Streaming Principal Component Analysis in Noisy Settings

Teodor V. Marinov [*1] **Poorya Mianjy** [*1] **Raman Arora** [1]

Abstract

We study streaming algorithms for principal component analysis (PCA) in noisy settings. We present computationally efficient algorithms with sub-linear regret bounds for PCA in the presence of noise, missing data, and gross outliers.

1. Introduction

Principal component analysis (PCA) is a ubiquitous technique in statistics, machine learning and data science. Given a dataset $\{\mathrm{x}_i\}_{i=1}^n \subset \mathbb{R}^d$, PCA finds a low dimensional subspace of $\mathbb{R}^d$ which captures maximum variance in the dataset. PCA is often performed as a pre-processing step for dimensionality reduction to help reduce computational and statistical burden for a downstream learning task, especially in the context of big data.

PCA can be posed as a stochastic optimization problem, where the samples are assumed to be drawn i.i.d. from an unknown distribution $\mathcal{D}$ and the goal is to find a subspace that is almost as good as the optimal subspace in terms of capturing the variance in the distribution; such a view motivates design of stochastic approximation (SA) algorithms that process one sample at a time with a computationally cheap update and can scale to large datasets (Arora et al., 2012; Balsubramani et al., 2013; Jain et al., 2016; Shamir, 2016; Allen-Zhu & Li, 2017a; Mianjy & Arora, 2018).

In this paper, we study PCA in a streaming setting. While our algorithms are motivated by previous work on stochastic approximation algorithms for PCA, in our analysis, we also consider a non-stochastic setting where we make no distributional assumptions on data. In particular, the data may have been generated deterministically or even adversarially. Such a setting makes sense for *big data* applications where the data needs to be processed as it streams in, and it is unreasonable to assume that successive samples are independent or even identically distributed.

[*]Equal contribution [1]Department of Computer Science, Johns Hopkins University, Baltimore, USA. Correspondence to: Teodor V. Marinov <tmarino2@cs.jhu.edu>.

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

Big data is characterized not only by its sheer "*volume*" but also by its "*veracity*", or the lack thereof. Most big data analysts do not trust the raw data due to large corruptions and deletions. It is therefore crucial to design algorithms for PCA that can handle extremely noisy data as well as missing data and also scale to very large datasets.

Instead of studying PCA as a stochastic optimization problem, one can consider *online* PCA. However, the focus of online analysis, and in particular of past work on online PCA (Tsuda et al., 2005; Warmuth & Kuzmin, 2006; 2008; Nie et al., 2013), is on bounding the adversarial regret, rather than on the runtime. A "good" online method might therefore have low online regret, and thus a low iteration complexity as a stochastic procedure, but its expensive runtime-per-iteration might make it unappealing for stochastic approximation. Some recent work by Garber et al. (2015); Allen-Zhu & Li (2017b) on online PCA has focussed on computationally efficient updates. Here, we are concerned both with designing robust streaming algorithms for PCA with arbitrary corruptions void of any distributional assumptions as well as obtaining methods with low overall runtime.

Therefore, in this paper, we consider both the nonstochastic and stochastic settings – for the former we give variants of online mirror descent with sublinear regret guarantees on the PCA objective, for the latter we give extensions of the computationally efficient Oja's algorithm. We study PCA in a streaming setting with noisy gradients, missing data, partial observations, and gross outliers.

To the best of our knowledge, PCA with corrupted gradients, where corruption can be in the form of noise, missing entries or outliers has not been studied in the online setting. Our first contribution is to provide variants of online mirror descent which obtain optimal regret guarantees in terms of time horizon T and scale gracefully with the amount of corruption to the data.

In the stochastic setting, there have been multiple works dealing with subspace recovery when data is missing under some model (Balzano et al., 2010; Lounici et al., 2014; Mitliagkas et al., 2014). Most of these works, either have very stringent requirements on the distribution of the data or prove only local convergence results. Our second main contribution is in extending Oja's algorithm, when data is assumed to be stochastic, to the setting of corrupted gra-

dients by nonstochastic noise, missing entries and partial observations.

The rest of the paper is organized as follows. In Section 3 we consider streaming PCA in presence of bounded noise, i.e. when the observations are corrupted. In Section 4 we give an algorithm for streaming PCA which is robust to missing data, when the entries are missing at random from a Bernoulli model. We then change the focus in Section 5 to the problem of streaming PCA with partial observations, where only a few entries per sample are observed due to the cost of obtaining measurements. We propose a robust streaming PCA algorithm that can handle outliers in Section 6. Finally, we give experimental evidence for proposed methods in Section 7 and conclude with a discussion in Section 8.

2. Notation and Preliminaries

We denote vectors and matrices with small and capital Roman letters, e.g. u and U. The identity matrix of size k is represented by I_k, where the subscript k is dropped whenever the size is clear from the context. The ℓ_2 norm of a vector is denoted by $\|\cdot\|$. Frobenius norm and spectral norm of matrices are denoted by $\|\cdot\|_F$ and $\|\cdot\|_2$ respepctively. For any two matrices $\mathrm{M}_1, \mathrm{M}_2 \in \mathbb{R}^{d\times d}$, the standard inner-product is given as $\langle \mathrm{M}_1, \mathrm{M}_2\rangle = \mathrm{Tr}\left(\mathrm{M}_1^\top \mathrm{M}_2\right)$. Furthermore, $\mathcal{P}_k = \{\mathrm{P} :\ \mathrm{P}^2 = \mathrm{P}, \mathrm{P} = \mathrm{P}^\top, \mathrm{rank}(\mathrm{P}) = k\}$ denotes the set of rank-k orthogonal projection matrices.

Online Setting. The k-dimensional PCA problem in the streaming setting can be formulated as follows. We are presented with a sequence of vectors $(\mathrm{x}_n)_{n=1}^\infty$ in $\mathbb{R}^d$. After observing $\mathrm{x}_1, \ldots, \mathrm{x}_{t-1}$, we need to predict a k-dimensional subspace, represented by a rank-k orthogonal projection matrix $\mathrm{P}^{(t)}$, so as to minimize the residual $\|\mathrm{x}_t - \mathrm{P}^{(t)}\mathrm{x}_t\|^2$ of the next vector in the stream.

We are interested in bounding the adversarial regret, i.e. obtaining an upper bound for $\sum_{t=1}^T \|\mathrm{x}_t - \mathrm{P}^{(t)}\mathrm{x}_t\|^2 - \sum_{t=1}^T \|\mathrm{x}_t - \mathrm{P}\mathrm{x}_t\|^2$, that holds for *any* sequence $\mathrm{x}_1, \ldots, \mathrm{x}_T \in \mathbb{R}^d$, and *any* competitor $\mathrm{P} \in \mathcal{P}_k$, where $\mathrm{P}^{(t)}$ is the sequence of projection matrices generated by an online algorithm. Minimizing the regret above defined in terms of the residual errors is equivalent to minimizing the regret defined in terms of the variance captured. Therefore, the k-dimensional PCA problem in the streaming setting can be formulated as finding a sequence of subspaces, represented by orthogonal projection matrices, $\mathrm{P}^{(t)}$, that minimizes

$$R(T, \mathrm{P}) = \sum_{t=1}^T \mathrm{x}_t^\top \mathrm{P}\mathrm{x}_t - \sum_{t=1}^T \mathrm{x}_t^\top \mathrm{P}^{(t)}\mathrm{x}_t, \tag{1}$$

where $\mathrm{P} \in \mathcal{P}_k$ is an arbitrary rank-k orthogonal projection matrix. A sublinear regret bound implies that we can drive the *average regret*, i.e. $\frac{1}{T}R(T, \mathrm{P})$, below any user-specified $\epsilon > 0$. This allows us to measure the performance of an online algorithm in terms of overall runtime required to achieve ϵ-average regret.

In the online setting, we consider algorithms that are variants of online mirror descent, a standard algorithm in Online Convex Optimization literature (Beck & Teboulle, 2003). However, since the feasible set $\mathcal{P}_k$ is not convex, we relax the feasible set by taking its convex hull,

$$\mathcal{C} = \{\mathrm{P} :\ \mathrm{Tr}\,(\mathrm{P}) := k,\ 0 \preceq \mathrm{P} \preceq \mathrm{I},\ \mathrm{P} = \mathrm{P}^\top\}. \tag{2}$$

Therefore, our updates are of the following form:

$$\mathrm{P}^{(t+1)} = \Pi_F(\mathrm{P}^{(t)} + \eta \mathrm{g}_t^\top), \tag{3}$$

where η is the learning rate, g_t is the gradient estimate at time t, and Π_F is the projection operator onto the set $\mathcal{C}$ with respect to Frobenius norm. The projection step is a simple shift-and-cap procedure described in (Arora et al., 2013).

Since each iterate $\mathrm{P}^{(t)} \in \mathcal{C}$ can have rank larger than k, we sample a rank-k projection matrix using the rounding procedure described in Algorithm 2 of (Warmuth & Kuzmin, 2008); we denote $\widehat{\mathrm{P}}^{(t)} = \texttt{rounding}(\mathrm{P}^{(t)})$. Since, the loss function in (1) is linear, is easy to check that the sequence $\widehat{\mathrm{P}}^{(t)}$ has the same adversarial regret in expectation w.r.t. the rounding. We refer to these updates as **matrix gradient descent (MGD)**. The following regret bound holds for MGD.

Theorem 2.1 (MGD regret)**.** Assume $\|\mathrm{x}_t\| \le 1$ for all t in $1, \ldots, T$. Then, after T iterations of MGD with step size $\eta = \sqrt{\frac{k}{T}}$, and starting at $\mathrm{P}^{(1)} = 0$, we have that

$$\mathbb{E}[R(T, \mathrm{P})] \le \sqrt{kT}. \tag{4}$$

where expectation is w.r.t. randomization in the algorithm, and $\mathrm{P} \in \mathcal{P}_k$ is any arbitrary rank-k projection matrix.

Stochastic Setting. The regret bound in Theorem 2.1 is minimax optimal with respect to the time horizon T and dimension d (Nie et al., 2013). The question of computational efficiency, however, still remains. In particular, the per iteration complexity of MGD can grow as large as $\Omega(d^3)$. This is not desirable or even computationally tractable in a big data setting, where the dimensionality of the data can be very large. To the best of our knowledge, there are no known algorithms in the regret minimization setting which can be computationally better in the worst case and still achieve optimal regret. This gives little hope that we can come up with computationally attractive algorithms for the online problem. Under the additional assumption that x_t are sampled i.i.d. from some unknown distribution $\mathcal{D}$ and that $\|\mathrm{x}_t\| \le 1$ almost surely, recent work (Allen-Zhu & Li, 2017b) has shown that Oja's algorithm can obtain sub-linear

regret for the online PCA problem in the special case of $k = 1$. At each iteration, Oja's algorithm, for general k, performs the following updates:

$$\tilde{\mathrm{U}}_{t+1} = \mathcal{P}\left(\left(\mathrm{I} + \eta \mathrm{x}_t \mathrm{x}_t^\top\right) \cdots \left(\mathrm{I} + \eta \mathrm{x}_1 \mathrm{x}_1^\top\right) \mathrm{U}\right)$$
$$\mathrm{P}^{(t+1)} = \tilde{\mathrm{U}}_{t+1} \tilde{\mathrm{U}}_{t+1}^\top,$$

where entries of $\mathrm{U} \in \mathbb{R}^{d \times k}$ are sampled from a standard Gaussian distribution and $\mathcal{P}(\mathrm{A})$ orthonormalizes the columns of A. Note that computing $\tilde{\mathrm{U}}_{t+1}$ takes $O(dk^2)$ time, and we never need to form $\mathrm{P}^{(t+1)}$ explicitly, so the per-iteration computational cost of Oja's algorithm is $O(dk^2)$.

3. Streaming PCA with corrupted gradients

Online Setting. We consider a setting where the streaming algorithm receives noisy gradients, i.e. instead of instantaneous gradients $\mathrm{x}_t \mathrm{x}_t^\top$, it receives the sequence $\widehat{\mathrm{g}}_t = \mathrm{x}_t \mathrm{x}_t^\top + \mathrm{E}_t$. The noise could be a result of an inaccurate computation in a big data setting, or a consequence of asynchronous and noisy communication in a distributed or parallel computing scenario. If the noise in the gradient is unbounded, then no learning is possible. Our first main result is in the case of bounded noise and shows that the regret bound for MGD degrades gracefully with the noise level. Furthermore, MGD can easily tolerate noise with overall budget that scales as $o(T)$ if we desire sublinear regret guarantees.

Theorem 3.1 (MGD noisy gradient)**.** Assume $\|\mathrm{x}_t\| \leq 1$ for all t in $1, \ldots, T$. Let $\mathrm{E}_1, \ldots, \mathrm{E}_T$ be an arbitrary sequence of error in gradients such that $\sum_{t=1}^T \|\mathrm{E}_t\|_2 \leq E$ and $\|\mathrm{E}_t\|_F \leq 1$ for all $t = 1, \cdots, T$. Then, after T iterations of MGD with step size $\eta = \sqrt{k/T}$, and starting at $\mathrm{P}^{(1)} = 0$, we have that

$$\mathbb{E}[R(T, \mathrm{P})] \leq 4\sqrt{kT} + 2kE, \tag{5}$$

where expectation is w.r.t. randomization in the algorithm and $\mathrm{P} \in \mathcal{P}_k$ is any arbitrary rank-k projection matrix.

Stochastic Setting. We consider the same stochastic setting as Allen-Zhu & Li (2017b) and further assume that $\mathbb{E}_{\mathcal{D}}[\mathrm{x}_t] = 0$. As before, we consider corrupted gradients, however, we assume that they arise due to additive corruption of data, i.e. each of the points x_t are perturbed by some noise vector y_t. The noisy gradients we observe are $\widehat{\mathrm{g}}_t = (\mathrm{x}_t + \mathrm{y}_t)(\mathrm{x}_t + \mathrm{y}_t)^\top$. We assume y_t is independent of x_t but make no other stochastic assumption on y_t. We do require that the total noise is bounded. We make this explicit in the following theorem.

Theorem 3.2 (Oja noisy gradient)**.** Assume that $\mathrm{x}_t \sim \mathcal{D}$, $\|\mathrm{x}_t\| \leq 1$ almost surely for all $t = 1, \ldots, T$ and $\mathbb{E}_{\mathcal{D}}[\mathrm{x}_t] = 0$. Assume that $\sum_{t=1}^T \|\mathrm{y}_t\| + \|\mathrm{y}_t\|^2 \leq \sqrt{T}$. After T iterations of Oja's algorithm starting with $\mathrm{u} \sim \mathcal{N}(0, \mathrm{I})$ and using step size $\eta = \frac{\beta}{\sqrt{T}}$ for some small enough β, with probability $1 - \delta$ it holds that:

$$R(T, \mathrm{P}^*) \leq c\sqrt{T}\frac{\log(d + \log(2/\delta))}{\beta} + \sqrt{T}\frac{\log\left(\frac{8}{3\delta^2}\right)}{\beta},$$

where c is some universal constant not depending on δ or d and $\mathrm{P}^* = \arg\max_{\mathrm{P} \in \mathcal{P}_1} \sum_{t=1}^T \langle \mathrm{g}_t, \mathrm{P} \rangle$.

We remark that β has no dependence on d or T, however, it is at most $O(\log(1+\delta))$. Since the gradients we are working with, $\widehat{\mathrm{g}}_t$, are not unbiased and not bounded by a constant at each iteration necessarily, the result provided in (Allen-Zhu & Li, 2017b) does not apply directly. We adapt their analysis, by decomposing the instantaneous gradient $\widehat{\mathrm{g}}_t = (\mathrm{x}_t + \mathrm{y}_t)(\mathrm{x}_t + \mathrm{y}_t)^\top$ into an unbiased term $\mathrm{g}_t = \mathrm{x}_t \mathrm{x}_t^\top$ and an error term $\widehat{\mathrm{g}}_t - \mathrm{g}_t$. The assumption that the noise is sublinear allows us to control this error term and still achieve a sublinear regret bound.

4. Learning with Missing Entries

Often in applications with large volumes of data streaming in, malfunction of the measurement or data acquisition systems results in data corruption or large gaps in the collection. Hence, it is crucial to design algorithms that can reliably handle missing data. In this section, we introduce a simple randomized scheme to extend the streaming MGD algorithm to handle missing data. The key insight here is that with the proposed randomized scheme we can still obtain unbiased estimates of the instantaneous gradient based on missing data.

The problem of PCA with missing data has been studied before in the stochastic setting (Balzano et al., 2010; Mitliagkas et al., 2014). The setting we consider here is closely related to that of Mitliagkas et al. (2014). In particular, both the missing-ness model as well as Oja's updates that we consider here are as in (Mitliagkas et al., 2014). However, there are two key differences. First, we give sublinear regret bounds in a non-stochastic setting. Second, in the stochastic setting, we make less stringent assumptions; while Mitliagkas et al. (2014) assume that the data is generated using a spiked-covariance model, we only need to assume that the distribution of data has bounded support.

In the Bernoulli model of missing-ness (Candès et al., 2011) that we consider here, for each data point x_t, we sample d Bernoulli random variables $Z_i \sim \text{Bernoulli}(q)$, $i = 1, \ldots, d$, to get the index set of observed entries $\Omega := \{i \in [d] : Z_i = 1\}$. The observed vector $\tilde{\mathrm{x}}_t$ is then given by

$$\tilde{x}_i = \begin{cases} x_i, & i \in \Omega \\ 0, & \text{otherwise} \end{cases}.$$

Lets denote the distribution of the observed vector conditioned on x by $\mathcal{R}$. Then, it is easy to check that $\mathbb{E}_{\mathcal{R}}[\|\tilde{\mathrm{x}}\|_0|\mathrm{x}] = qd$, i.e, on average qd elements of each vector are observed under the model $\mathcal{R}$. It is known that $\tilde{\mathrm{x}}\tilde{\mathrm{x}}^\top$ is not an unbiased estimator of $\mathrm{x}\mathrm{x}^\top$ with respect to $\mathcal{R}$ (see (Lounici et al., 2014) or Lemma **??** in the Appendix). To address this issue, we propose the following random model for constructing an unbiased estimator of $\mathrm{x}\mathrm{x}^\top$. Assume r entries of x are observed, and let $\Omega = \{i_1, \cdots, i_r\}$ be the indices of observed elements. We construct $\widehat{\mathrm{g}} := \widehat{\mathrm{x}}\widehat{\mathrm{x}}^\top - \mathrm{z}\mathrm{z}^\top$ where $\widehat{\mathrm{x}} = \frac{1}{q}\tilde{\mathrm{x}}$ and $\mathrm{z} = \frac{\sqrt{r-rq}}{q} x_{i_s} \mathrm{e}_{i_s}$ and i_s is sampled uniformly at random from Ω. Let $\mathcal{S}$ denote the conditional distribution of z given x. The following holds.

Lemma 4.1. $\widehat{\mathrm{g}}$ is an unbiased estimator of $\mathrm{x}\mathrm{x}^\top$, that is:

$$\mathbb{E}_{\mathcal{S},\mathcal{R}}[\widehat{\mathrm{x}}\widehat{\mathrm{x}}^\top - \mathrm{z}\mathrm{z}^\top|\mathrm{x}] = \mathrm{x}\mathrm{x}^\top, \tag{6}$$

where the expectation is with respect to both $\mathcal{S}$ and $\mathcal{R}$.

Online Setting. Lemma 4.1 motivates the following r-MGD updates with missing entries based on $\widehat{\mathrm{g}}_t := -\widehat{\mathrm{x}}_t\widehat{\mathrm{x}}_t^\top + \mathrm{z}_t\mathrm{z}_t^\top$:

$$\mathrm{P}^{(t+1)} = \Pi_F\left(\mathrm{P}^{(t)} - \eta_t \widehat{\mathrm{g}}_t\right). \tag{7}$$

MGD enjoys the following regret bound.

Theorem 4.2 (MGD regret with missing data)**.** Assume $\|\mathrm{x}_t\| \leq 1$ for all t in $1, \ldots, T$. Then, after T iterations of the MGD update in (7), with step size $\eta = \frac{q^2}{\sqrt{q^2+dq(1-q)^3+d^2q^2(1-q)^2}}\sqrt{\frac{k}{T}}$, and starting at $\mathrm{P}^{(1)} = 0$, we have that:

$$\mathbb{E}\left[R(T, \mathrm{P})\right] \leq C_{r,d}\sqrt{kT}, \tag{8}$$

for any rank-k projection matrix P. Here, $C_{r,d} = \frac{\sqrt{q^2+dq(1-q)^3+d^2q^2(1-q)^2}}{q^2}$, and the expectation is w.r.t. randomization in the algorithm.

Note that the regret bound in Theorem 4.2 degrades gracefully with the parameter q. As $q \to 1$, $C_{r,d} \to 1$ and we recover the bound in Theorem 2.1. On the other hand, as q becomes smaller, the tradeoff parameter $C_{r,d}$ grows and for $q = O(1/d)$, we get $C_{r,d} = O(d^2)$.

Stochastic Setting. As discussed in Section 2, MGD can be inefficient. Again, we consider the stochastic setting where x_t are sampled i.i.d. from some distribution $\mathcal{D}$ and $\|\mathrm{x}_t\| \leq 1$ almost surely. We consider Oja's algorithm for $k = 1$ with gradients $\widehat{\mathrm{g}}_t$. However, as the gradients are not guaranteed to be positive-semidefinite, the results in (Allen-Zhu & Li, 2017b) do not apply directly. We are able to adapt the proof techniques by decomposing the gradient into a positive semidefinite part and a negative semidefinite part. It turns out that the negative semidefinite part can only hurt us in terms of increasing the norm of the gradient, however, this only leads to a constant factor in the regret bound.

Theorem 4.3 (Oja regret from missing data)**.** Assume that $\mathrm{x}_t \sim \mathcal{D}$ for all t in $1, \ldots, T$ and that $\|\mathrm{x}_t\| \leq 1$ almost surely. Let $\mathrm{C} = \mathbb{E}_{\mathrm{x}\sim\mathcal{D}}[\mathrm{x}\mathrm{x}^\top]$ with top eigenvalue λ, let $\alpha_n = \frac{2^{n-1}}{q^n} + \frac{2^{n-1}\mu_n(1-q)^n}{q^{2n}}$, where μ_n is the n-th moment of the Binomial distribution, and let $\alpha = \alpha_4 + 4\alpha_3 + 6\alpha_2$. Then, after T iterations of Oja's algorithm with gradients $\widehat{\mathrm{g}}_t$, initialization $\mathrm{P}^{(1)} = \mathrm{u}\mathrm{u}^\top$, where $\mathrm{u} \sim \mathcal{N}(0, \mathrm{I})$ and step size $\eta = \frac{\log(1+\delta/9)}{(\alpha+4\lambda^2)\sqrt{T}}$, with probability $1-\delta$ it holds that:

$$\mathbb{E}_{\mathcal{S},\mathcal{R}}[R(T, \mathrm{P})] \leq \frac{\sqrt{T}}{\log(1+\delta/18)} + \frac{\sqrt{T}(\alpha+4\lambda^2)O\left(\log(d + \log\left(\frac{1}{\delta}\right)) - \log\left(\frac{1}{\delta}\right)\right)}{\log(1+\delta)},$$

for any rank-1 projection matrix P.

As $q \to 1$ we see that the regret tends to $O\left(\frac{\sqrt{T}\log(d+\log(1/\delta))}{\log(1+\delta)}\right)$. For any fixed δ, the regret above equals $O(\sqrt{T}\log(d))$, which has an additional multiplicative factor of $\log(d)$ compared to the bound in Theorem 4.2. However, this does not take the per-iteration computational cost of both algorithms into account. To better understand the trade-off between Oja's algorithm with missing entries and MGD with missing entries we look at the overall runtime needed to achieve ϵ-average regret. The total runtime of MGD for achieving ϵ-average regret is $O(\frac{d^3}{\epsilon^2})$. On the other hand the per-iteration complexity of Oja's algorithm is $O(d)$ and thus the total run-time for achieving ϵ-average regret is $O(\frac{d\log^2(d)}{\epsilon^2})$. Therefore, we see that Oja's algorithm has much better overall performance in terms of runtime when considering average regret as $q \to 1$. The case is a bit different as we let $q \to \frac{1}{d}$. This suggests that $\alpha = O(d^8)$ and the regret bound for Oja's algorithm is $O(d^8\sqrt{T}\log(d))$, while the regret bound for MGD is $O(d^2\sqrt{T})$. Thus, Oja's algorithm with missing entries becomes intractable in such a setting. However, we note that the regret bound for Oja's algorithm with missing entries might not be minimax optimal in terms of the dependence on dimensionality d and in practice we have not observed such discrepancy in our experiments.

5. Learning from Partial Observations

In many real world applications, acquiring a full feature vector may be challenging or there may be cost associated with fetching each entry of the vector. Consequently, in such settings, it is essential that data analysis solutions are designed to work reliably and robustly with partially observed data. In this section, we introduce a randomized scheme that ensures obtaining unbiased estimates of the gradient based on partial observations. This allows a simple extension of the streaming MGD algorithm to handle partial observations.

We consider the uniform sampling model that has been used extensively in the literature (see, e.g. (Recht, 2011)). In particular, we observe r entries uniformly at random from all subsets of cardinality r. At each iterate, we sample an indexing subset $\Omega := \{i_1, .., i_r\} \subseteq [1 \cdots d]$ with cardinality $0 < r \leq d$ uniformly at random from all subsets of size r. The observed vector $\tilde{\mathrm{x}}$ is now constructed as

$$\tilde{x}_i = \begin{cases} x_i, & i \in \Omega \\ 0, & \text{otherwise.} \end{cases}$$

As in the previous section, let $\mathcal{R}$ denote the conditional distribution of the observed vector. Again, we observe that $\tilde{\mathrm{x}}\tilde{\mathrm{x}}^\top$ is not an unbiased estimator of $\mathrm{x}\mathrm{x}^\top$ with respect to $\mathcal{R}$ (see Lemma **??** in the Appendix). To construct an unbiased estimator of $\mathrm{x}\mathrm{x}^\top$, we propose the following stochastic model. We define $\widehat{\mathrm{g}} := \widehat{\mathrm{x}}\widehat{\mathrm{x}}^\top - \mathrm{z}\mathrm{z}^\top$ where $\widehat{\mathrm{x}} = \sqrt{\frac{d(d-1)}{r(r-1)}}\tilde{\mathrm{x}}$, and $\mathrm{z} = \sqrt{\frac{dr-r^2}{r-1}}\tilde{x}_{i_s}\mathrm{e}_{i_s}$ and e_i is the i-th standard basis vector and i_s is sampled uniformly at random from the set of observed elements Ω. Let $\mathcal{S}$ be the conditional distribution induced by this model. Then, the following holds.

Lemma 5.1. $\widehat{\mathrm{g}}$ is an unbiased estimator of $\mathrm{x}\mathrm{x}^\top$, i.e.

$$\mathbb{E}_{\mathcal{S},\mathcal{R}}[\widehat{\mathrm{x}}\widehat{\mathrm{x}}^\top - \mathrm{z}\mathrm{z}^\top | \mathrm{x}] = \mathrm{x}\mathrm{x}^\top, \tag{9}$$

where the expectation is with respect to both $\mathcal{S}$ and $\mathcal{R}$.

Online Setting. Lemma 5.1 motivates the following MGD updates with partial observations based on $\widehat{\mathrm{g}}_t := -\widehat{\mathrm{x}}_t\widehat{\mathrm{x}}_t^\top + \mathrm{z}_t\mathrm{z}_t^\top$:

$$\mathrm{P}^{(t+1)} = \Pi_F\left(\mathrm{P}^{(t)} - \eta\widehat{\mathrm{g}}_t\right). \tag{10}$$

We show that the following regret bound holds for MGD with partial observations.

Theorem 5.2 (MGD regret from partial observations)**.** Assume $\|\mathrm{x}\|_t \leq 1$ for all t in $1, \ldots, T$. Then, after T iterations of the MGD update in (10), with step size $\eta = \frac{r(r-1)}{\sqrt{d^2(d-1)^2 + r^4(d-r)^2}}\sqrt{\frac{k}{T}}$, and starting at $\mathrm{P}^{(1)} = 0$, we have that:

$$\mathbb{E}\left[R(T, \mathrm{P})\right] \leq C_{r,d}\sqrt{kT}, \tag{11}$$

for any rank-k projection matrix P. Here, the expectation is w.r.t. randomization in the algorithm, and the multiplicative factor $C_{r,d} = \frac{\sqrt{d^2(d-1)^2 + r^4(d-r)^2}}{r(r-1)}$.

Note that the regret bound degrades gracefully with the fraction of observed entries. The parameter $C_{r,d}$ determines the tradeoff between iteration complexity and the cost of data access. As $r \to d$, one can see that $C_{r,d} \to 1$, which recovers the bound in Theorem 2.1. On the other hand, as r becomes smaller, $C_{r,d}$ grows and for $r = 2$, one can see that $C_{r,d} = O(d^2)$. This is especially interesting for applications where abundant number of samples are provided, but obtaining measurements per sample is highly costly.

Stochastic Setting As in Section 4, MGD with partial observations can have worst case per-iteration complexity $O(d^3)$. We make the same stochastic assumptions as in the previous section, extend Oja's algorithm to work with the gradients $\widehat{\mathrm{g}}_t$ and adapt the regret bound from (Allen-Zhu & Li, 2017b). All of our remarks about the per-iteration computational cost of Oja from section 4 still hold.

Theorem 5.3 (Oja regret from partial observations)**.** Assume that $\mathrm{x}_t \sim \mathcal{D}$ for all t in $1, \ldots, T$ and that $\|\mathrm{x}_t\| \leq 1$ almost surely. Let $\mathrm{C} = \mathbb{E}_{\mathrm{x}\sim\mathcal{D}}[\mathrm{x}\mathrm{x}^\top]$ with top eigenvalue λ. Then, after T iterations of Oja's algorithm with gradients $\widehat{\mathrm{g}}_t$, initialization $\mathrm{P}^{(1)} = \mathrm{u}\mathrm{u}^\top$, where $\mathrm{u} \sim \mathcal{N}(0, \mathrm{I})$ and step size $\eta = \frac{\log(1+\delta/9)}{(11\alpha^2+4\lambda^2)\sqrt{T}}$, where $\alpha = \frac{d(d-1)}{r(r-1)}$, with probability $1 - \delta$ it holds that:

$$\mathbb{E}_{\mathcal{S},\mathcal{R}}[R(T, \mathrm{P})] \leq \frac{2\sqrt{T}}{\log(1 + \delta/18)} + \frac{\sqrt{T}(11\alpha^2 + 4\lambda^2)}{\log(1+\delta)} O\left(\log\left(d + \log\left(\frac{1}{\delta}\right)\right) - \log\left(\frac{1}{\delta}\right)\right),$$

for any rank-1 projection matrix P.

When $r \to d$, we essentially recover the regret bound in Theorem 4.3 and the same comparison done in section 4 holds when discussing total computational time for MGD versus Oja to achieve ϵ-average regret. When $r \to 2$, we see that $\alpha \to d^2$ and the regret for Oja's algorithm becomes $O\left(d^4 \log(d)\sqrt{T}\right)$. Again we see that MGD with partial observations has a better worst-case regret bound in terms of d and that both algorithms become intractable for large d.

6. Robust streaming PCA

Despite its ubiquitous nature, PCA as well as other subspace learning methods have a major weakness – they are extremely sensitive to outliers. Corrupted data points, which we refer to as outliers, can completely throw off the estimate of the principal subspace even with a single outlier (Huber & Ronchetti, 2009). In practice, we may encounter a high percentage of corruption (Zhang & Lerman, 2014) and in theory (under some assumptions) the percentage of outliers tolerated by robust PCA algorithms can be significantly higher than the common 50% breakdown point of point estimators (Zhang & Lerman, 2014; Lerman et al., 2012; Hardt & Moitra, 2013). In such cases, the inliers may still be viewed as arising from $\mathcal{D}$, but the outliers are likely to be generated by a different distribution or may be even hard to model. The presence of these outliers, whose proportion may be significant, can completely distort the estimate of the expected variance and therefore the PCA subspace.

There have been several attempts to endow PCA with resilience against outliers or other forms of gross corruptions (see e.g., (De La Torre & Black, 2003; Fischler &

Bolles, 1981; Gnanadesikan & Kettenring, 1972; Hampel et al., 2005; Huber & Ronchetti, 2009; Hubert et al., 2005; Ke & Kanade, 2005; Maronna et al., 2006; Recht et al., 2010; Xu et al., 2010)). Following (Chandrasekaran et al., 2011), Candès et al. (2011) established a convex de-convolution method for extracting low-dimensional subspace structure in the presence of gross but sparse uniformly distributed *element-wise corruptions*. Given a dataset $\mathrm{X} \in \mathbb{R}^{d\times n}$, the robust PCA formulation considered by (Chandrasekaran et al., 2011) and (Candès et al., 2011), seeks a rank-k representation of X, denoted by $\mathrm{L} \in \mathbb{R}^{d\times n}$, that minimizes the ℓ_1-norm of the residuals, $\|\mathrm{S}\|_1$, where $\mathrm{S} := \mathrm{X} - \mathrm{L}$.

The seminal work of (Chandrasekaran et al., 2011) and (Candès et al., 2011) inspired the development of many other convex methods for robust PCA, that are robust in the *presence of outliers* (instead of element-wise corruptions) (Xu et al., 2012; McCoy & Tropp, 2011; Zhang & Lerman, 2014; Lerman et al., 2012). These works consider *absolute subspace deviations*, i.e. they seek a rank-k subspace that minimizes $\sum_{i=1}^{n} \|\mathrm{x}_i - \mathrm{P}\mathrm{x}_i\|_2$, where $\|\cdot\|$ denotes the ℓ_2-norm. They involve various convex relaxations of this minimizer. Of particular interest to us are the Geometric Median Subspace (GMS) algorithm (Zhang & Lerman, 2014) and the REAPER algorithm (Lerman et al., 2012). We prefer them since they do not require an arbitrary unknown regularization parameter, they can deal with significantly high percentage of outliers (both in theory and in practice) and their batch formulations are faster.

However, both GMS and REAPER are batch algorithms and therefore do not scale to big data. In this section, we study robust PCA in a streaming setting. We build on absolute subspace deviations model of (Zhang & Lerman, 2014) and (Lerman et al., 2015) and propose a robust analogue of streaming PCA that imparts it robustness in face of outliers. Unlike Goes et al. (2014) who consider robust PCA in a stochastic setting and focus on the ϵ-suboptimality, our goal is to bound the following regret,

$$R_{\mathtt{abs}}(T) = \sum_{t=1}^{T} \|\mathrm{x}_t - \mathrm{P}^{(t)}\mathrm{x}_t\|_2 - \inf_{\mathrm{P}\in\mathcal{P}_k} \sum_{t=1}^{T} \|\mathrm{x}_t - \mathrm{P}\mathrm{x}_t\|_2,$$

for *any* sequence $\mathrm{x}_1, \ldots, \mathrm{x}_T \in \mathbb{R}^d$.

The gradient of the loss function $\ell(\mathrm{x}_t) = \|\mathrm{x}_t - \mathrm{P}^{(t)}\mathrm{x}_t\|_2$ in the formulation above is given as $\frac{(\mathrm{I}-\mathrm{P}^{(t)})\mathrm{x}_t\mathrm{x}_t^\top}{\|\mathrm{x}_t - \mathrm{P}^{(t)}\mathrm{x}_t\|}$. This is a rank-one update that is not guaranteed to be symmetric. In order for our analysis to go through we consider the following symmetrized loss: $\frac{1}{2}\mathbb{E}_{\mathrm{x}}[\|\mathrm{x}-\mathrm{P}\mathrm{x}\|_2 + \|\mathrm{x}-\mathrm{P}^\top\mathrm{x}\|_2]$. The instantaneous gradient at the t-th iteration is then given by

$$\mathrm{g}_t = -\frac{\mathrm{x}_t\mathrm{x}_t^\top(\mathrm{I}-\mathrm{P}^{(t)}) + (\mathrm{I}-\mathrm{P}^{(t)})\mathrm{x}_t\mathrm{x}_t^\top}{2\|(\mathrm{I}-\mathrm{P}^{(t)})\mathrm{x}_t\|_2}.$$

We denote $\mathrm{y}_t = (\mathrm{I} - \mathrm{P}^{(t)})\mathrm{x}_t$, and $c_t = \frac{\eta}{2\|\mathrm{y}_t\|_2}$, then the proposed `abs`-MGD update can be written as:

$$\mathrm{P}^{(t+1)} = \Pi_F\left(\mathrm{P}^{(t)} + c_t(\mathrm{x}_t\mathrm{y}_t^\top + \mathrm{y}_t\mathrm{x}_t^\top)\right). \qquad (12)$$

We bound the regret of `abs`-MGD updates in (12) as follows.

Theorem 6.1. Assume $\|\mathrm{x}_t\| \le 1$ for all t in $1, \cdots, T$. Then, after T iterations of MGD with step size $\eta = \sqrt{\frac{k}{T}}$, and starting at $\mathrm{P}^{(1)} = 0$, we have that:

$$R_{\mathtt{abs}}(T) \le \sqrt{kT}.$$

7. Experimental Results

Per iteration complexity. Before presenting an empirical evaluation of our algorithms we would like to discuss their computational efficiency in theory. Note that each variant of the MGD algorithm involves updating the current iterate $\mathrm{P}^{(t)} \in \mathbb{R}^{d\times d}$ with a rank-1 or a rank-2 matrix and then projecting onto a convex set of constraints. Since the projection step operates on the eigenvalues of the current iterate (Arora et al., 2013), a naive implementation would require $O(d^3)$ time per iteration. To avoid recomputing eigenvalues, we can keep an up-to-date eigendecomposition of each iterate and perform an efficient rank-1 (or rank-2) update which takes $O(d\tilde{k}^2)$ time where $\tilde{k} = \mathrm{rank}(\mathrm{P}^{(t)})$. Of course, $\tilde{k}$ may grow as large as d. In contrast, Oja's algorithm and its variants take only $O(dk^2)$ time per iteration.

Datasets and step-size tuning. We evaluate empirical performance of our algorithms with missing data (MGD-MD, Oja-MD) and partial observations (MGD-PO, Oja-PO) on two real datasets, MNIST (LeCun et al., 1998) and XRMB (Westbury, 1994) against vanilla MGD and classic Oja's algorithm (Oja, 1982) as well as with the state-of-the-art algorithm (GROUSE) of Balzano et al. (2010). The learning rate for variants of MGD and Oja's algorithm is set to $\eta_t = \frac{\eta_0}{\sqrt{t}}$, for MGD-PO to $\eta_t = \frac{r^2\eta_0}{d^2\sqrt{t}}$, and for MGD-MD to $\eta_t = q^2\frac{\eta_0}{\sqrt{t}}$. The initial learning rate η_0 is chosen using cross validation on a held-out set. The learning rate for GROUSE is set to $\eta_t = \frac{\eta_0}{\sqrt{t}}$, even though the theory suggests a step size proportional to $\frac{1}{t}$; this choice was made since GROUSE did not converge in our experiments with a step size of $\Theta(1/t)$.

Empirical results. Figures 1 and 2 show the objective as a function of the number of *observations* as well as a function of elapsed time, for different values of rank (k), on XRMB and MNIST datasets, respectively. We see that both MGD-MD and MGD-PO recover the subspace even when nearly 92% of the observations are missing. We see consistently across experiments that (a) MGD outperforms

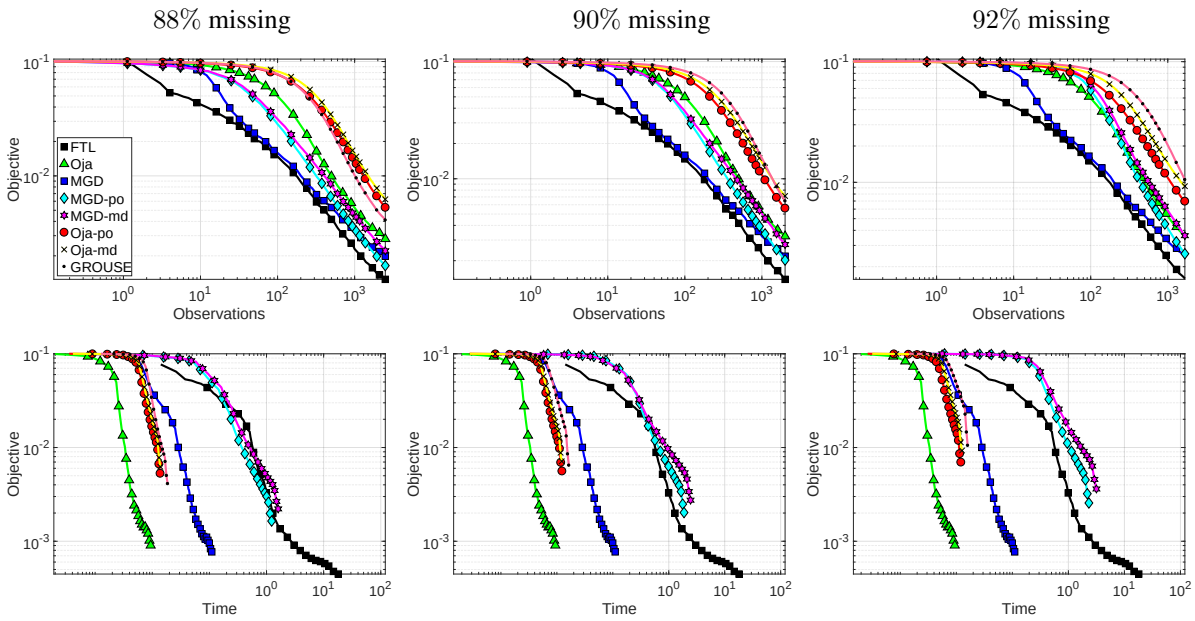

Figure 1: Comparisons of Oja, MGD, MGD-PO, MGD-MD, Oja-PO, Oja-MD, GROUSE for PCA with missing data on XRMB dataset, in terms of the variance captured on a test set as a function of number of observed entries for rank-4 (top) and runtime (bottom).

all other algorithms in terms of progress per number of observations, and (b) Oja's algorithm always performs better than Oja-MD and Oja-PO both of which are nearly as bad as GROUSE. For Oja's algorithm, we note that even though the theoretical guarantees in Theorems 4.3, and 5.3 only hold for $k = 1$, our experiments suggest that the sub-linear regret guarantees perhaps still hold for larger k. Furthermore, the experimental results seem to be consistent with theoretical guarantees in terms of per-iteration progress and total runtime.

Comparison with theory. Our theoretical bounds for $k = 1$ suggest that MGD is always better than MGD-PO and MGD-MD when the observation ratio is small. Again, when the observation ratio is small, Oja-MD and Oja-PO have worse upper bounds on the average regret, compared to MGD-PO and MGD-MD, by at least a factor of dimensionality and that both should perform worse than Oja with full observations. Our experiments confirm these observations. Note that even though both in Figures 1 and 2, MGD-PO and MGD-MD seem to perform as well as MGD, this is in term of observations and not in terms of number of iterations, which are far fewer for MGD. Our experiments suggest extensions of our theory in at least two directions. First, for Oja's algorithm and its variants, similar theoretical upper bounds on the regret should hold for general k. Second, it is possible that there are matching lower bounds for the algorithms dealing with partially observed and missing data.

Runtime. As expected, Oja's algorithm and its variants perform much better in terms of progress per runtime as their per-iteration complexity is only $O(dk^2)$. MGD performs as well as GROUSE, Oja-PO, and Oja-MD in terms of runtime. This is because the rank of the iterates $\mathrm{P}^{(t)}$ remains in $O(k)$. This is not the case, however, for MGD-PO and MGD-MD and their progress per runtime is significantly slower.

Because of space constraints we deferred some experiments, including plots for per-iteration progress of the algorithms and plots for Robust-MGD, to the supplementary; the above observations still hold for the deferred plots.

8. Discussion

In this paper, we study PCA in a streaming setting with no distributional assumptions on data. In the big data setting we consider, data is often contaminated with noise, outliers, or observed partially. We propose several efficient algorithms to solve the above problems and prove sublinear regret bounds. As we already discuss in the paper, the data which our algorithms process can be generated by an adversary and thus we quantify the loss of our algorithms in terms of regret. Theorem 2.1 gives a bound on the regret of MGD with respect to any *fixed* subspace P chosen in hindsight. One might argue that this is not a real-world setting and the subspaces we are comparing against should be allowed to vary as incoming data is observed. We can strengthen our results by comparing against sequences of subspaces, with a bounded total shift, all chosen in hindsight.

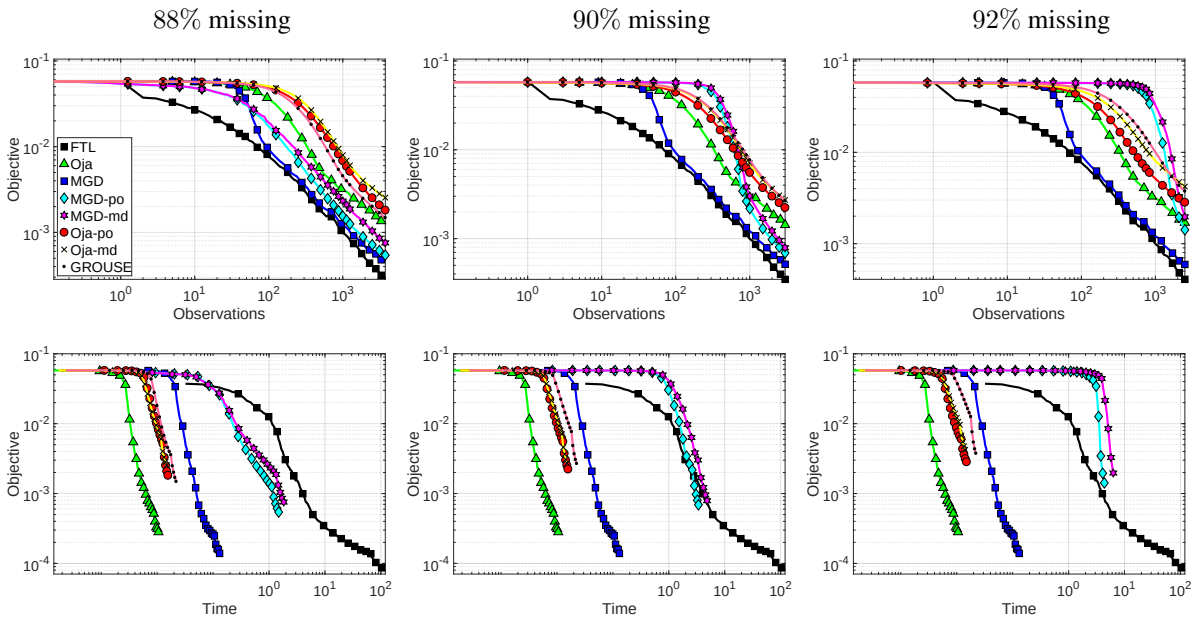

Figure 2: Comparisons of Oja, MGD, MGD-PO, MGD-MD, Oja-PO, Oja-MD and GROUSE for PCA with missing data on MNIST dataset, in terms of the variance captured on a test set as a function of number of observed entries for rank-2 (top), and runtime (bottom).

Theorem 8.1 (MGD switch regret)**.** Assume $\|\mathrm{x}_t\| \leq 1$ for all t in $1, \cdots, T$. Then, after T iterations of MGD with step size $\eta = \sqrt{\frac{k}{T}}$, and starting at $\mathrm{P}^{(1)} = 0$, we have that

$$\sum_{t=1}^{T} \mathrm{x}_t^\top \mathrm{P}_*^{(t)} \mathrm{x}_t - \sum_{t=1}^{T} \mathbb{E}_{\texttt{round}} \left[\mathrm{x}_t^\top \mathrm{P}^{(t)} \mathrm{x}_t \right] \leq \sqrt{(6\sqrt{k}S + k)T}$$

where $(\mathrm{P}_*^{(t)})_{t=1}^T$ is any competing sequence of subspaces in $\mathcal{C}$ with total shift $\sum_{t=1}^{T} \|\mathrm{P}_*^{(t)} - \mathrm{P}_*^{(t-1)}\|_F \leq S$.

Our experiments suggest the following directions for future work: (a) extend the analysis of the Oja's algorithm (i.e. results in Theorems 3.2, 4.3 and 5.3) to general $k > 1$), and (b) show lower bounds for regret guarantees in Section 4 and Section 5 which depend on the number of missing entries.

We would also like to investigate Oja-like updates for the ℓ_2 Robust PCA formulation in Section 6, which preserve the low-rank structure of the iterates $\mathrm{P}^{(t)}$. Analyzing such an algorithm, even in the special cases when data is stochastic and $k = 1$, seems like a daunting task, because unlike the standard PCA formulation, we do not have a closed form solution for the optimization problem. This in turn is an obstacle when trying to come up with potential functions tracking the progress of the proposed algorithms.

We also remark that for robust streaming PCA in Section 6, the iterates $\mathrm{P}^{(t)} \in \mathbb{R}^{d \times d}$ are in the convex set $\mathcal{C}$ defined in equation (2), however, they need not necessarily be projection matrices. Furthermore, due to the non-linear nature of the objective we can not simply use the rounding procedure as in (Warmuth & Kuzmin, 2008). In practice, we observe that one can use the rank-k projection retrieved from the top k eigen-vectors of $\mathrm{P}^{(t)}$. This can be partially justified by the results in (Lerman et al., 2015) which state that under certain mild assumptions on the outliers, the solution to the optimization problem $\min_{\mathrm{P} \in \mathcal{P}} \sum_{t=1}^{T} \|\mathrm{x}_t - \mathrm{P}\mathrm{x}_t\|_2$ is close to the rank-k projection matrix retrieved from the solution of the convex relaxation $\min_{\mathrm{P} \in \mathcal{C}} \sum_{t=1}^{T} \|\mathrm{x}_t - \mathrm{P}\mathrm{x}_t\|_2$. The result in Theorem 6.1 can be extended to show that the sequence $(\mathrm{P}^{(t)})_{t=1}^T$ does not suffer large regret against the optimal $\mathrm{P}^* \in \mathcal{C}$. In future work, we hope to show that this implies that the average of the iterates $(\mathrm{P}^{(t)})_{t=1}^T$, or the final iterate $\mathrm{P}^{(T)}$, is close in norm to P^*. This together with results in (Lerman et al., 2015) would explain why in practice using the rank-k projection closest to $\mathrm{P}^{(t)}$ works well.

Another possible direction for future work is to design and analyze streaming algorithms for related component analysis techniques in noisy settings. In particular, algorithms based on online mirror descent have been used in the context of partial least squares (Arora et al., 2016) and canonical correlation analysis (Arora et al., 2017; Ge et al., 2016). It is natural to consider extensions of these algorithms to noisy settings with missing data and outliers.

Acknowledgements

This research was supported in part by NSF BIGDATA grant IIS-1546482.

References

Allen-Zhu, Z. and Li, Y. First efficient convergence for streaming k-PCA: a global, gap-free, and near-optimal rate. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pp. 487–492. IEEE, 2017a.

Allen-Zhu, Z. and Li, Y. Follow the compressed leader: Faster online learning of eigenvectors and faster MMWU. In *International Conference on Machine Learning*, pp. 116–125, 2017b.

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

Arora, R., Cotter, A., and Srebro, N. Stochastic optimization of PCA with capped MSG. In *Advances in Neural Information Processing Systems*, pp. 1815–1823, 2013.

Arora, R., Mianjy, P., and Marinov, T. Stochastic optimization for multiview representation learning using partial least squares. In *International Conference on Machine Learning*, pp. 1786–1794, 2016.

Arora, R., Marinov, T. V., Mianjy, P., and Srebro, N. Stochastic approximation for canonical correlation analysis. In *Advances in Neural Information Processing Systems*, pp. 4778–4787, 2017.

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

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

Beck, A. and Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3): 11, 2011.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optim.*, 21(2):572–596, 2011. ISSN 1052-6234. doi: 10.1137/090761793.

De La Torre, F. and Black, M. J. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Garber, D., Hazan, E., and Ma, T. Online learning of eigenvectors. In *International Conference on Machine Learning*, pp. 560–568, 2015.

Ge, R., Jin, C., Netrapalli, P., Sidford, A., et al. Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis. In *International Conference on Machine Learning*, pp. 2741–2750, 2016.

Gnanadesikan, R. and Kettenring, J. R. Robust estimates, residuals, and outlier detection with multiresponse data. *Biometrics*, pp. 81–124, 1972.

Goes, J., Zhang, T., Arora, R., and Lerman, G. Robust stochastic principal component analysis. In *Artificial Intelligence and Statistics (AISTATS)*, pp. 266–274, 2014.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. *Robust Statistics: The Approach Based on Influence Functions (Wiley Series in Probability and Statistics)*. Wiley-Interscience, New York, revised edition, April 2005. ISBN 0471735779. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471735779.

Hardt, M. and Moitra, A. Can we reconcile robustness and efficiency in unsupervised learning? In *Proceedings of the Twenty-sixth Annual Conference on Learning Theory (COLT 2013)*, 2013.

Huber, P. J. and Ronchetti, E. M. *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley, Hoboken, NJ, 2nd edition, 2009. ISBN 978-0-470-12990-6. doi: 10.1002/9780470434697.

Hubert, M., Rousseeuw, P. J., and Branden, K. V. ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47(1), 2005.

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

Ke, Q. and Kanade, T. Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pp. 739–746, June 2005.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lerman, G., McCoy, M., Tropp, J. A., and Zhang, T. Robust computation of linear models, or how to find a needle in a haystack. *ArXiv e-prints*, February 2012.

Lerman, G., McCoy, M. B., Tropp, J. A., and Zhang, T. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15 (2):363–410, 2015.

Lounici, K. et al. High-dimensional covariance matrix estimation with missing observations. *Bernoulli*, 20(3): 1029–1058, 2014.

Maronna, R. A., Martin, R. D., and Yohai, V. J. *Robust statistics: Theory and methods*. Wiley Series in Probability and Statistics. John Wiley & Sons Ltd., Chichester, 2006. ISBN 978-0-470-01092-1; 0-470-01092-4.

McCoy, M. and Tropp, J. Two proposals for robust PCA using semidefinite programming. *Elec. J. Stat.*, 5:1123–1160, 2011.

Mianjy, P. and Arora, R. Stochastic PCA with l 2 and l 1 regularization. In *International Conference on Machine Learning*, 2018.

Mitliagkas, I., Caramanis, C., and Jain, P. Streaming PCA with many missing entries. *Preprint*, 2014.

Nie, J., Kotłowski, W., and Warmuth, M. K. Online PCA with optimal regrets. In *International Conference on Algorithmic Learning Theory*, pp. 98–112. Springer, 2013.

Oja, E. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

Recht, B. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 12:3413–3430, 2011.

Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Shamir, O. Convergence of stochastic gradient descent for PCA. In *International Conference on Machine Learning*, pp. 257–265, 2016.

Tsuda, K., Rätsch, G., and Warmuth, M. K. Matrix exponentiated gradient updates for on-line learning and bregman projection. In *Journal of Machine Learning Research*, pp. 995–1018, 2005.

Warmuth, M. K. and Kuzmin, D. Online variance minimization. In *Learning theory*, pp. 514–528. Springer, 2006.

Warmuth, M. K. and Kuzmin, D. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9 (10), 2008.

Westbury, J. X-ray microbeam speech production database users handbook: Madison. *WI: Waisman Center, University of Wisconsin*, 1994.

Xu, H., Caramanis, C., and Mannor, S. Principal component analysis with contaminated data: The high dimensional case. In *COLT*, pp. 490–502, 2010.

Xu, H., Caramanis, C., and Sanghavi, S. Robust PCA via outlier pursuit. *Information Theory, IEEE Transactions on*, PP(99):1, 2012. ISSN 0018-9448. doi: 10.1109/TIT.2011.2173156.

Zhang, T. and Lerman, G. A novel M-estimator for robust PCA. *Journal of Machine Learning Research*, 15(1): 749–808, 2014.