not possible.* To do this, we improve on the analysis in [13] providing tighter bounds for the model.

- **Memory Complexity**: Our algorithm requires memory $O(pk)$ – this is the best possible. This much memory is required to store the output alone.

Notation: Unless stated explicitly, $||A||$ denotes the spectral norm, i.e., the natural norm induced by the L2 vector norm. If U is a matrix, $U_\perp$ denotes an orthogonal basis for the subspace perpendicular to span(U). For $\mathbf{x} \in \mathbb{R}^p$ and $\Omega \subseteq [p]$, we use $\mathbf{x}_\Omega$ to denote the restriction of $\mathbf{x}$ to the elements in the set Ω. Finally, if U is a matrix, we denote its ith row by U^i and its ith column by $\mathbf{u}_i$. Similarly for a vector $\mathbf{x}$, we write x^i for its ith entry.

2. PROBLEM FORMULATION

System Model. Assume that at each time step t, we receive a point $\dot{\mathbf{x}}_t$, which is a partially erased version of $\mathbf{x}_t \in \mathbb{R}^p$. Our goal is to compute the top k principal components of the data: the k-dimensional subspace that offers the best squared-error estimate for the points. Our total storage capacity is $O(kp)$ – the storage required to store the output. The streaming setting means, in particular, that any vector not explicitly stored can never be revisited.

Our analytical (sample complexity) guarantees are based on the following generative model for the data: the full samples are described by

$$\mathbf{x}_t = U\Lambda\mathbf{z}_t + \mathbf{w}_t, \tag{1}$$

where each component of $\mathbf{z}_t$, i.e., $\mathbf{z}_t^i, 1 \le i \le p$ is sampled i.i.d. from a fixed distribution $\mathcal{D}$, s.t., $\mathbb{E}[\mathbf{z}_t^i = 0]$, $\mathbb{E}[(\mathbf{z}_t^i)^2 = 1]$, and finally $|\mathbf{z}_t^i| \le M_\infty$ almost surely. Similarly, we assume that each component of $\mathbf{w}_t$ is sampled i.i.d. from another fix distribution $\mathcal{D}'$ which also satisfies the same set of normalization constraints, i.e., $\mathbb{E}[\mathbf{w}_t^i = 0]$, $\mathbb{E}[(\mathbf{w}_t^i)^2 = 1]$, and $|\mathbf{w}_t^i| \le M_\infty$ almost surely. Note that, our analysis holds even when $\mathbf{z}_t, \mathbf{w}_t$ are sampled from any general fixed sub-Gaussian. However, we assume bounded distribution for simplicity of exposition.

Also, we assume that sequences $\{\mathbf{z}_t\}_t$ and $\{\mathbf{w}_t\}_t$ are mutually independent.

Moreover, let $U \in \mathbb{R}^{p\times k}$ be a matrix with orthonormal columns and $\Lambda \in \mathbb{R}^{k\times k}$ a diagonal matrix. Note that we can generalize our model to include V (an orthonormal matrix) in the measurement as well, i.e., $\mathbf{x}_t = U\Lambda V^T\mathbf{z}_t + \mathbf{w}_t$. However, as $\mathbf{z}_t$ is a spherically symmetric variable, recovering V is not possible and hence, WLOG, we can assume $V = I$.

Finally, we assume that the observed samples, $\dot{\mathbf{x}}_t$, are erased versions of $\mathbf{x}_t$, where for each entry j, independently,

$$\dot{\mathbf{x}}_t(j) = \begin{cases} \mathbf{x}_t(j) & w.p.,\ \ \delta \\ 0, & otherwise \end{cases}. \tag{2}$$

Hence, each vector we observe has δp observed entries in expectation.

Objective and Metric. Given data stream $\{\mathbf{x}_1, \mathbf{x}_2, \dots\}$, the standard goal of streaming PCA is to recover the variance-maximizing subspace, i.e., of maximizing the *explained variance.* However, for our generative model, this corresponds to recovering the subspace spanned by the orthonormal matrix U. Now, we measure the error in estimation of the required subspace using the largest principle angle based distance ([20]). That is, given any unitary matrix Q, we use the following distance:

$$\begin{aligned} \text{dist}(U, Q) =& \text{dist}\left(\text{span}(U), \text{span}(Q)\right) \\ =& \|U_\perp^\top Q\|_2 = \|Q_\perp^\top U\|_2 \end{aligned} \tag{3}$$

The distance is symmetric and takes values in $[0, 1]$. Given enough samples, the minimization of (3) is equivalent to maximizing the explained variance. In our experiments, Section 6, we use both metrics.

We now present some of the related works to the above mentioned streaming PCA with missing entries problem. We then present our algorithm and analysis in Section 4, and Section 5, respectively.

3. RELATED WORK

As mentioned above, to the best of our knowledge, this paper represents the only work that is able to address this problem, in the setting of limited memory and partial observations, while providing global rates of convergence.

The literature contains a vast body of work dealing with the different aspects of this problem. Most methods are not applicable in our setup: they are either batch algorithms or require more than $O(kp)$ memory. Indeed, any algorithm that involves the computation of the empirical covariance matrix, including the standard PCA algorithm in the fully observed case, requires $O(p^2)$ storage. This essentially rules out all optimization-based solutions, including matrix completion style algorithms that do not explicitly force a low-rank factorization, such as [15, 5]. Yet algorithms that force such a factorization, are no longer convex; this is at the core of the challenge for obtaining global guarantees.

For the complete observation setting, stochastic approximation [16], and related stochastic gradient-based methods (e.g., [1]) fall into this category, and accordingly are memory-efficient. While empirically they have been observed to do well, there are no guarantees of their convergence with sample complexity order-wise comparable to batch PCA algorithms (the impressive recent work in [2] is the first to provide any convergence rate for stochastic approximation (there called *incremental PCA*), though it is orderwise worse than batch or our earlier work in [14]).

An important line of work for the setting of missing variables, is the Expectation-Maximization (EM) approach [7]. For this setting, there are no global guarantees for EM, nor is it clear how the M-step would be modified without violating the memory constraint.

For the partially observed setting which is of interest here, there are two directions stand out: covariance estimation and imputation-based algorithms. We provide detailed discussion of the tools and issues in the remainder of this section. We refer to our earlier work in [14] and [1], for a more comprehensive literature review for the full observation setting.

3.1 Unbiased Covariance Estimation

A critical element of many PCA algorithms is some form of covariance estimation, be that explicit or implicit. The former is true for the classic batch PCA algorithm. The algorithm computes the empirical covariance matrix,

$$\Sigma_n = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T, \tag{4}$$

and then performs a Singular Value Decomposition (SVD) on Σ_n to recover the range of U. The statistical limits of this process are characterized in [10]. Specifically, $O(p)$ samples are necessary in the full-rank, subgaussian case. This includes the spiked covariance model.

The introduction of erasures in the data stream, renders the estimator in (4) biased. The authors in [13] discuss this issue for the batch setting and provide an alternative algorithm. It is based on,

$$\tilde{\Sigma}_n = \delta^{-2}\Sigma_n + (\delta^{-1} - \delta^{-2})\mathrm{diag}(\Sigma_n), \tag{5}$$

and employs regularized optimization to make the method efficient in the high-dimensional case. That algorithm is not applicable in a streaming setup, however the estimator in (5) and accompanying concentration analysis, provided therein is a useful tool for our purposes.

3.2 Imputation-based Algorithms

A line of empirically successful algorithms introduced in [3] (GROUSE) and studied further in [8] and [4], avoid covariance estimation. To that end, they use an updates that resemble stochastic approximation, except they are performed along the Grassmanian manifold. In its general form, the algorithm first calculates the projection of the latest sample, $\dot{\mathbf{x}}_t$, on the current subspace estimate, say Q_t. As per the model, only a subset Ω_t of indices is observed, i.e. $\dot{\mathbf{x}}_t|_{\Omega_t} = \mathbf{x}_t|_{\Omega_t}$ and $\dot{\mathbf{x}}_t|_{\overline{\Omega}_t} = 0$. Restricting $\dot{\mathbf{x}}_t$ and Q_t to the observed indices, the projection is calculated as follows:

$$\mathbf{w}_t = \mathrm{argmin}_{\mathbf{w}\in\mathbb{R}^k} \left\|\dot{\mathbf{x}}_t|_{\Omega_t} - Q_t|_{\Omega_t}\mathbf{w}\right\|_2. \tag{6}$$

Then, Q_t and the optimal weights in $\mathbf{w}_t$ are used to impute the entries missing from $\dot{\mathbf{x}}_t$.

$$\dot{\mathbf{x}}_t|_{\overline{\Omega}_t} \leftarrow Q_t|_{\overline{\Omega}_t}\mathbf{w}_t \tag{7}$$

Finally, the algorithm uses the imputed vector to update Q_t, performing a descent step on the Grassmanian.

This method has proven to perform well in practice. However, in the regime where the number of observed elements per vector ($|\Omega_t|$) is less than the number of components (k), the projection in (6) is underdetermined, making the step ill-defined. Picking the minimum-norm solution is a reasonable way to deal with this issue and we put this idea to the test in our experiments (Section 6).

Another natural way to modify these algorithms to deal with this case, is discarding all samples with an insufficient number of observed entries (less than k). This makes a very small difference in experiments – not included here for brevity – but there is a simple probabilistic argument against it: Assuming each entry is observed independently, the number of observed entries is given by a binomial random variable (more generally Poisson trials). For $k = (c+1)\delta p$, with $c > 0$, a Chernoff bound gives

$$\mathbb{P}\left(|\Omega_t| \geq k\right) \leq \exp\left\{-\left(c^2 \wedge c\right)\delta p/3\right\}. \tag{8}$$

This implies that, for any $c > 0$, the number of wasted samples would range from large to overwhelming, depending on the scaling of δp.

We conclude that methods based on projection-based imputation face significant problems in the regime of many missing entries and set out to provide an alternative.

4. ALGORITHM

We now present our algorithm (see Algorithm 1) for the problem of streaming PCA. Our algorithm is based on the block-wise update introduced in [14]. At a high level, the algorithm essentially leverages concentration of the sample covariance to the true covariance (see Theorem 2, Theorem 11) to estimate the next iterate.

Algorithm 1, takes in the stream of data vectors $\mathbf{x}_i$, the (known) probability of observation δ, the number of components k, and a block size B. It starts with a random k-dimensional subspace and refines that estimate doing a single pass over the data. Every subset of B subsequent samples is considered a block, even though only one sample is held in memory at any time.

To see why this algorithm works, consider line 7 of the algorithm and over the course of block τ:

$$\begin{aligned} S_\tau =& \frac{1}{B}\sum_{t=B(\tau-1)+1}^{B\tau}\left[\frac{1}{\delta^2}\mathbf{x}_t\mathbf{x}_t^\top + \left(\frac{1}{\delta} - \frac{1}{\delta^2}\right)D_t\right]Q_{\tau-1} \\ =& \left[\frac{1}{B}\sum_{t=B(\tau-1)+1}^{B\tau}\frac{1}{\delta^2}\mathbf{x}_t\mathbf{x}_t^\top + \left(\frac{1}{\delta} - \frac{1}{\delta^2}\right)D_t\right]Q_{\tau-1} \\ =& \tilde{\Sigma}_B Q_{\tau-1}, \end{aligned}$$

where $D_t = \mathrm{diag}(\mathbf{x}_t\mathbf{x}_t^\top)$. From the last line, we see that after every block, the algorithm is equivalent to performing a power iteration step. That is, the previous subspace estimate, $Q_{\tau-1}$, is essentially premultiplied by the estimator in (5) using all the samples in the block. The complication is that, with every block, the covariance estimate, $\tilde{\Sigma}_B$, is different. As we know from [14], this complicates the analysis requiring more advanced tools when compared to the simpler analysis of the classic power method.

It should be noted that, even though the algorithm effectively performs a power iteration per block, $\tilde{\Sigma}_n$ is never formed explicitly – all of the calculations can be performed in $O(kp)$ memory.

In Section 3 we discuss connections to other recent work, related to this problem and algorithm and in Section 5 we provide theoretical guarantees for the convergence of Algorithm 1.

5. CONVERGENCE ANALYSIS

In this section we give theoretical guarantees for the convergence of Algorithm 1. In particular, we can show the fol-

Algorithm 1

```
Input: $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, $\delta$, $k$, Block size: $B$
$H^i \sim \mathcal{N}(0, I_{p\times p}), 1 \le i \le k$ (Initialization)
$H \leftarrow Q_0 R_0$ (QR-decomposition)
for $\tau = 1, \ldots, n/B$ do
    $S_\tau \leftarrow 0$
    for $t = B(\tau - 1) + 1, \ldots, B\tau$ do
        $D_t \leftarrow \mathrm{diag}(\mathbf{x}_t \mathbf{x}_t^\top)$
        $S_\tau \leftarrow S_\tau + \frac{1}{B}\left[\frac{1}{\delta^2}\mathbf{x}_t\mathbf{x}_t^\top + \left(\frac{1}{\delta} - \frac{1}{\delta^2}\right) D_t\right] Q_{\tau-1}$
    end for
    $S_\tau = Q_\tau R_\tau$ (QR-decomposition)
end for
Return: $Q_\tau$
```

lowing convergence result for our algorithm (Algorithm 1).

Theorem 1. *Consider a data stream, where $\dot{\mathbf{x}}_t \in \mathbb{R}^p$ for every t is generated by (2), and the SVD of $A \in \mathbb{R}^{p\times k}$ is given by $A = U\Lambda V^\top$. Let, WLOG, $\lambda_1 = 1 \ge \lambda_2 \ge \cdots \ge \lambda_k > 0$.*

Furthermore, let A be μ-incoherent, i.e., $\|U^i\|_2 \le \frac{\mu\sqrt{k}}{\sqrt{p}}$, where U^i is the i-th row of U. Finally, let

$$T = \Omega\left(\log(p/k\epsilon)/\log\left(\frac{\sigma^2 + 0.75\lambda_k^2}{\sigma^2 + 0.5\lambda_k^2}\right)\right),$$

$$B = \Omega\left(\frac{M_\infty^2\left(\frac{k\mu^2}{p} + \sigma^2 + pk^2(\frac{\mu^2}{p} + \sigma^2)^2\right)\cdot\log(p\cdot T)}{\log^2((\sigma^2 + .75)/(\sigma^2 + .5))\cdot\delta^2\epsilon^2}\right).$$

Then, after T B-size-block-updates, w.p. 0.99, $dist(U, Q_T) \le \epsilon$. Hence, the sufficient number of samples for ϵ-accurate recovery of all the top-k principal components is:

$$n = \tilde{\Omega}\left(\frac{M_\infty^2\left(\frac{k\mu^2}{p} + \sigma^2 + pk^2(\frac{\mu^2}{p} + \sigma^2)^2\right)\cdot\log(p)}{\lambda_k^4\epsilon^2\delta^2\log\left(\frac{\sigma^2+0.75\lambda_k^2}{\sigma^2+0.5\lambda_k^2}\right)}\right).$$

We use $\tilde{\Omega}(\cdot)$ to suppress the extra $\log(T)$ factor.

Note that, the number of samples required by our algorithm depends on the *incoherence parameter* μ of U, which is defined as:

$$\mu = \sqrt{p}\max_i \|U^i\|_2, \tag{9}$$

where U^i is the i-th row of U. Hence, incoherent matrices, i.e., ones with small μ, are "spread" out matrices where one of the rows of U do not dominate the others.

Incoherence parameter plays a critical role in understanding sample complexity of several problems where the entries are missing randomly [6]. The reason being, if the matrix is *not* incoherent, then there are a small number of entries that contain most of the norm (or energy). Hence, recovering such matrices/principal components become significantly more challenging.

We would like to stress that the main novelty of our analysis is that it is able to exploit this incoherence condition to guarantee fast convergence to the covariance matrix, that is critical in obtaining small sample complexity; see Section 5.1 for more discussion. Moreover, due to our block updates, we are able to show that each of our update is incoherent as well. This property is also critical for obtaining tight sample complexity bounds. We would also like to highlight that some of the existing methods [4] struggle to maintain this property, and hence suffer from relatively worse sample complexity bounds.

To bring out the key ideas in our convergence analysis and for notational simplicity, we first analyse the special case of our problem when the goal is to recover only one principal component, i.e., $k = 1$. For this special case, we first state a few lemmas that are useful for the derivation of our main result. The proofs are deferred to the appendix. Note that statements of lemmata from other sources have been adapted to use our own notation, described in Section 1.

Then, in Section 5.2, we present our proof sketch for the general rank-k case. Due to space constraints, we provide detailed analysis of the rank-k case in the full version of the paper [**?**].

5.1 Rank-one Case

Recall that, for the rank-one case, the observed data $\mathbf{x}_t$ is sampled from the following generative model:

$$\mathbf{x}_t = \mathbf{u}z_t + \mathbf{w}_t, \quad \dot{\mathbf{x}}_t = P_{\Omega_t}(\mathbf{x}_t), \quad P_{\Omega_t}(\mathbf{x}_t)^i = \delta_t^i x_t^i,$$
$$\delta_t^i = \begin{cases} 1 & w.p.\ \delta \\ 0 & otherwise \end{cases}, \tag{10}$$

where $\mathbb{E}[z_t] = 0$, $\mathbb{E}[z_t^2] = 1$ and $|z_t| \le M_\infty$ (almost surely). Similarly, each element w_t^i, of the vector $\mathbf{w}_t$ is also sampled independently from a zero-mean bounded distribution. That is, $\mathbb{E}[w_t^i] = 0$, $\mathbb{E}[(w_t^i)^2] = \sigma^2$, and $|w_t^i| \le M_\infty$.

A key component of our proof is to analyze the rate at which the sample covariance matrix $\widetilde{\Sigma} = \frac{1}{\delta^2}\frac{1}{B}\sum_t \mathbf{x}_t\mathbf{x}_t^T - (\frac{1}{\delta^2} - \frac{1}{\delta})\sum_t diag(\mathbf{x}_t\mathbf{x}_t^T)$ converges to the true covariance matrix Σ. While the result of [13] directly applies to our model (see Theorem 11), we will show below that for the specific case of spiked covariance model, their result is significantly sub-optimal and can be improved upon significantly. In particular, we will provide our bounds in terms of the incoherence parameter μ (see (9)) and the noise variance σ^2, and then later show that for a large class of incoherence and noise variance values, our result is significantly better than that of [13].

Now, we present our concentration inequality for covariance estimation:

Theorem 2. *Let $\widetilde{\Sigma} = \frac{1}{\delta^2 n}\sum_{t=1}^B \dot{\mathbf{x}}_t\dot{\mathbf{x}}_t{}^T - \frac{1}{n}\cdot\left(\frac{1}{\delta^2} - \frac{1}{\delta}\right)\cdot\sum_{t=1}^n diag(\dot{\mathbf{x}}_t\dot{\mathbf{x}}_t{}^T)$ where $\dot{\mathbf{x}}_t$ is generated using (10). Also, let $\Sigma = uu^T + \sigma^2 I$, $\|u\|_2 = 1$, and let,*

$$B \ge \frac{100M_\infty^2\log(p\cdot T)\left(2\frac{\mu^2}{p} + 4\sigma^2\mu^2 + \sigma^4 p\right)}{\delta^2\epsilon^2}.$$

Then, w.p. $\ge 1 - 1/T^2$:

$$\|\widetilde{\Sigma} - \Sigma\|_2 \le \epsilon.$$

The above result shows that if σ or μ as well as the observation probability $\delta \le \log p/p$, then we need around $O(p^3/\epsilon^2)$ samples to reduce error between $\widetilde{\Sigma}$ and Σ to be ϵ. This matches the result of [13] (see Theorem 11). However, for certain reasonable settings of σ and μ, our results can be significantly better than that of Theorem 11. We present the result in the corollary below:

Corollary 3. *Let the incoherence of u (see (10)) be a constant and let $\sigma^2 \le \frac{1}{\sqrt{p}}$. Also, let $\delta < \log p/p$. Then, if*

$B \geq Cp \log^3 p/\epsilon$ for a global constant $C > 0$ depending only on M_∞, we have (w.p. $\geq 1 - 1/T^2$): $\|\widetilde{\Sigma} - \Sigma\|_2 \leq \epsilon$.

In contrast, Theorem 11 requires $O(p^2)$ samples even when $\sigma = 0$ while μ and δ are set as above. Moreover, $\sigma^2 \leq \frac{1}{\sqrt{p}}$ is till an interesting setting for our problem, as the signal to noise ratio is still asymptotically zero, i.e., $\|u\|/\|w_t\|_2 \to 0$ as $p \to \infty$. Moreover, incoherence is a fairly standard assumption that appears in several real-world scenarios (see [6, 9]).

We now present our result for the rank-one online PCA with missing entries.

THEOREM 4. *Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ be obtained using the spiked covariance model given by* (10)*. Moreover, set the number of iterations to be $T = \frac{100 \log(1/\epsilon)}{\log((\sigma^2+.75)/(\sigma^2+.5))}$. Also, let the block size B satisfies:*

$$B \geq \frac{C \cdot M_\infty^2 \left(\frac{\mu^2}{p} + \sigma^2 + p(\frac{\mu^2}{p} + \sigma^2)^2\right) \cdot \log(p \cdot T)}{\log^2((\sigma^2 + .75)/(\sigma^2 + .5)) \cdot \delta^2 \epsilon^2},$$

where $C > 0$ is a global constant. Then, w.p. $\geq .99$ Algorithm 1 outputs iterate q_T, s.t., $\|q_T - u\| \leq \epsilon$.

The above theorem shows that the sample complexity of our algorithm grows as $n \geq \frac{C \cdot M_\infty^2 \left(\mu^2/p + \sigma^2 + p(\mu^2/p+\sigma^2)^2\right) \cdot \log p/\epsilon}{\log^2((\sigma^2+.75)/(\sigma^2+.5)) \cdot \delta^2 \epsilon^2}$.

Our proof of the above result critically relies on the following lemma:

LEMMA 5. *Let $Q_\tau \in \mathbb{R}^p$ be the iterate obtained after τ iterations of Algorithm 1. Also, let Q_τ be $10 \cdot \mu$ incoherent. Also, let the block size B be as given in Theorem 4. Then, w.p. $\geq 1 - 1/T^2$: $\|U^T \left(\widetilde{\Sigma} - \Sigma\right) Q_\tau\|_2 \leq 2\epsilon \frac{1}{\sqrt{p}}$.*

Note that the above given lemma *only* holds when each of the iterate q_τ is at most $10 \cdot \mu$ incoherent. Hence, an additional challenge here is to show that each of the iterate is indeed $10 \cdot \mu$ incoherent (w.h.p.). To this end, we provide the below given lemma. Note that, to prove this lemma we are able to explicitly use the fact that the block updates ensure that the update is similar to the one obtained by the population covariance, hence incoherence is maintained. In contrast, existing stochastic gradient based approach such as [4] are not able to show incoherence of each iterate due to large deviations from the population covariance behavior. Hence, the sample complexity of such methods can be suboptimal by a multiplicative factor of p samples.

LEMMA 6. *Let $q_{\tau+1}$ be the $(\tau+1)$-th iterate of Algorithm 1, and let B, T be as selected in Theorem 4. Also, let the spiked direction u be μ incoherent and let μ_τ be the incoherence parameter (see* (9)*) of iterate q_τ, $\forall 1 \leq \tau \leq T$. Then, w.p. $\geq 1 - 1/T^2$, for each τ, $q_{\tau+1}$ is $\mu_\tau(1+\epsilon)$-incoherent. Moreover, $q_{\tau+1}$ is at most $10 \cdot \mu$ incoherent where $\tau < T$.*

Now, using the above mentioned lemmas, we can complete the proof of Theorem 4. Our proof uses arguments similar to the ones used by the proof of Theorem 1 by [14]. See Appendix A for the proof.

5.2 Rank-k Case

We now discuss proof of our result (Theorem 1) for the general rank-k case. As the details are similar to the rank 1 case, we defer them to the full version in the interest of space. As mentioned in the previous section, the main tool here again is a strong concentration bound for the sample covariance matrix. Another critical step for our rank-1 analysis was the proof of incoherence of each iterate, which also can be extended to the general rank-k case.

After obtaining the above mentioned two key results, we can then complete the proof using an analysis similar to that of Theorem 3 in [14]. Note that, WLOG, we can assume that $\|A\|_2 = 1$.

First, we present our general result for the covariance estimation in the spiked covariance model (see Section 2).

THEOREM 7. *Let $\widetilde{\Sigma} = \frac{1}{\delta^2 n} \sum_{t=1}^{B} \dot{\mathbf{x}}_t \dot{\mathbf{x}}_t{}^T - \frac{1}{n} \cdot \left(\frac{1}{\delta^2} - \frac{1}{\delta}\right) \cdot \sum_{t=1}^{n} diag(\dot{\mathbf{x}}_t \dot{\mathbf{x}}_t{}^T)$ where $\dot{\mathbf{x}}_t$ is generated using* (1),(2)*. Also, let $\Sigma = U \Lambda U^T + \sigma^2 I$, $\|u\|_2 = 1$, and let,*

$$B \geq \frac{100 M_\infty^2 \log(p \cdot T) \cdot (k + \sigma^2 p) \cdot (\frac{\mu^2 k}{p} + \sigma^2)}{\delta^2 \epsilon^2}.$$

Then, w.p. $\geq 1 - 1/T^2$:

$$\|\widetilde{\Sigma} - \Sigma\|_2 \leq \epsilon.$$

Note that, similar to the rank-one case, there is a large class of problems under the spiked covariance model where the above given concentration bound is significantly better than the generic bound obtained by [13]. For example, when μ is a constant, $\sigma^2 = O(1/\sqrt{p})$, and $\delta = O(\frac{\log p}{p})$.

The above lemma gives the spectral norm bound for $\widetilde{\Sigma} - \Sigma$, which is a worst case bound, i.e., $\forall v \in \mathbb{R}^p$, $\|\widetilde{\Sigma} - \Sigma\|_2 \leq \epsilon \|v\|_2$. However, to get tight sample complexity for our algorithm, we need to use the fact that even though spectral norm of the error in covariance estimation $(\widetilde{\Sigma} - \Sigma)$ is ϵ, in any *fixed* and *incoherent* direction, the projection of the error matrix is significantly smaller. That is, $|u^T(\widetilde{\Sigma} - \Sigma)q| \leq \frac{\epsilon}{\sqrt{p}}$ for any fixed unit vectors u, q with incoherence parameters μ_u, μ_q being at most $O(\mu)$, where μ is the incoherence of U (from (1)).

LEMMA 8. *Let $Q_\tau \in \mathbb{R}^p$ be the iterate obtained after τ iterations of Algorithm 1. Also, let Q_τ be $10 \cdot \mu$ incoherent. Also, let the block size B be as given in Theorem 4. Then, w.p. $\geq 1 - 1/T^2$: $\|U^T \left(\widetilde{\Sigma} - \Sigma\right) Q_\tau\|_2 \leq 2\epsilon \frac{1}{\sqrt{p}}$.*

Also, similar to the rank-1 case, we need incoherence of the intermediate iterates Q_τ for the above mentioned result to hold. Following lemma shows that the incoherence parameter of the intermediate iterates can indeed be shown to be reasonably small:

LEMMA 9. *Let $Q_{\tau+1}$ be the $(\tau+1)$-th iterate of Algorithm 1, and let parameters B, T be as selected in Theorem 4. Also, let the spiked direction U be μ incoherent, i.e., $\|U^i\| \leq \frac{\mu\sqrt{k}}{\sqrt{p}}, 1 \leq i \leq p$, where U^i is the i-th row of U. Also, let μ_τ be the incoherence parameter of iterate Q_τ, $\forall 1 \leq \tau \leq T$. Then, w.p. $\geq 1 - 1/T^2$, for each τ, $Q_{\tau+1}$ is $\mu_\tau(1+\epsilon)$-incoherent. Moreover, $Q_{\tau+1}$ is at most $10 \cdot \mu$ incoherent where $\tau < T$.*

Finally, using the above given lemmas with an analysis similar to that of Theorem 3 of [14], we can complete the proof of Theorem 1. In the interest of space in this manuscript, the full proof is deferred to a longer version.

6. EXPERIMENTS

In this section, we perform a number of experiments that corroborate our theoretical claims and provide evidence that Algorithm 1 can perform better than the state of the art in several important regions. We start with describing the algorithms used in the experiments along with any implementation considerations. Then we proceed to experiments with synthesized, artificially sparsified real data, and naturally sparse data. For all these cases we compare the algorithms based on several performance metrics and discuss their running times and robustness to parametrization. Since all the data sets are, of course, stored, we simulate the streaming and no-storage aspect for our algorithm.

Algorithm 1: The algorithm we propose. Reworking the equations on the number and size of blocks from Theorem 1 we can get an expression for T (the number of blocks) as a function of all given parameters. One important missing quantity is the ratio of eigenvalues at the cutoff point *which we do not assume we know.* For all the experiments that follow we use the following simplified formula:

$$T = C_{\text{Algo1}} \log \frac{pn\delta}{k}. \tag{11}$$

All of the parameters in the formula, are available before the start of the experiment, except for the erasure probability δ which can be very quickly and accurately estimated from the data stream, much faster than the PCA procedure itself. For all of our experiments in this manuscript, we use the constant $C_{Algo1} = \frac{1}{4}$ (see Section 6.4) and round the result to the nearest integer to get the number of blocks.

Stochastic Approximation: The most popular manifestation of Stochastic Approximation for PCA is Oja's rule ([16]). Even though it is not designed to deal with missing data, we nonetheless include it in our experiments as it is an industry standard. With every new sample ($\dot{\mathbf{x}}_t$) received, the algorithm updates its estimate based on the following rule.

$$\tilde{U}_t = U_{t-1} + \frac{C_{SA}}{t} \dot{\mathbf{x}}_t \dot{\mathbf{x}}_t' U_{t-1} \tag{12}$$

After each step, the intermediate estimate, $\tilde{U}_{t+1}$ is orthonormalized to give U_t.

The $O(\frac{1}{t})$ rule for the step size is accepted universally – see [16] and [2] for some discussion. However, to the best of our knowledge, the only complete characterization of the constant depends on the uknown eigengap at the cut-off point. For our experiments, we resort to picking a different constant C as suited to different datasets, as summarized in Section 6.4.

GROUSE: We include GROUSE in our experiments for it is a lightweight, fast and efficient algorithm, having proven to do well in most situations. For use in our experiments, we download the GROUSE Matlab code from the author's website. To make the algorithm well-defined in the region $k > \delta p$ (see discussion in Section 3), we make sure to use the pseudo-inverse operator for the projection step in (6). GROUSE is more complicated than Stochastic Approximation (see [3], or Section 3 for more references). The two algorithms, however, share a diminishing step-size, $\frac{C}{t}$. Again, we are faced with selecting a constant C_{GROUSE} and, much like Stochastic Approximation, there is no formula we can use in all cases. As we discuss in Subsection 6.4, we resort to using an individually tuned constant for every dataset.

Batch: As a simple – but not necessarily optimal – baseline for our experiments, we use the unbiased covariance estimator described in (5). This is computed bringing all the samples in memory at once, hence the characterization "batch." We only include it in our first few experiments for validation purposes. It is ommitted in the larger, real datasets as it is the most resource intensive of all algorithms considered here.

6.1 Simulations on the model

We start our experiments in a fully controlled setting. For that, we synthesize data points based on the model at (2). While this is a fairly general model, we widen our scope to real datasets in the remainder on this section.

Figure 1 demonstrates a single sample run for a case when the number of observed entries per sample is smaller than the target number of principal components.

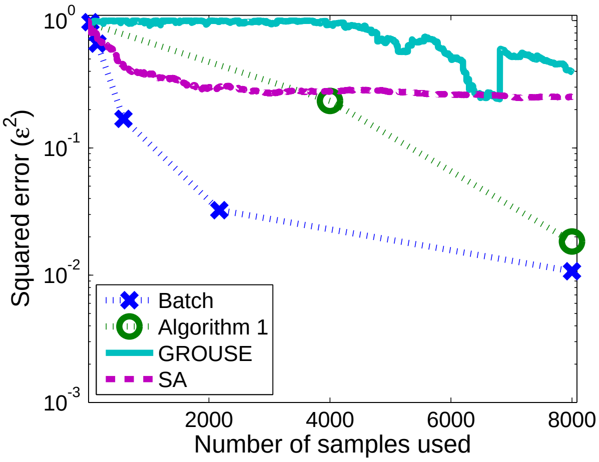

Figure 1: Example convergence curve with fewer observed entries than rank on average ($p = 20$, $k = 5$, $\delta = 0.2$, $\sigma = 0.2$).

Single-run convergence figures give us a good understanding of how things look, but are by no means evidence of a trend. To demonstrate the performance of all algorithms, we perform many independent runs in several diverse scenaria and present the averages.

Figure 2 showcases a qualitative difference between the studied algorithms. We study the transition from the region where $k < \delta p$ (more observed entries than components) to $k > \delta p$, or the *no-completion region.* Notice that the performance of Algorithm 1 deteriorates gracefully. On the other hand, imputation-based algorithms (like GROUSE) are ill-defined in that region (as discussed in Section 3) and show rapid deterioration in performance.

In Figure 3 we study the dependence of performance on the coherence of the signal components (spikes). Most algorithms show a gradual deterioration as the component becomes more coherent, with the exception of the Stochastic Approximation algorithm.

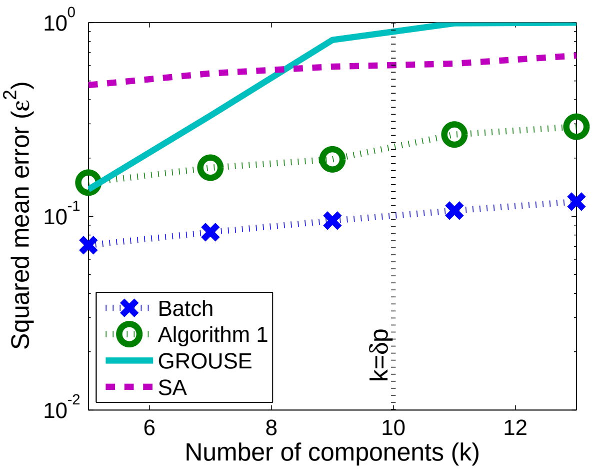

Figure 2: Transition around the boundary $k = \delta p$ ($p = 100$, $\delta = 0.01$, $\sigma = 0.2$, average of 134 runs)

6.2 Gas Sensor Array Data

For our first experiment with real data we use the gas sensor array drift dataset from [19]. It consists of 13910 samples with 128 entries each, all measurements of gas concentrations. The dataset has no missing entries and we use it as an intermediate step between synthetic and real data as follows: First we randomly permute its samples. Then we consider the samples in order, and simulate our erasure model from Section 2. That is, every entry is observed independently with probability δ. Unobserved entries are replaced with zeros. We do a predetermined number of passes over the whole dataset before reporting the final performance. To evaluate performance we use the classic metric of explained variance. Let X denote a matrix containing all samples, and let $Q \in \mathbf{R}^{p\times k}$ denote the subspace estimate provided by the algorithm. The metric of *explained variance*, is given by $||Q^T X||_F$, which we normalize with $||X||_F$ to bring into the $[0, 1]$ range.

In Figure 4, we see that Algorithm 1 is able to achieve maximal explained variance, while being more robust with respect to the choice of k. To compare the running times of the 3 algorithms we calculate the average running time in seconds per sample and report these times in Table 1.

Table 1: Average running time per processed sample

Experiment		Algorithm 1	GROUSE	SA
Gas	**k=1**	1.049e-04	1.312e-04	5.274e-05
	k=2	1.027e-04	1.306e-04	4.587e-05
	k=3	1.094e-04	1.521e-04	5.654e-05
	k=4	9.666e-05	1.347e-04	4.870e-05
	k=5	1.157e-04	1.681e-04	6.372e-05
ML	**k=5**	4.617e-02	1.796e-01	3.179e-01
	k=10	3.854e-02	3.138e-02	3.078e-02
	k=15	5.075e-02	3.711e-02	3.210e-01
	k=20	5.530e-02	6.148e-02	7.646e-02

6.3 MovieLens

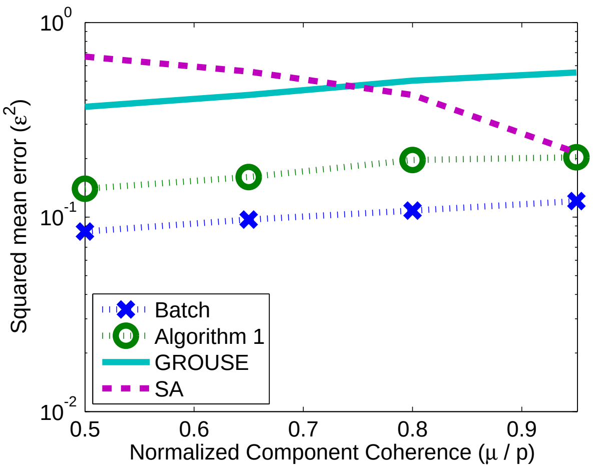

Figure 3: Performance vs coherence of the signal component ($p = 100$, $\delta = 0.05$, $\sigma = 0.2$, average of 56 runs)

In our last set of experiments, we use the MovieLens dataset from `http://grouplens.org/datasets/movielens/`. It contains about 10 million ratings for 10 thousand movies by 72 thousand users of the MovieLens service. The dataset is naturally sparse: every user only rates a tiny fraction of the movies in the database.

In this case again, there is no access to the "true" principal components, so instead of the distance metric in (3), we evaluate based on the explained variance.

To separate training from testing, we adhere to the following procedure: We first split the 10M ratings in the dataset into training and testing sets, with a 70/30 ratio. The training ratings are fed into the algorithms; each user is considered as a sample. Finally, let M_{test} denote the testing set in matrix form (movies by users) and let $Q \in \mathbf{R}^{p\times k}$ denote the subspace output by the algorithm. We evaluate based on the normalized explained variance, given by $||Q^T M_{test}||_F/||M_{test}||_F$.

In Figure 5 we see that, after *only a single pass over the dataset*, our algorithm is able to explain almost as much variance, as the batch algorithm and achieve a significant gap over GROUSE and improve over SA.

Again, the running times for this experiments are reported in Table 1.

6.4 Ease of parametrization

As discussed in this section, a common theme for GROUSE and SA is the choice of the constant used in the step-size sequence. This can prove to be a very time-consuming task, especially in the case when no ground truth information is available. A "good" constant for one experiment might be completely unsuitable for another. This forces us to look for a good parameter in every experiment we run. We went to great lengths to pick an ideal constant every time – still it is likely that slightly better choices exist. Such is the nature of this endeavor.

To demonstrate the complexity we are faced with in our

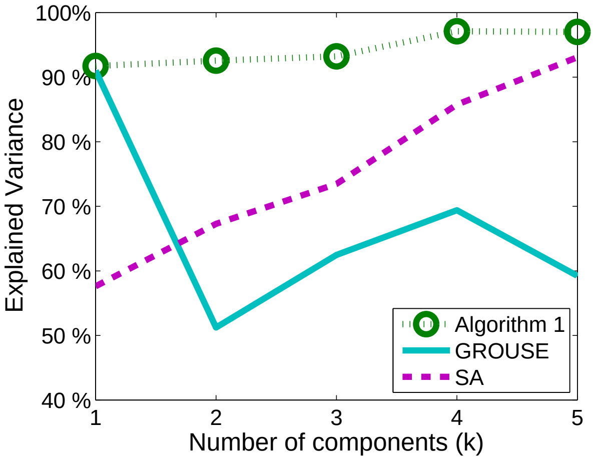

Figure 4: Performance on the gas sensor array dataset ($\delta = 0.02$, 30 independent passes)

experiments and to enable reproducibility of our results, we corral the values that were used in our real-data experiments and present them in Table 2. An important feature of Algorithm 1 that we want to emphasize here is that we were able to use *a single parameter for all of our experiments*. This makes Algorithm 1 very appealing for deployment on new datasets.

Table 2: Parametrization for real data

Experiment		C_{Algo1}	C_{GROUSE}	C_{SA}
Gas	k=1	**0.25**	2.644e-06	3.944e-06
	k=2	**0.25**	1.261e-05	1.881e-05
	k=3	**0.25**	3.158e-05	4.711e-05
	k=4	**0.25**	1.216e-04	1.813e-04
	k=5	**0.25**	2.861e-04	4.269e-04
ML	k=5	**0.25**	5.265e+00	6.582e+00
	k=10	**0.25**	2.315e+01	2.893e+01
	k=15	**0.25**	2.569e+00	3.854e+00
	k=20	**0.25**	4.652e+00	6.978e+00

7. REFERENCES

[1] R. Arora, A. Cotter, K. Livescu, and N. Srebro. Stochastic optimization for PCA and PLS. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 861–868. IEEE, 2012.

[2] A. Balsubramani, S. Dasgupta, and Y. Freund. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems*, pages 3174–3182, 2013.

[3] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 704–711. IEEE, 2010.

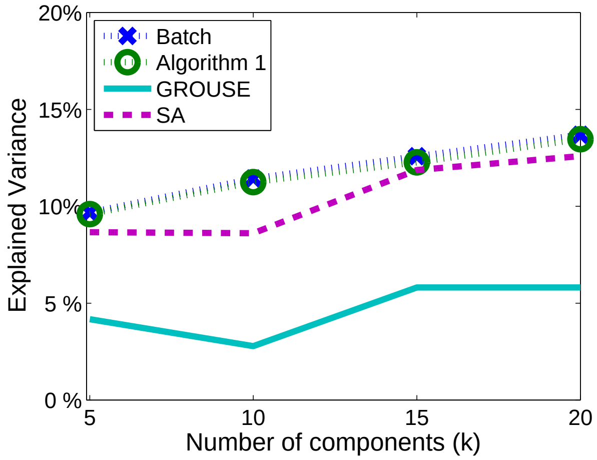

Figure 5: Performance on the MovieLens dataset

[4] L. Balzano and S. J. Wright. Local convergence of an algorithm for subspace identification from partial data. *arXiv preprint arXiv:1306.3391*, 2013.

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

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

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[8] J. He, L. Balzano, and J. Lui. Online robust subspace tracking from partial information. *arXiv preprint arXiv:1109.3827*, 2011.

[9] C.-J. Hsieh, K.-Y. Chiang, and I. S. Dhillon. Low rank modeling of signed networks. In *KDD*, pages 507–515, 2012.

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

[11] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(2057-2078):1, 2010.

[12] J. D. Lee, B. Recht, R. Salakhutdinov, N. Srebro, and J. A. Tropp. Practical large-scale optimization for max-norm regularization. In *NIPS*, pages 1297–1305, 2010.

[13] K. Lounici. High-dimensional covariance matrix estimation with missing observations. *arXiv preprint arXiv:1201.2577*, 2012.

[14] I. Mitliagkas, C. Caramanis, and P. Jain. Memory-limited, Streaming PCA. *arXiv preprint arXiv:1307.0032*, 2013.

[15] S. Negahban, M. J. Wainwright, et al. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*,

39(2):1069–1097, 2011.

[16] E. Oja and J. 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.

[17] J. D. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM, 2005.

[18] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[19] A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320–329, 2012.

[20] P. Å. Wedin. On angles between subspaces of a finite dimensional inner product space. In *Matrix Pencils*, pages 263–285. Springer, 1983.

APPENDIX

A. PRELIMINARIES

ASSUMPTION 10 (SUB-GAUSSIAN OBSERVATIONS [13]). *The random vector* $\mathbf{x} \in \mathbb{R}^p$ *is sub-gaussian, that is* $\|\mathbf{x}\|_{\psi_2} < \infty$. *In addition, there exists a numerical constant* $c_1 > 0$*, such that:* $\mathbb{E}(\langle \mathbf{x}, \mathbf{u}\rangle)^2 \geq c_1 \|\langle \mathbf{x}, \mathbf{u}\rangle\|^2_{\psi_2}$, $\forall \mathbf{u} \in \mathbb{R}^p$*, where* $\|\mathbf{x}\|_{\psi_2} = \inf\left\{u > 0 : \mathbb{E}\exp\left(|\mathbf{x}|^2/u^2\right) \leq 2\right\}$.

THEOREM 11 (PROP. 3, [13]). *Let* $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ *be i.i.d. random vectors satisfying Assumption 10. Let* $\mathbf{y}_1, \ldots, \mathbf{y}_n$ *be the corresponding observed vectors with* $\delta \in (0, 1]$. *Then, for any* $t > 0$*, we have with probability at least* $1 - e^{-t}$,

$$\|\widetilde{\Sigma}_n - \Sigma\|_2 \leq C\frac{\|\Sigma\|_2}{c_1} \cdot \max\left\{\sqrt{\frac{\mathbf{r}(\Sigma)\,(t + \log(2p))}{\delta^2 n}},\right.$$
$$\left.\frac{\mathbf{r}(\Sigma)\,(t + \log(2p))}{\delta^2 n}\,(c_1\delta + t + \log n)\right\},$$

where $C > 0$ *is an absolute constant and* $\mathbf{r}(\Sigma) = \frac{\mathrm{tr}(\Sigma)}{\|\Sigma\|_2}$.

THEOREM 12 (THEOREM 1.4 OF [18]). *Consider a finite sequence* X_k *of independent, random, self-adjoint matrices with dimension* d. *Assume that each random matrix satisfies* $\mathbb{E}[X_k] = 0$ *and* $\|X_k\|_2 \leq R$ *almost surely. Then, for all* $t \geq 0$,

$$Pr(\|\sum_k X_k\|_2 \geq t) \leq d \cdot \exp\left(\frac{-t^2/2}{\sigma^2 + Rt/3}\right),$$

where $\sigma^2 = \|\sum_k \mathbb{E}[X_k X_k^T]\|_2$.

B. PROOFS RANK-ONE CASE

We first state the following lemma that is repeatedly used throughout our proof.

LEMMA 13. *Let* $\mathbf{x}_t$ *be sampled from the generative model (10). Then, the following holds:*

$$\mathbb{E}[x_t^i] = 0, \qquad \mathbb{E}[x_t^i \cdot x_t^j] = u^i u^j + \sigma^2\mathbb{I}[i = j],$$
$$\mathbb{E}[(x_t^i)^2 \cdot (x_t^j)^2] = (u^i)^2 \cdot (u^j)^2 \cdot M_4 + \sigma^2(u^j)^2 + \sigma^2(u^i)^2$$
$$+ \sigma^4(\mathbb{I}[i \neq j] + M_4\mathbb{I}[i = j]) + 4u^i u^j \sigma^2\mathbb{I}[i = j],$$

where $M_4 = \mathbb{E}[z_t^4] = \mathbb{E}[(w_t^i)^4]$.

PROOF. Clearly, $\mathbb{E}[x_t^i] = u^i\mathbb{E}[z_t] + \mathbb{E}[w_t^i] = 0$. Next,

$$\mathbb{E}[x_t^i \cdot x_t^j] = u^i u^j \mathbb{E}[z_t^2] + \mathbb{E}[w_t^i w_t^j] = u^i u^j + \sigma^2\mathbb{I}[i = j].$$

Finally,

$$\mathbb{E}[(x_t^i)^2 \cdot (x_t^j)^2] = \mathbb{E}[(u^i \cdot u^i \cdot z_t \cdot z_t + w_t^i \cdot w_t^i + 2u^i z_t w_t^i)$$
$$(u^j \cdot u^j \cdot z_t \cdot z_t + w_t^j \cdot w_t^j + 2u^j z_t w_t^j)],$$
$$= (u^i)^2 \cdot (u^j)^2 \cdot M_4 + \sigma^2(u^j)^2 + \sigma^2(u^i)^2$$
$$+ \sigma^4(\mathbb{I}[i \neq j] + M_4\mathbb{I}[i = j]) + 4u^i u^j \sigma^2\mathbb{I}[i = j].$$

□

PROOF OF THEOREM 2. Let $\widetilde{\Sigma} - \Sigma = \widetilde{\Sigma}_N - \Sigma_N + \widetilde{\Sigma}_D - \Sigma_D$, where $\widetilde{\Sigma}_N - \Sigma_N$ is the non-diagonal part of $\widetilde{\Sigma} - \Sigma$, and $\widetilde{\Sigma}_D - \Sigma_D$ is the diagonal part of $\widetilde{\Sigma} - \Sigma$.

Then, using triangular inequality:

$$\|\widetilde{\Sigma} - \Sigma\|_2 \leq \|\widetilde{\Sigma}_N - \Sigma_N\|_2 + \|\widetilde{\Sigma}_D - \Sigma_D\|_2. \tag{13}$$

We now consider the non-diagonal part of the covariance matrices:

$$\widetilde{\Sigma}_N - \Sigma_N = \sum_t \sum_{i \neq j} \frac{1}{B}\left(\frac{\delta_t^i \delta_t^j}{\delta^2} x_t^i x_t^j - u^i u^j\right) e_i e_j^T = \sum_{t, i \neq j} H_{tij}.$$

$$\|H_{tij}\|_2 \leq \frac{4}{\delta^2 B}\left(\frac{\mu^2}{p} + \sigma^2\right) M_\infty^2. \tag{14}$$

$$\sum_{t, i \neq j} \mathbb{E}[H_{tij} H_{tij}^T] = \frac{1}{B^2} \sum_{t, i \neq j} \mathbb{E}\left[\frac{\delta_t^i \delta_t^j}{\delta^4}(x_t^i x_t^j)^2 - (u^i u^j)^2\right] e_i e_i^T,$$
$$= \frac{1}{B} \sum_{i \neq j} e_i e_i^T \left(\frac{1}{\delta^2}((u^i u^j)^2 \cdot M_4 + \sigma^2(u^j)^2\right.$$
$$\left. + \sigma^2(u^i)^2 + \sigma^4) - (u^i u^j)^2\right),$$
$$= D_N,$$

where, $D_N(i, i) \leq \frac{M_\infty^2}{\delta^2 B}\left(\frac{\mu^2}{p} + 2\sigma^2\mu^2 + \sigma^4 p\right)$.

That is,

$$\|\sum_{t, i \neq j} \mathbb{E}[H_{tij} H_{tij}^T]\|_2 \leq \frac{M_\infty^2}{\delta^2 B}\left(\frac{\mu^2}{p} + 2\sigma^2\mu^2 + \sigma^4 p\right). \tag{15}$$

Now, using matrix Bernstein's inequality (see Theorem 12) with (14), (15) and by setting B as given in Theorem 4, we get:

$$\|\widetilde{\Sigma}_N - \Sigma_N\|_2 \leq \epsilon/2. \tag{16}$$

Now, lets consider the diagonal part of Σ, i.e., :

$$\widetilde{\Sigma}_D(i, i) - \Sigma_D(i, i) = \frac{1}{B}\sum_t \left(\frac{\delta_t^i}{\delta}(x_t^i)^2 - (u^i)^2 - \sigma^2\right) = \sum_t h_t,$$

where $|h_t| \leq \frac{4}{\delta B}(\frac{\mu^2}{p} + \sigma^2) M_\infty^2$. Also,

$$\mathbb{E}[\sum_t h_t^2] = \frac{1}{B}\left(\frac{1}{\delta}\mathbb{E}[(x_t^i)^4] - ((u^i)^2 + \sigma^2)^2\right),$$
$$\leq \frac{6}{\delta^2 B}\left(\frac{\mu^4}{p^2} M_4 + \sigma^2\frac{\mu^2}{p} + \sigma^4 M_4\right) \leq \frac{6M_\infty^2}{\delta^2 B}\left(\frac{\mu^2}{p} + \sigma^2\right)^2.$$

Again, by using Bernstein's inequality and setting B as defined in Theorem 4, we have:

$$\|\widetilde{\Sigma}_D - \Sigma_D\|_2 \leq \epsilon/2. \tag{17}$$

Theorem now follows by using (13), (16), (17). □

Proof of Lemma 5. Note that,

$$|u^T(\widetilde{\Sigma} - \Sigma)q| \le |u^T(\widetilde{\Sigma}_N - \Sigma_N)q| + |u^T(\widetilde{\Sigma}_D - \Sigma_D)q|, \quad (18)$$

where, $\widetilde{\Sigma}_N - \Sigma_N$ is the non-diagonal part of $\widetilde{\Sigma} - \Sigma$, and $\widetilde{\Sigma}_D - \Sigma_D$ is the diagonal part of $\widetilde{\Sigma} - \Sigma$.

We first bound the non-diagonal part:

$$u^T\left(\widetilde{\Sigma}_N - \Sigma_N\right)q = \frac{1}{n}\sum_{t,i\neq j}\left(\frac{\delta_t^i\delta_t^j}{\delta^2}x_t^i x_t^j - u^i u^j\right)u^i q^j = \sum_{t,i\neq j} H_{tij}.$$

Note that, $|H_{tij}| \le \frac{1600}{\delta^2 B}(\frac{\mu^2}{p} + \sigma^2)\frac{\mu^2}{p}M_\infty^2$. Also,

$$\begin{aligned}\mathbb{E}[\sum_{t,i\neq j} H_{tij}^2] = \frac{1}{B}\sum_{i\neq j}\Bigg(\frac{1}{\delta^2}\Big((u^i)^2\cdot(u^j)^2\cdot M_4 + \sigma^2(u^j)^2 \\ +\sigma^2(u^i)^2 + \sigma^4\Big) - (u^i u^j)^2\Bigg)(u^i)^2(q^j)^2, \\ \le \frac{M_\infty^2}{\delta^2 B}\left(\frac{\mu^4}{p^2}M_4 + 2\sigma^2\frac{\mu^2}{p} + \sigma^4\right).\end{aligned}$$

Hence, using Bernstein's inequality, w.p. $\ge 1 - 1/T^2$:

$$|u^T(\widetilde{\Sigma}_N - \Sigma_N)q| \le \frac{\epsilon}{2\sqrt{p}}. \quad (19)$$

Now, consider the diagonal part:

$$u^T\left(\widetilde{\Sigma}_D - \Sigma_D\right)q = \frac{1}{B}\sum_{t,i}\left(\frac{\delta_t^i}{\delta}(x_t^i)^2 - (u^i)^2 - \sigma^2\right)u^i q^i = \sum_{ti} h_{ti}.$$

Here again, we bound the two quantities:

$$\begin{aligned}|h_{ti}| &\le \frac{2}{B}(\frac{\mu^2}{p} + \sigma^2)\frac{\mu^2}{p}M_\infty^2, \\ \mathbb{E}[h_{ti}^2] &= \frac{1}{B}\Bigg(\sum_i \frac{1}{\delta}\cdot((u^i)^4 M_4 + 2\sigma^2(u^i)^2 + \sigma^4 M_4 \\ &\quad +4(u^i)^2\sigma^2) - ((u^i)^2 + \sigma^2)^2)(u^i q^i)^2\Bigg) \\ &\le \frac{M_\infty^2}{\delta B}\frac{\mu^2}{p}(\frac{\mu^4}{p^2} + 6\sigma^2\frac{\mu^2}{p} + \sigma^4).\end{aligned}$$

Hence, setting B appropriately, we get (w.p. $\ge 1 - 2/T^2$):

$$|u^T\left(\widetilde{\Sigma} - \Sigma\right)q| \le \epsilon\frac{1}{2\sqrt{p}}. \quad (20)$$

Lemma now follows by using (18), (19), (20). □

Proof of Lemma 6. We prove the lemma using mathematical induction. That is, we assume that $\mu_\tau \le 10\mu$.

Recall that $s_{\tau+1} = \widetilde{\Sigma}q_\tau$ and $q_{\tau+1} = s_{\tau+1}/\|s_{\tau+1}\|_2$.

Using Theorem 2, we have $\|s_{\tau+1}\|_2 \ge (1+\sigma^2) - \epsilon$. Also,

$$e_i^T(s - \Sigma q_\tau) = \frac{1}{B}\sum_{t,j\neq i} h_{tj} + \frac{1}{B}\sum_t h_t,$$

where $h_{tj} = \frac{1}{B}(\frac{\delta_t^i\delta_t^j}{\delta^2}x_t^i x_t^j - u^i u^j)q_\tau^j$ and $h_t = (\frac{\delta_t^i}{\delta}x_t^i x_t^i - u^i u^i - \sigma^2)q_\tau^i$. Note that, $|h_{tj}| \le \frac{1}{\delta^2 B}(\frac{\mu^2}{p} + \sigma^2)\frac{\mu_\tau}{\sqrt{p}}M_\infty^2$. Using arguments similar to the proof of the previous lemma:

$$\mathbb{E}[\sum_{tj} h_{tj}^2] \le \frac{M_\infty^2}{\delta^2 B}\left(\frac{\mu^4}{p^2} + 2\sigma^2\frac{\mu^2}{p} + \sigma^4\right).$$

Hence, using induction hypothesis and by using B as given in Theorem 4, we have (w.p. $\ge 1 - 1/T^2$):

$$|\frac{1}{B}\sum_{t,j\neq i}(\frac{\delta_t^i\delta_t^j}{\delta^2}x_t^i x_t^j - u^i u^j)q_\tau^j| \le \epsilon/\sqrt{p}$$

We can bound $\sum_t h_t$ also in a similar manner.

Hence, $\frac{s_{\tau+1}^i}{\sqrt{p}\|s_{\tau+1}\|_2} \le \frac{\mu_\tau(1+\sigma^2)+\epsilon}{(1+\sigma^2)-\epsilon} \le (\mu_\tau + \frac{\epsilon}{(1+\sigma^2)} + 2\mu_\tau\frac{\epsilon}{1+\sigma^2} + 2\frac{\epsilon^2}{(1+\sigma^2)^2} \le \mu_\tau(1 + 4\epsilon/(1+\sigma^2))$.

Moreover, since $\max_i |q_0^i| \le 5/\sqrt{p}$ with at least a constant probability, and,

$$\tau + 1 \le T = O(\frac{\log(1/\epsilon)}{\log(\sigma^2 + .75)/(\sigma^2 + .5)}),$$

we have $\mu_{\tau+1} \le 10\cdot\mu$. □

Proof of Theorem 4. As mentioned earlier, our proof uses arguments similar to that of Theorem 1 of [14], along with the above mentioned lemmas.

That is, let $q_\tau = \sqrt{1-\delta_\tau}u + \sqrt{\delta_\tau}u_\perp^\tau$ where $u_\perp$ is the component of q that is orthogonal to u. Now, using Lemma 5, as well as the fact that with high probability $u^T q_0 \ge \frac{1}{\sqrt{p}}$, we have w.p. at least 0.99:

$$u^T s_{\tau+1} = u^T\Sigma q_\tau \ge \sqrt{1-\delta_\tau}(1+\sigma^2)\left(1 - \frac{\epsilon}{4(1+\sigma^2)}\right).$$

Similarly, using Theorem 2, we have: $(u_\perp^\tau)^T s_{\tau+1} \le \sigma^2\sqrt{\delta_\tau} + \epsilon$. Also, $\delta_{\tau+1} = \frac{((u_\perp^\tau)^T s_{\tau+1})^2}{(u_\perp^\tau)^T s_{\tau+1} + (u^T s_{\tau+1})^2}$. Now, using the bounds on both the quantities on RHS, we have:

$$\begin{aligned}\sqrt{1 - (u^T q_{\tau+1})^2} &= \delta_{\tau+1} \\ &\le \frac{\delta_\tau(\sigma^2 + .5)^2}{(1-\delta_\tau)(\sigma^2 + .75)^2 + \delta_\tau(\sigma^2 + .5)^2} \le \frac{\gamma^{2\tau}\delta_0}{1-\delta_0},\end{aligned}$$

where $\gamma = \frac{\sigma^2+.5}{\sigma^2+.75}$. Above, the second inequality follows from Lemma 2 of [14]. Now, using the fact that $u^T q_0 \ge \frac{1}{\sqrt{p}}$ (w.h.p.), we have:

$$\sqrt{1 - (u^T q_{\tau+1})^2} \le C\left(\frac{\sigma^2 + .5}{\sigma^2 + .75}\right)^{2\tau} p.$$

Theorem now follows by setting $\tau + 1 = T$. □