Spectral guarantees for adversarial streaming PCA

Eric Price
UT Austin

September 2, 2023

Abstract

In streaming PCA, we see a stream of vectors $x_1, \dots, x_n \in \mathbb{R}^d$ and want to estimate the top eigenvector of their covariance matrix. We ask: how does the space complexity vary with the spectral ratio $R = \lambda_1/\lambda_2$? Existing algorithms require $\Omega(d^2/R)$ space. We show that:

- For $R \geq O(\log n \log d)$, Oja's algorithm solves PCA with $O(d)$ words of space to within $O(\frac{\log d}{R})$ error, and
- $\Omega(\frac{1}{R^2})$ error is necessary for any algorithm that uses $o(d^2/R^3)$ bits of space.

This shows, for $R = O(1)$, that $\Omega(d^2)$ space is needed to get an arbitrarily small constant approximation; but for $R > O(\log n \log d)$, that $\widetilde{O}(d)$ space is sufficient. These results stand in contrast to the stochastic setting, where the x_i are drawn iid from a (somewhat nice) distribution: there, Oja's algorithm works well down to $R = 1 + \varepsilon$.

1 Introduction

Principal Component Analysis (PCA) is a fundamental primitive for handling high-dimensional data by finding the highest-variance directions. At its most simple, given a data set $X \in \mathbb{R}^{n \times d}$ of n data points in d dimensions, we want to find the top eigenvector v^* of the covariance matrix $\Sigma = \frac{1}{n} X^T X$.

One common way to approximate v^* is the power method: start with a random vector u_0, then repeatedly multiply by Σ and renormalize. This converges to v^* at a rate that depends on the ratio of the top two eigenvalues, $R := \lambda_1/\lambda_2$. In particular, after $O(\log_R \frac{d}{\varepsilon})$ iterations we have $\|Pu_k\|^2 = 1 - \langle u_k, v^* \rangle^2 = \sin^2(u_k, v^*) \leq \varepsilon$ with high probability, where $P = I - v^*(v^*)^T$ projects away from v^*.

But what if the data points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ arrive in a streaming fashion? Directly applying the power method requires either nd space to store X, or d^2 space to store Σ. What can be done in smaller space? The question of streaming PCA has been extensively studied, in two main settings: *adversarial* and *stochastic* streams.

In the adversarial streaming setting, we want to solve PCA for an arbitrary set of data points in arbitrary order. Many of these algorithms store linear sketches of the data, such as AX and XB for Gaussian matrices A, B [CW09, BWZ16, Woo14b, Upa18, TYUC17]. These results give a Frobenius guarantee for rank-k approximation of X. Specialized to $k = 1$, the result direction $\widehat{u}$ satisfies

$$\|X(I - \widehat{u}\widehat{u}^T)\|_F^2 \leq (1 + \varepsilon)\|XP\|_F^2$$

which is equivalent to

$$\widehat{u}^T \Sigma \widehat{u} \geq \lambda_1 - \varepsilon \sum_{i>1} \lambda_i.$$

The best result here is FrequentDirections [Lib13, GLPW16], which is a deterministic insertion-only algorithm rather than a linear sketch. It uses $O(d/\varepsilon)$ space to get the guarantee, which is optimal [CW09]. Unfortunately, this Frobenius guarantee can be quite weak: if the eigenvalues do not decay and we only have a bound on $R = \lambda_1/\lambda_2$, to get $\|P\widehat{u}\|^2 \leq 0.1$ we need $\varepsilon < \frac{R}{d}$, which means $\Theta(d^2/R)$ space. The well-known spiked covariance model, where the x_i are iid Gaussian with covariance that has eigenvalues $\lambda_2 = \lambda_3 = \cdots = \lambda_d$, is one example where this quadratic space bound appears.

In the stochastic streaming setting, the x_i are drawn iid from a somewhat nice distribution. The goal is to converge to the principal component of the true distribution using little space and few samples. Algorithms for the stochastic setting are typically iterative, using $O(d)$ space and converging to the true solution with a sample complexity depending on how "nice" the distribution is. Examples include Oja's algorithm [Oja82, BDF13, JJK+16, AL17, HNWTW21, HNWW21, LSW21] and the block power method [ACLS12, MCJ13, HP14, BDWY16]. Oja's algorithm starts with a random v_0, then repeatedly sets

$$v_i = v_{i-1} + \eta_i x_i x_i^T v_{i-1}$$

for some small learning rate η_i. These analyses depend heavily on the data points being iid[1]. In return, they can get a stronger *spectral* guarantee than the sketching algorithms. The bounds are not directly comparable to the sketching algorithms (not only does the sample complexity depend on the data distribution, but the convergence is to the principal component of the true distribution rather than the empirical Σ), but in the spiked covariance setting they just need $n \geq \widetilde{O}((1 + \frac{1}{R-1})^2 d)$ rather than $O(d^2/R)$. That is, they use near-linear samples down to $R = 1 + \varepsilon$.

[1] Or nearly so; for example, [JJK+16] requires that the x_i are independent with identical covariance matrices.

So the situation is: algorithms that handle arbitrary data need $O(d^2/R)$ space for a spectral guarantee. Iterative methods have a good spectral guarantee—linear space and often near-linear samples for constant R—but only handle iid data. Is this separation necessary, or can we get a good spectral guarantee in the arbitrary-data setting? In this paper we ask:

Is $\Omega(d^2)$ space necessary for constant R?
How large does R need to be for $\widetilde{O}(d)$ space to be possible?

Our main result is that linear space *is* possible for polylogarithmic spectral gaps. In fact, Oja's method essentially achieves this:

Theorem 1.1 (Performance of Oja's method in adversarial streams)**.** *For any sufficiently large universal constant C, suppose η is such that $\eta n\lambda_1 > C\log d$ and $\eta n\lambda_2 < \frac{1}{C\log n}$. If $\eta\|x_i\|^2 \le 1$ for every i, then Oja's algorithm with learning rate η returns $\widehat{v}$ satisfying $\|P\widehat{v}\| \le \sqrt{\eta n\lambda_2} + d^{-9}$ with $1 - d^{-\Omega(C)}$ probability.*

Moreover, Oja's method can be modified (Algorithm 1) so that in addition, regardless of λ_1 and λ_2, if $\eta\|x_i\|^2 \le 1$ for all i then either $\|P\widehat{v}\| \le \sqrt{\eta n\lambda_2} + d^{-9}$ or $\widehat{v} = \perp$.

If $R > O(\log n \log d)$, there exists an η that satisfies the eigenvalue condition. However, Theorem 1.1 requires knowing η and that no single $\|x_i\|$ is too large. It's fairly easy to extend the algorithm to remove both restrictions, as well as describe the performance with respect to finite precision. Algorithm 2 simply runs Oja's method for different learning rates and picks the smallest one that works; unless any single x_i has too large $\|x_i\|^2$ violating Theorem 1.1, in which case it outputs that x_i.

Algorithm 1 Oja's Algorithm, checking the growth of $\|v_n\|$ to identify too-small learning rates.

procedure OjaCheckingGrowth(X, η)
  Choose $\widehat{v}_0 \in S^{d-1}$ uniformly. ▷ All numbers stored to $O(\log(nd))$ bits of precision
  Set $s_0 = 0$.
  for $i = 1, \dots, n$ **do**
    $v_i' \leftarrow (1 + \eta x_i x_i^T)\widehat{v}_{i-1}$.
    $\widehat{v}_i \leftarrow \frac{v_i'}{\|v_i'\|}$
    $s_i \leftarrow s_{i-1} + \log\|v_i'\|$
  end for
  if $s_n \le 10\log d$, **return** $\perp$ ▷ Returns $\perp$ rather than a wrong answer if η is too small.
  else return $\widehat{v}_n$
end procedure

Theorem 1.2 (Full algorithm)**.** *Let $X \in \mathbb{R}^{n\times d}$ have b-bit entries, so each $X_{i,j} = 0$ or $2^{-b} \le |X_{i,j}| \le 2^b$, for $b > \log(dn)$. Whenever $R > O(\log n \log d)$, Algorithm 2 uses $O(b^2 d)$ bits of space and returns $\widehat{v}$ satisfying $\|P\widehat{v}\|^2 \lesssim \frac{\log d}{R} + d^{-9}$ with high probability.*

Lower bounds. Say an algorithm solves ε-approximate PCA if it returns u with $\|Pu\|^2 \le \varepsilon$. So Theorem 1.2 shows that it is possible to solve $O(\frac{\log d}{R})$-approximate PCA in near-linear space. A natural question is whether much higher accuracy is possible. Unfortunately, we show that quadratic space is needed to beat accuracy polynomial in R:

Algorithm 2 Algorithm handling unknown learning rate and large-norm entries

procedure AdversarialPCA(X, b) ▷ $X \in \mathbb{R}^{n\times d}$ has $X_{i,j} = 0$ or $2^{-b} \le |X_{i,j}| \le 2^b$
  Define $\eta_i = 2^i$ for integer i, $|i| \le 4b + \log(nd^2) + O(1)$.
  In parallel run OjaCheckingGrowth(X, η_i) for all i, getting $v^{(i)}$.
  In parallel record $\overline{x}$, the single x_i of maximum $\|x_i\|$.
  Let i^* be the smallest i with $v^{(i)} \neq \perp$.
  if $\eta_{i^*}\|\overline{x}\|_2 \ge 1$, **return** $\frac{\overline{x}}{\|\overline{x}\|}$.
  else return $v^{(i^*)}$.
end procedure

Theorem 1.3 (Lower bound)**.** *There exists a universal constant $C > 1$ such that: for any $R > 1$, $\frac{1}{CR^2}$-approximate PCA on streams with spectral gap R requires at least $\frac{d^2}{CR^3}$ bits of space for sufficiently large $d > \text{poly}(R)$.*

In particular this means, for any constant R, there exists a constant $\varepsilon > 0$ such that ε-solving PCA requires $\Omega(d^2)$ bits of space, i.e., storing the entire covariance matrix is nearly optimal. By contrast, Theorem 1.2 shows that for $R = \Theta(\log n \log d)$, ε-solving PCA for any constant $\varepsilon > 0$ is possible in $\widetilde{O}(d)$ bits of space. This is a much lower threshold than the $R = \widetilde{\Theta}(d)$ needed for near-linear space by existing analyses.

1.1 Related Work

Oja's algorithm has been extensively studied in the stochastic setting; see, e.g., [BDF13, JJK+16, AL17, HNWTW21, HNWW21, LSW21]. Since the goal in this setting is to approximate the underlying distribution's principal components, there is a minimum sample complexity for even an offline algorithm to estimate the principal component. This line of work [JJK+16] can show that Oja's algorithm has a similar sample complexity to the optimal offline algorithm, even for spectral ratios R close to 1.

Our analysis of Oja's algorithm is of necessity quite different from these stochastic-setting analyses. Oja's algorithm returns $v_n = B_n v_0$ for a transformation matrix $B_n = (I + \eta_n x_n x_n^T)(I + \eta_{n-1} x_{n-1} x_{n-1}^T) \cdots (I + \eta_1 x_1 x_1^T)$. In the stochastic setting, B_n is a random variable, with $\mathbb{E}[B_i \mid B_{i-1}] = (I + \eta\Sigma)B_i$; the analyses focus on matrix concentration of B_n, essentially to bound the deviation of B_n around the "expected" $(I + \eta\Sigma)^n$. In our arbitrary-data setting, B_n isn't a random variable at all. The only randomness is the initialization v_0. This makes our analysis quite different, instead tracking how much $\widehat{v}_i$ can move under the covariance constraints.

Our lower bound construction is closely related to one in [Woo14a], which shows an $\Omega(dk/\varepsilon)$ lower bound for a $(1+\varepsilon)$-approximate rank-k approximation of Σ in Frobenius norm. The [Woo14a] construction for $k = 1$ and $\varepsilon = \Theta(\frac{1}{n})$ is very similar to ours, and would give an $\Omega(d^2)$ lower bound for a small constant approximation when $R < 2$.

Much of the prior work on streaming PCA, for both the adversarial and stochastic settings, is focused on solving k-PCA not just the single top direction. We leave the extension of our upper bound to general k as an open question.

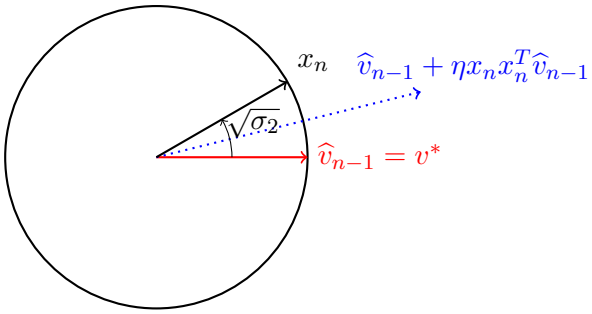

Figure 1: Suppose $\eta = 1$. Then even after convergence to v^* exactly, a single final sample can skew the result by $\Theta(\sqrt{\sigma_2})$. For smaller η, the same can happen with $\frac{1}{\eta}$ final samples.

2 Proof Overview

2.1 Upper bound

Unlike most work in the stochastic setting, we use a fixed learning rate η throughout the stream. The x_i correlated with v^* could all arrive at the beginning or the end of the stream, and we want to weight them equally so that at least we can solve the commutative case where Oja's algorithm is relatively simple.

As a basic intuition, Oja's algorithm returns $\widehat{v}_n = \frac{v_n}{\|v_n\|}$, where

$$\begin{aligned} v_n &= (I + \eta x_n x_n^T)(I + \eta x_{n-1} x_{n-1}^T) \cdots (I + \eta x_1 x_1^T) v_0 \\ &\approx e^{\eta x_n x_n^T} e^{\eta x_{n-1} x_{n-1}^T} \cdots e^{\eta x_{n-1} x_{n-1}^T} v_0 \end{aligned}$$

where the approximation is good when $\eta \|x_i\|^2 \ll 1$. Imagine that these matrix exponentials commute (e.g., each x_i is e_j for some j). Then we would have

$$v_n \approx e^{\eta X^T X} v_0. \tag{1}$$

This suggests that the important property of the learning rate η is the spectrum of $\eta X^T X$. Let $\eta X^T X$ have top eigenvalue $\sigma_1 = n\eta\lambda_1$, with corresponding eigenvector v^*, and all other eigenvalues at most $\sigma_2 = n\eta\lambda_2$. For Theorem 1.1, we would like to show that Oja's algorithm works if $\sigma_1 > O(\log d)$ and $\sigma_2 < \frac{1}{O(\log n)}$.

For (1) to converge to v^*, as in the power method, we want the v^* coefficient of v_0 to grow by a poly(d) factor more than any other eigenvalue, i.e., $e^{\sigma_1} \geq \text{poly}(d) e^{\sigma_2}$ or $\sigma_1 \geq \sigma_2 + O(\log d)$. So we certainly need to set η such that $\sigma_1 \geq O(\log d)$. But how large a spectral gap do we need, i.e., how small does σ_2 need to be?

One big concern for adversarial-order Oja's algorithm is: even if most of the stream clearly emphasizes v^* so v_i converges to it, a small number of inputs at the end could cause v_n to veer away from v^* to a completely wrong direction. This can't happen in the commutative setting, but it can happen in general: v_n can rotate by $\Theta(\sqrt{\sigma_2})$, by ending the stream with $\frac{1}{\eta}$ copies of $v^* + \sqrt{\sigma_2} v'$ (see Figure 1). But this is the worst that can happen. We show:

Lemma 3.2 (Growth implies correctness)**.** *For any v_0 and all i, $\|P\widehat{v}_i\| \leq \sqrt{\sigma_2} + \frac{\|Pv_0\|}{\|v_i\|}$.*

This lemma has two useful implications: first, if we ever get close to v^*, the final solution will be at most $\sqrt{\sigma_2}$ further from v^*. Second, no matter where we start, the final output is good if $\|v_n\|$

is very large. This is how Algorithm 1 can return either a correct answer or $\perp$: it just observes whether $\|v_n\|$ has grown by $\text{poly}(d)$.

So it suffices to show that $\|v_n\|$ is large for a random v_0; and since v_0 starts with a random $\frac{1}{\text{poly}(d)}$ component in the v^* direction, it in fact suffices to show that $\|v_n\|$ would grow by $\text{poly}(d)$ if Oja's algorithm started at $v_0 = v^*$. Now, one can show that

$$\|v_n\|^2 \geq e^{\sum_{i=1}^n \eta \langle x_i, \widehat{v}_{i-1}\rangle^2}. \tag{2}$$

So if v_i were always exactly v^*, we would have $\|v_n\|^2 \geq e^{\eta (v^*)^T X^T X v^*} = e^{\sigma_1} \geq \text{poly}(d)$ as needed. And if we start at v^*, then Lemma 3.2 implies $\|P\widehat{v}_i\| \leq \sqrt{\sigma_2}$ for all i, so we don't ever deviate *much* from v^*. But v_i can deviate a little bit, which could decrease $\langle x_i, \widehat{v}_{i-1}\rangle^2$. By how much? Well, it's easy to show

$$\eta \langle x_i, \widehat{v}_{i-1}\rangle^2 \geq \eta \frac{1-\sigma_2}{2} \langle x_i, v^*\rangle^2 - \eta \langle x_i, P\widehat{v}_{i-1}\rangle^2 \tag{3}$$

so we just need to show that

$$\eta \sum_i \langle x_i, P\widehat{v}_{i-1}\rangle^2 \ll \sigma_1. \tag{4}$$

We know that $\|P\widehat{v}_{i-1}\|^2 \leq \sigma_2$, and $\eta \sum_i \langle x_i, w\rangle^2 \leq \sigma_2$ for any fixed unit vector $w \perp v^*$, but the worry is that $P\widehat{v}_{i-1}$ could rotate through many different orthogonal directions; each direction w can only contribute σ_2^2 to $\eta \sum_i \langle x_i, P\widehat{v}_{i-1}\rangle^2$, but the total could conceivably be up to $\sigma_2^2 d$.

Our main technical challenge is to rule this out, so $\eta \sum_i \langle x_i, P\widehat{v}_{i-1}\rangle^2$ is small. For intuition, in this overview we just rule out $P\widehat{v}_{i-1}$ moving through many *elementary* basis vectors by showing

$$\sum_{j=1}^d \max_i \langle e_j, P\widehat{v}_{i-1}\rangle^2 \lesssim \sigma_2 \log^2 n \log \|v_n\|. \tag{5}$$

That is, $P\widehat{v}_{i-1}$ cannot rotate through $\sqrt{\sigma_2}$ correlation with each of the d different basis vectors (which would give a value of $\sigma_2 d$) unless $\|v_n\|$ is large (which is what we wanted to show in the first place).

First, we show that $\|v_n\|$ grows proportional to the *squared* movement of $P\widehat{v}_i$:

Lemma 3.3. *Suppose $Pv_0 = 0$. For any two time steps $0 \leq a < b \leq n$,*

$$\|P\widehat{v}_b - P\widehat{v}_a\|^2 \leq 4\sigma_2 \log \frac{\|v_b\|}{\|v_a\|}$$

As a result, for any subsequence $i_0, \ldots, i_k$ of iterations, the sum of squared movement has

$$\sum_{j=1}^k \|P\widehat{v}_{i_j} - P\widehat{v}_{i_{j-1}}\|^2 \lesssim \sigma_2 \log \|v_n\|.$$

We use a combinatorial lemma to turn this bound on squared distances over subsequences into (5). For any set of vectors A the following holds (see Figure 2):

Lemma 2.1 (Simplified version of Lemma 3.4)**.** *Let $A_0 = 0$, and $A_1, \ldots, A_n \in \mathbb{R}^d$ satisfy that every subsequence S of $\{0, \ldots, n\}$ has*

$$\sum_i \|A_{S_i} - A_{S_{i-1}}\|_2^2 \leq B.$$

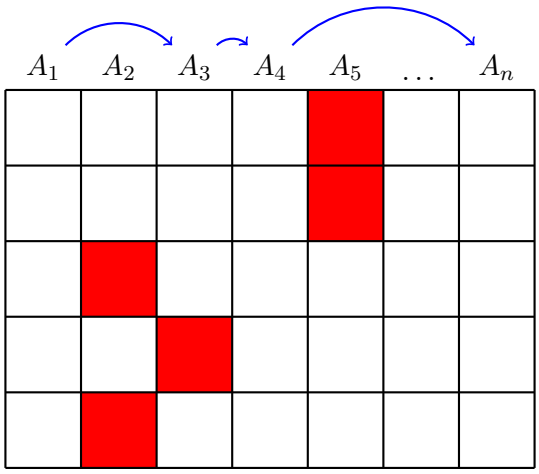

Figure 2: Lemma 2.1 states that, if the sum of squared distances across any subsequence of vectors A_i is at most B, then the vector selecting the maximum value in each coordinate has norm $B \log^2 n$.

for some $B > 0$. Then

$$\sum_{j=1}^{d} \max_{i \in [n]} (A_i)_j^2 \leq B(1 + \log_2 n)^2.$$

Applying Lemma 2.1 to $A_i := P\widehat{v}_i$ immediately gives (5).

Remark 2.2. *The $\log^2 n$ factor in Lemma 2.1 is why we need $R > O(\log d \log n)$, rather than just $R > O(\log d)$. At least as far as Lemma 2.1 is concerned, the factor is tight for $n = \Theta(d)$: $A_{i,j} := \log \frac{n}{1+|i-j|}$ has $B = \Theta(n)$ while $\sum_{j=1}^{d} \max_{i \in [n]} (A_i)_j^2$ is $\Theta(n \log^2 n)$.*

A similar approach, applied to $A_{i,j} = x_i^T P\widehat{v}_j$, lets us bound our actual target (4):

Lemma 3.5. *If $v_0 = v^*$, then*

$$\eta \sum_{i=1}^{n} \langle x_i, P\widehat{v}_{i-1} \rangle^2 \lesssim \sigma_2^2 \log^2 n \log \|v_n\|$$

Combined with (2) and (3), this implies that $\|v_n\| \geq e^{\Omega(\sigma_1)}$ if $\sigma_2 \ll \frac{1}{\log n}$. And by Lemma 3.2 this means the final answer $\widehat{v}_n$ is $\sqrt{\sigma_2} + d^{-C}$ close to v^*, so the algorithm succeeds.

2.2 Lower bound

To give an $\Omega(d^2)$ lower bound for constant R, we construct a two-player one-way communication game, where Alice feeds a uniformly random stream into the algorithm and passes the state to Bob. Bob then repeatedly takes this state, adds a few more vectors, and extracts the PCA estimate. We will show that Bob is able to learn $\Omega(d^2)$ bits about Alice's input, and therefore the stream state must have $\Omega(d^2)$ bits. Our approach is illustrated in Figure 3.

Suppose that Alice feeds in a random binary stream $x_1, x_2, \ldots, x_n \in \{-1, 1\}^d$. What can Bob insert so the PCA solution reveals information about (say) x_1?

First, suppose Bob inserted $k-1$ more copies of x_1 for some constant k. Then (if $n < d/100$) the PCA solution would be very close to x_1: $v = \frac{x_1}{\|x_1\|}$ has $\|Xv\|^2 \geq kd$ from just the copies of x_1, while

1	-1	-1	1	-1	-1	1	1
1	**1**	**1**	**-1**	**1**	**1**	**-1**	**-1**
-1	1	1	-1	-1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1
1	**1**	**1**	**-1**	**1**	-1	1	1
1	**1**	**1**	**-1**	**1**	1	1	1
1	**1**	**1**	**-1**	**1**	-1	1	-1

Figure 3: Lower bound approach: Alice inserts a sequence of random bits (all but the last row). Bob knows the left side and wants to approximate the right side. To estimate the **blue** bits on the right, he adds $O(1)$ vectors using the corresponding **red** bits on the left and random bits on the right. With high probability, the principal component has constant correlation with the blue bits.

every orthogonal direction has variance at most $(\sqrt{n}+\sqrt{d})^2 \approx 1.1d$ by standard bounds on singular values of subgaussian matrices [RV10]. Thus the spectral ratio $R = \frac{\lambda_1}{\lambda_2} > \frac{k}{1.1}$, so the streaming algorithm should return a vector highly correlated with x_1. The problem with this approach is that Bob can't insert x_1 without knowing x_1, so the streaming PCA solution doesn't reveal any *new* information to him.

But what if Bob inserts $z_2, \ldots, z_k$ that match x_1 on the first 90% of bits, and are random on the remaining 10%? The top principal component u^* will still be highly correlated with x_1: the vector v that matches $x_1, z_2, \ldots, z_k$ on the first 90% of bits and is zero on the rest has variance that is a $\frac{0.9k}{1.1}$ factor larger than any orthogonal direction. A more careful analysis shows that the top principal component v^* is not only correlated with the 90% fraction of bits of x_1 shared with the z_i, but (on the remaining bits) is very highly correlated with the average $\frac{1}{k}(x_1 + z_2 + \cdots + z_k)$. In fact, it is *so* highly correlated with the average that v^* must be at least somewhat—$\Theta(1/k^2)$—correlated with the last 10% of bits in x_1. This analysis is robust to a PCA approximation, so the streaming PCA algorithm lets Bob construct $\widehat{v}$ with constant correlation with the last 10% of bits in x_1.

Thus Bob can learn $\Omega(d)$ bits about the first row by inserting $y_2, \ldots, y_k$ that match on the first 90% of bits and looking at the PCA solution on the last 10% of bits. If he does this for every row, he learns $\Omega(nd) = \Omega(d^2)$ bits about Alice's input. Therefore the algorithm state Alice sent needs $\Omega(d^2)$ space.

This construction is very similar to the one in [Woo14a] for lower-bounding low-rank Frobenius approximation. The differences in [Woo14a] are (1) Bob only inserts one row, so necessarily $R < 2$; and (2) Bob sets his unknown bits to 0 rather than ± 1 randomly. Presumably the second change would work just fine in our setting, so our main contribution here is the more careful analysis in terms of R.

3 Proof of Upper Bound

For most of this section we focus on Oja's method (Theorem 1.1), then in Section 3.4 we show Theorem 1.2. For simplicity, the proof is given assuming exact arithmetic. In Section 3.5 we discuss why $O(\log(nd))$ bits of precision suffice.

Setup. $\widehat{v}_i$ is the normalized state at time i, v_i is the unnormalized state, x_i is the sample, η is the learning rate, v^* is the direction of maximum variance, $P = I - v^*(v^*)^T$ to be the projection

matrix that removes the v^* component. Let $\sigma_1 = \eta\|X^TX\|$ and $\sigma_2 = \eta\|PX^TXP\|$, so:

$$\sum_{i=1}^{n}\langle v^*, x_i\rangle^2 = \sigma_1 \tag{6}$$

$$\eta\sum_{i=1}^{n}\langle w, x_i\rangle^2 \le \sigma_2 \qquad (\forall w \perp v^*) \tag{7}$$

For much of the proof we will also need $\sigma_1 \ge C\log d$ and $\sigma_2 \le \frac{1}{C\log n}$, but this will be stated as needed.

Oja's algorithm works by starting with a (typically random) vector v_0, then repeatedly applying Hebb's update rule that "neurons that fire together, wire together":

$$v_i = v_{i-1} + \eta\langle x_i, v_{i-1}\rangle x_i = (I + \eta x_i x_i^T)v_{i-1}. \tag{8}$$

The algorithm only keeps track of the normalized vectors $\widehat{v_i} = v_i/\|v_i\|$, but for analysis purposes we will often consider the unnormalized vectors v_i.

The norm $\|v_i\|$ grows in each step, according to

$$\|v_i\|^2 = \|v_{i-1}\|^2(1 + (2\eta + \eta^2\|x\|^2)\langle x_i, \widehat{v}_{i-1}\rangle^2), \tag{9}$$

and in particular (since Theorem 1.1 assumes $\eta\|x_i\|^2 \le 1$)

$$\log\frac{\|v_i\|^2}{\|v_{i-1}\|^2} \ge \eta\langle x_i, \widehat{v}_{i-1}\rangle^2. \tag{10}$$

Our goal is to show that $\widehat{v}_n \approx v^*$, or equivalently, that $\|P\widehat{v}_n\|$ is small.

3.1 Initial Lemmas

Claim 3.1. *Let $0 \le a_1, a_2, \ldots, a_n$ and define $b_i = e^{\sum_{j\le i} a_i}$ for $i \in \{0, 1, \ldots, n\}$. Then:*

$$\sum_{i=1}^{n} a_i b_{i-1} \le b_n - 1.$$

Proof. This follows from induction on n. $n = 0$ is trivial, and then

$$\sum_{i=1}^{n} a_i b_{i-1} \le b_{n-1} - 1 + a_n b_{n-1} = (1 + a_n)b_{n-1} - 1 \le e^{a_n} b_{n-1} - 1 = b_n - 1.$$

□

Define $B_i = \frac{\|v_i\|^2}{\|v_0\|^2}$, and $A_i = \log\frac{B_i}{B_{i-1}}$ which satisfies $A_i \ge \eta\langle x_i, \widehat{v}_{i-1}\rangle^2$ by (10). Therefore

$$\eta\sum_{i=1}^{n}\langle x_i, v_{i-1}\rangle^2 \le \|v_0\|^2\sum_{i=1}^{n} A_i B_{i-1} \le \|v_0\|^2(B_n - 1) = \|v_n\|^2 - \|v_0\|^2 \tag{11}$$

by Claim 3.1. Then for any unit vector w with $Pw = w$,

$$\begin{aligned}
\langle v_n - v_0, w\rangle^2 &= \left(\eta\sum_{i=1}^{n}\langle x_i, v_{i-1}\rangle\langle x_i, w\rangle\right)^2 && \text{by (8)} \\
&\le \eta\sum_{i=1}^{n}\langle x_i, v_{i-1}\rangle^2 \cdot \eta\sum_{i=1}^{n}\langle x_i, w\rangle^2 && \text{by Cauchy-Schwarz} \\
&\le (\|v_n\|^2 - \|v_0\|^2)\sigma_2. && \text{by (11) and (7)}
\end{aligned}$$

There's nothing special about the start and final indices, giving the following bound for general indices $a \le b$:

$$\langle v_b - v_a, w\rangle^2 \le (\|v_b\|^2 - \|v_a\|^2)\sigma_2. \tag{12}$$

Lemma 3.2 (Growth implies correctness)**.** *For any v_0 and all i, $\|P\widehat{v}_i\| \le \sqrt{\sigma_2} + \frac{\|Pv_0\|}{\|v_i\|}$.*

Proof. By (12), for any w with $w = Pw$,

$$\langle v_i - v_0, w\rangle \le \sqrt{\sigma_2}\|v_i\|.$$

Hence

$$\langle \widehat{v}_i, w\rangle = \frac{\langle v_i - v_0, w\rangle + \langle v_0, w\rangle}{\|v_i\|} \le \sqrt{\sigma_2} + \frac{\langle v_0, w\rangle}{\|v_i\|}.$$

Setting $w = P\widehat{v}_i/\|P\widehat{v}_i\|$, we have $\langle \widehat{v}_i, w\rangle = \|P\widehat{v}_i\|$ and $\langle v_0, w\rangle \le \|Pv_0\|$, giving the result. □

Lemma 3.2 implies that, if we start at v^*, we never move by more than $\sqrt{\sigma_2}$ from it. We now show that you can't even move $\sqrt{\sigma_2}$ without increasing the norm of v.

Lemma 3.3. *Suppose $Pv_0 = 0$. For any two time steps $0 \le a < b \le n$,*

$$\|P\widehat{v}_b - P\widehat{v}_a\|^2 \le 4\sigma_2 \log \frac{\|v_b\|}{\|v_a\|}$$

Proof. Define w to be the unit vector in direction $P(\widehat{v}_b - \widehat{v}_a)$. By (12) we have

$$\langle v_b - v_a, w\rangle^2 \le \sigma_2(\|v_b\|^2 - \|v_a\|^2).$$

Therefore

$$\begin{aligned}
\|P\widehat{v}_b - P\widehat{v}_a\|^2 = \langle P(\widehat{v}_b - \widehat{v}_a), w\rangle^2 &= \langle \widehat{v}_b - \widehat{v}_a, w\rangle^2 \\
&\le 2\langle \widehat{v}_b - \frac{\|v_a\|}{\|v_b\|}\widehat{v}_a, w\rangle^2 + 2\langle \frac{\|v_a\|}{\|v_b\|}\widehat{v}_a - \widehat{v}_a, w\rangle^2 \\
&\le 2\frac{1}{\|v_b\|^2}\langle v_b - v_a, w\rangle^2 + 2(\frac{\|v_a\|}{\|v_b\|} - 1)^2\|P\widehat{v}_a\|^2 \\
&\le 2\sigma_2(1 - \frac{\|v_a\|^2}{\|v_b\|^2}) + 2(1 - \frac{\|v_a\|}{\|v_b\|})^2\sigma_2 \\
&= 4\sigma_2(1 - \frac{\|v_a\|}{\|v_b\|}).
\end{aligned}$$

Finally, $(1 - 1/x) \le \log x$ for all $x > 0$. □

3.2 Results on Sequences

The following combinatorial result is written in a self-contained fashion, independent of the streaming PCA application.

Lemma 3.4. *Let $A \in \mathbb{R}^{d\times n}$ have first column all zero. Define $b_i^{(k)}$ to be column $1+2^k i$ of A. Then:*

$$\sum_i \max_j A_{ij}^2 \le (1 + \log_2 n) \sum_{k=0}^{\log_2 n} \sum_{j>0} \|b_j^{(k)} - b_{j-1}^{(k)}\|^2$$

Proof. We will show this separately for each row i; the result is just the sum over these rows. For fixed i, let $j^* = \arg\max_j A_{ij}^2$.

Let $j^{(k)} = 1 + 2^k \lfloor \frac{j^*-1}{2^k} \rfloor$ set the last k bits of $j^* - 1$ to zero. We have that $j^{(0)} = j^*$ and $j^{\log_2 n} = 0$. Therefore

$$A_{ij^*} = \sum_{k=0}^{\log_2 n} (A_{i,j^{(k)}} - A_{i,j^{(k+1)}})$$

Now, $j^{(k)}$ is either $j^{(k+1)}$ or $j^{(k+1)} + 2^k$. Each value in the right sum is either zero (if $j^{(k)}$ is $j^{(k+1)}$) or the ith coordinate of $b_{j'}^{(k)} - b_{j'-1}^{(k)}$ for some j' (if $j^{(k)} = j^{(k+1)} + 2^k$, using $j' = j^{(k)}/2^k$). Thus, by Cauchy-Schwarz,

$$\begin{aligned} A_{ij^*}^2 &\le (1 + \log_2 n) \cdot \sum_{k=0}^{\log_2 n} (A_{i,j^{(k)}} - A_{i,j^{(k+1)}})^2 \\ &\le (1 + \log_2 n) \cdot \sum_{k=0}^{\log_2 n} \sum_{j>0} ((b_j^{(k)})_i - (b_{j-1}^{(k)})_i)^2. \end{aligned}$$

Summing over i,

$$\sum_i \max_j A_{ij}^2 \le (1 + \log_2 n) \sum_{k=0}^{\log_2 n} \sum_{j>0} \|b_j^{(k)} - b_{j-1}^{(k)}\|^2.$$

□

3.3 Proof of Growth

We return to the streaming PCA setting. The goal of this section is to show that, if $v_0 = v^*$, then $\|v_n\|$ is large.

Lemma 3.5. *If $v_0 = v^*$, then*

$$\eta \sum_{i=1}^n \langle x_i, P\widehat{v}_{i-1} \rangle^2 \lesssim \sigma_2^2 \log^2 n \log \|v_n\|$$

Proof. Define $u_i = P\widehat{v}_i$. We apply Lemma 3.4 to the matrix $A_{ij} = \langle x_i, u_{j-1} \rangle$ for $i, j \in [n]$, getting:

$$\sum_{i=1}^n \max_{j \le n-1} \langle x_i, u_j \rangle^2 \le (1 + \log_2 n) \sum_{k=0}^{\log_2 n} \sum_{j>0} \sum_{i=1}^n (\langle x_i, u_{2^k j} \rangle - \langle x_i, u_{2^k (j-1)} \rangle)^2.$$

Now,

$$\begin{aligned} \sum_{i=1}^n (\langle x_i, u_{2^k j} \rangle - \langle x_i, u_{2^k (j-1)} \rangle)^2 &= (u_{2^k j} - u_{2^k (j-1)}) X^T X (u_{2^k j} - u_{2^k (j-1)}) \\ &\le \frac{\sigma_2}{\eta} \|u_{2^k j} - u_{2^k (j-1)}\|^2. \end{aligned}$$

by the assumption (7) on X and that every $u_j \perp v^*$. Then, for each k, Lemma 3.3 shows that

$$\sum_{j>0} \|u_{2^k j} - u_{2^k (j-1)}\|^2 \le 4\sigma_2 \log \frac{\|v_n\|}{\|v_0\|} = 4\sigma_2 \log \|v_n\|$$

and thus

$$\eta \sum_i \langle x_i, P\widehat{v}_{i-1}\rangle^2 \le \eta \sum_i \max_j \langle x_i, u_j\rangle^2 \le (1+\log_2 n) \sum_{k=0}^{\log_2 n} 4\sigma_2^2 \log\|v_n\| \lesssim \sigma_2^2 \log^2 n \log\|v_n\|$$

as desired. □

Lemma 3.6 (The right direction grows.)**.** *Suppose $\sigma_2 < \frac{1}{2}$. Then if $v_0 = v^*$ we have*

$$\log\|v_n\| \gtrsim \frac{\sigma_1}{1+\sigma_2^2 \log^2 n}.$$

Proof. We will show that $\eta \sum_{i=1}^n \langle x_i, \widehat{v}_{i-1}\rangle^2 \gtrsim \sigma_1$, giving the result by (10).

Recall that $(x+y)^2 \ge \frac{1}{2}x^2 - y^2$ for all x, y. Thus, if $\widehat{v}_i = a_i v^* + u_i$ for $u_i \perp v^*$, we have

$$\langle x_i, \widehat{v}_{i-1}\rangle^2 \ge \frac{a_{i-1}^2}{2}\langle x_i, v^*\rangle^2 - \langle x_i, u_{i-1}\rangle^2.$$

Lemma 3.2 shows that $a_i^2 \ge 1 - \sigma_2 \ge \frac{1}{2}$, so summing up over i,

$$\eta \sum_{i=1}^n \langle x_i, \widehat{v}_{i-1}\rangle^2 \ge \frac{1}{4}\sigma_1 - \eta \sum_{i=1}^n \langle x_i, u_{i-1}\rangle^2.$$

Then (10) and Lemma 3.5 give

$$\log\|v_n\| \ge \frac{1}{2}\eta \sum_{i=1}^n \langle x_i, \widehat{v}_{i-1}\rangle^2 \ge \frac{1}{8}\sigma_1 - O(\sigma_2^2 \log^2 n \log\|v_n\|),$$

or

$$\log\|v_n\| \gtrsim \frac{\sigma_1}{1+\sigma_2^2 \log^2 n}.$$

□

Claim 3.7. *Let $a \sim N(0,1)$. For any two vectors u and v, with probability $1-\delta$,*

$$\|au + v\| \ge \delta\sqrt{\pi/2}\|u\|.$$

Proof. First, without loss of generality v is collinear with u; any orthogonal component only helps. So we can only consider real-valued u and v, and in fact rescale so $u = 1$. The claim is then: with probability $1-\delta$, a sample from $N(v,1)$ has absolute value at least $\delta\sqrt{\pi/2}$. This follows from the standard Gaussian density being at most $1/\sqrt{2\pi}$. □

Theorem 1.1 (Performance of Oja's method in adversarial streams)**.** *For any sufficiently large universal constant C, suppose η is such that $\eta n\lambda_1 > C\log d$ and $\eta n\lambda_2 < \frac{1}{C\log n}$. If $\eta\|x_i\|^2 \le 1$ for every i, then Oja's algorithm with learning rate η returns $\widehat{v}$ satisfying $\|P\widehat{v}\| \le \sqrt{\eta n\lambda_2} + d^{-9}$ with $1 - d^{-\Omega(C)}$ probability.*

Moreover, Oja's method can be modified (Algorithm 1) so that in addition, regardless of λ_1 and λ_2, if $\eta\|x_i\|^2 \le 1$ for all i then either $\|P\widehat{v}\| \le \sqrt{\eta n\lambda_2} + d^{-9}$ or $\widehat{v} = \perp$.

Proof. We assume that $\eta\|x_i\|^2 \le 1$ for all i, since the theorem is otherwise vacuous.

We begin with the last statement. Algorithm 1 only returns $\widehat{v} \neq \perp$ if $s_n = \log \frac{\|v_n\|}{\|v_0\|} > 10 \log d$. But then by Lemma 3.2,

$$\|P\widehat{v}_n\| \le \sqrt{\sigma_2} + \frac{\|v_0\|}{\|v_n\|} \le \sqrt{\sigma_2} + d^{-10}.$$

All that remains is to show that, if $\sigma_1 > C \log d$ and $\sigma_2 < \frac{1}{C \log n}$, $\widehat{v} \neq \perp$ with at least $1 - d^{-\Omega(C)}$ probability. And of course, $\widehat{v} \neq \perp$ if $\frac{\|v_n\|}{\|v_0\|} \ge d^{10}$.

Oja's algorithm starts with $\widehat{v}_0$ uniformly on the sphere, and is indifferent to the initial scale $\|v_0\|$, so v_0 could be constructed as $\frac{v_0}{\|v_0\|}$ for $v_0 \sim N(0, I_d)$.

Let $v_0 = av^* + u$ for $u \perp v^*$. Let $B = \prod_{i=1}^n (I + \eta x_i x_i^T)$, so $v_n = Bv_0$.

By Lemma 3.6 and the bound on σ_2, we know $\|Bv^*\| \ge e^{c'\sigma_1}$ for some constant c'. Then by Claim 3.7, with probability $1 - \delta$,

$$\|v_n\| = \|aBv^* + Bu\| \ge \delta\sqrt{\pi/2}\|Bv^*\| \ge \delta e^{c'\sigma_1}.$$

The (very naive) Markov bound from $\mathbb{E}[\|v_0\|^2] = d$ gives that

$$\frac{\|v_n\|}{\|v_0\|} \ge \frac{\delta^{3/2} e^{c'\sigma_1}}{\sqrt{d}}$$

with probability $1 - 2\delta$. For sufficiently large C in $\sigma_1 \ge C \log d$, this gives

$$\frac{\|v_n\|}{\|v_0\|} \ge d^{10}$$

with probability $1 - d^{-\Omega(C)}$. □

3.4 Proof of Theorem 1.2

Theorem 1.2 (Full algorithm)**.** *Let $X \in \mathbb{R}^{n \times d}$ have b-bit entries, so each $X_{i,j} = 0$ or $2^{-b} \le |X_{i,j}| \le 2^b$, for $b > \log(dn)$. Whenever $R > O(\log n \log d)$, Algorithm 2 uses $O(b^2 d)$ bits of space and returns $\widehat{v}$ satisfying $\|P\widehat{v}\|^2 \lesssim \frac{\log d}{R} + d^{-9}$ with high probability.*

Proof. Let C be the constant in Theorem 1.1. For R to be well defined, $\lambda_1 \neq 0$ so some $x_i \neq 0$. Therefore $2^{-2b} \le \lambda_1 \le nd^2 2^{2b}$. Thus one of the η_i considered in Algorithm 2 is such that $\eta n \lambda_1 \in [C \log d, 2C \log d]$. Let $\widehat{i}$ be this i. For sufficiently large constant in the choice of R, we have

$$\eta_i n \lambda_2 \le \frac{1}{C \log n}$$

for all $i \le \widehat{i}$.

Let $\overline{x}$ be the x_i of maximum norm, as computed by the algorithm. We now show that $\widehat{x} := \frac{\overline{x}}{\|\overline{x}\|}$ is a sufficiently good answer if $\eta_{\widehat{i}} \|\overline{x}\|^2 \ge 1$. Decompose $\overline{x} = av^* + bw$ for $w \perp v^*$ a unit vector. By (7), b is fairly small:

$$b^2 \le \eta_{\widehat{i}} \sum_i \langle x_i, w\rangle^2 \le \|PX^T XP\| = n\lambda_2 \le \frac{2C \log d}{R\eta_{\widehat{i}}}.$$

The unit vector $\widehat{x}$ in direction $\overline{x}$ has error

$$\|P\widehat{x}\|^2 = \frac{b^2}{\|\overline{x}\|^2} \le \frac{2C \log d}{R\eta_{\widehat{i}}\|\overline{x}\|^2} \lesssim \frac{\log d}{R\eta_{\widehat{i}}\|\overline{x}\|^2}. \tag{13}$$

Therefore if $\eta_{\widehat{i}}\|\overline{x}\|^2 \geq 1$, $\widehat{x}$ is a sufficiently accurate answer.

The last statement in Theorem 1.1 shows that, if $\eta_{i^*}\|\overline{x}\|^2 \leq 1$ and $i^* \leq \widehat{i}$, then

$$\|Pv^{(i^*)}\|^2 \leq (\sqrt{\eta_{i^*} n\lambda_2} + d^{-9})^2 \lesssim \eta_{i^*} n\lambda_2 + d^{-18} \leq \frac{2C\log d}{R} + d^{-18} \tag{14}$$

which is sufficiently accurate. We now split into case analysis.

In one case, suppose $\eta_{\widehat{i}}\|\overline{x}\|^2 < 1$. Therefore the main body of Theorem 1.1 states that $v^{(\widehat{i})} \neq \perp$ with high probability. In particular, this means $i^* \leq \widehat{i}$, so $\eta_{i^*}\|\overline{x}\|^2 < 1$, and the algorithm's answer is $v^{(i^*)}$ which is sufficiently accurate by (14).

Otherwise, $\eta_{\widehat{i}}\|\overline{x}\|^2 \geq 1$. Then outputting $\overline{x}$ is sufficiently accurate by (13). If $i^* \geq \widehat{i}$, the algorithm will definitely output $\overline{x}$; if $i^* < \widehat{i}$, the algorithm might output $v^{(i^*)}$, but only if $\eta_{i^*}\|\overline{x}\|^2 < 1$, in which case this is sufficiently accurate by (14). □

3.5 Precision

Finally, we discuss why $O(\log(nd))$ bits of precision suffice for the algorithm. Algorithm 1 tracks two values: a unit vector $\widehat{v}_i$ and the log-norm s_i of the unnormalized v_i. The main concern is that the error in $\widehat{v}_i$ could compound.

Consider $\widehat{v}_i$ and s_i to be the values computed by the algorithm, which has some $\varepsilon = \frac{1}{\mathrm{poly}(nd)}$ error (in ℓ_2) added in each iteration. We can enforce $s_i \geq s_{i-1}$ despite the error. Redefine v_i to $2^{s_i}\widehat{v}_i$.

We now redo the proof of (12) with ε error in each step. Define $B_i = \frac{\|v_i\|^2}{\|v_0\|^2} = 2^{s_i}$, and $A_i = \log\frac{B_i}{B_{i-1}} = (s_i - s_{i-1})$ which satisfies $A_i \geq \eta\langle x_i, \widehat{v}_{i-1}\rangle^2 - O(\varepsilon)$ by (10). Therefore

$$\eta\sum_{i=1}^{n}\langle x_i, v_{i-1}\rangle^2 \leq \|v_0\|^2\sum_{i=1}^{n}(A_i + O(\varepsilon))B_{i-1} \leq \|v_0\|^2((B_n - 1) + O(\varepsilon n B_n)) = \|v_n\|^2 - \|v_0\|^2 + O(\varepsilon n\|v_n\|^2) \tag{15}$$

by Claim 3.1. Then for any unit vector w with $Pw = w$,

$$\begin{aligned}
\langle v_n - v_0, w\rangle^2 &= (\sum_{i=1}^{n}\langle v_i - v_{i-1}, w\rangle)^2 \\
&= (\sum_{i=1}^{n}\eta\langle x_i, v_{i-1}\rangle\langle x_i, w\rangle + O(\varepsilon)\|v_{i-1}\|)^2 \\
&= \left(O(n\varepsilon\|v_n\|) + \eta\sum_{i=1}^{n}\langle x_i, v_{i-1}\rangle\langle x_i, w\rangle\right)^2 && \text{by (8)} \\
&\leq \eta\sum_{i=1}^{n}\langle x_i, v_{i-1}\rangle^2 \cdot \eta\sum_{i=1}^{n}\langle x_i, w\rangle^2 + O(n^2\varepsilon\|v_n\|^2) && \text{by Cauchy-Schwarz} \\
&\leq (\|v_n\|^2 - \|v_0\|^2)\sigma_2 + O(\varepsilon n^2\|v_n\|^2). && \text{by (15) and (7)}
\end{aligned}$$

There's nothing special about the start and final indices, giving the following bound for general indices $a \leq b$:

$$\langle v_b - v_a, w\rangle^2 \leq (\|v_b\|^2 - \|v_a\|^2)\sigma_2 + O(\varepsilon n^2\|v_b\|^2). \tag{16}$$

Given (16), the error tolerance flows through the rest of the proof easily. Lemmas 3.2 and 3.3 follow immediately with $O(\varepsilon n^2)$ additive error. Lemma 3.5 gets additive error $O(\sigma_2 \varepsilon n^3 \log^2 n)$, so both the numerator and denominator of Lemma 3.6 change by $\varepsilon \operatorname{poly}(n)$. Both the conditions and result of Theorem 1.1 only change by an additive $\varepsilon \operatorname{poly}(n)$ error, which for sufficiently small polynomial ε are absorbed by the constant factors and $\frac{1}{d^9}$ additive error. And Algorithm 2 does nothing that could compound the error by more than a constant factor, so Theorem 1.2 holds as well.

4 Lower Bound

Our lower bound is based on the PartialDuplicate instance we define here:

Definition 4.1. *The* PartialDuplicate *instance is defined as follows:* $X \in \{\pm 1\}^{n+k \times d}$ *uniformly at random, except that the first* k *rows match on the first* $(1-\gamma)d$ *coordinates.*

4.1 Spectral properties of PartialDuplicate

We can express a PartialDuplicate instance as follows:

- For $i \in [k]$, $x_i = \overline{x} + y_i$ where $\overline{x}, y_i \in \{0, -1, 1\}^d$ have $\operatorname{supp}(x) = \{1, 2, \ldots, (1-\gamma)d\}$ and $\operatorname{supp}(y_i) = \{(1-\gamma)d + 1, \ldots, d\}$.
- $X' \in \{-1, 1\}^{n \times d}$ consists of the last n rows of X.

That is, the entries look like:

$$X = \begin{array}{|c|c|}\hline \overline{x} & y_1 \\ \vdots & \vdots \\ \overline{x} & y_k \\ \hline \multicolumn{2}{|c|}{X'} \\ \hline \end{array}$$

except that $\overline{x}, y_i$ are zero-padded to d dimensions. The nonzero entries of $\overline{x}$, the y_i, and X' are all independent. Let $Y = (y_1, \ldots, y_k)^T \in \{-1, 0, 1\}^{k \times d}$, and let $\widetilde{Y} \in \mathbb{R}^{k \times \gamma d}$ contain the nonzero columns of Y.

We can decompose any unit vector w into three components: the $\overline{x}$ direction, the $\sum_{i=1}^k y_i$ direction, and the component orthogonal to both of these. This is:

$$w = a\frac{\overline{x}}{\sqrt{(1-\gamma)d}} + b\frac{\sum y_i}{\|\sum y_i\|} + cw'$$

where $a^2 + b^2 + c^2 = 1$ and w' is a unit vector orthogonal to $\overline{x}$ and $\sum y_i$.

We have that

$$\begin{aligned} \|Xw\|^2 &= \|X'w\|^2 + \sum_{i=1}^k \langle w, \overline{x} + y_i\rangle^2 \\ &= \|X'w\|^2 + \sum_{i=1}^k (a\sqrt{(1-\gamma)d} + \langle w, y_i\rangle)^2 \\ &= \|X'w\|^2 + ka^2(1-\gamma)d + 2ab\sqrt{(1-\gamma)d}\langle w, \sum_{i=1}^k y_i\rangle + \|Yw\|^2 \end{aligned}$$

Define

$$\Delta_w := \frac{1}{\sqrt{d}}\langle w, \sum_{i=1}^{k} y_i\rangle,$$

so we have

$$\frac{1}{d}\|Xw\|^2 = ka^2(1-\gamma) + 2ab\sqrt{1-\gamma}\Delta_w + \frac{1}{d}\|X'w\|^2 + \frac{1}{d}\|Yw\|^2. \tag{17}$$

We now give upper and lower bounds on how large $\frac{1}{d}\|Xw\|^2$ can be. For any (γ, k) define

$$C_{\gamma,k} := \frac{1}{2}\left(k(1-\gamma) + \sqrt{(k(1-\gamma))^2 + 4(1-\gamma)\gamma k}\right).$$

This satisfies $(1-\gamma)k \le C_{\gamma,k} \le (1-\gamma)k + \gamma$.

Our lemmas will be "with high probability in d", meaning at least $1 - d^{-C}$ probability for an arbitrary constant C, and will involve an $o_d(1)$ term that is polynomial in k, γ, C and inverse polynomial in d.

Lemma 4.2. *For $d \ge k^3$, with high probability in d, there exists a unit vector w^* with*

$$\frac{1}{d}\|Xw^*\|^2 \ge C_{\gamma,k} - o_d(1).$$

Proof. Consider $w^* = a\frac{\overline{x}}{\sqrt{(1-\gamma)d}} + b\frac{\sum y_i}{\|\sum y_i\|}$ for a, b with $a^2 + b^2 = 1$ to be chosen later. By (17),

$$\frac{1}{d}\|Xw\|^2 \ge a^2k(1-\gamma) + 2ab\sqrt{1-\gamma}\Delta_{w^*}$$

So with a little algebra (Claim A.4) there exists a choice of a, b such that:

$$\frac{1}{d}\|Xw^*\|^2 \ge \frac{1}{2}\left(k(1-\gamma) + \sqrt{(k(1-\gamma))^2 + 4(1-\gamma)\Delta_{w^*}^2}\right).$$

For $0 \le b \le a$, we have $\sqrt{a-b} \ge \sqrt{a} - \sqrt{b}$. Thus

$$\frac{1}{d}\|Xw^*\|^2 \ge C_{\gamma,k} - \sqrt{(1-\gamma)\max(0, \gamma k - \Delta_{w^*}^2)} \ge C_{\gamma,k} - \sqrt{|\gamma k - \Delta_{w^*}^2|}$$

Now, $\Delta_{w^*}^2 = \frac{1}{d}\|\sum_{i=1}^k y_i\|^2$. This has $\mathbb{E}[\Delta_{w^*}^2] = \gamma k$, and it concentrates quite well: the first $(1-\gamma)d$ coordinates of $\sum y_i$ are zero, and the rest are $\sqrt{k}\widetilde{Y}^T u$ for $u = \frac{1}{\sqrt{k}}(1, \dots, 1) \in \mathbb{R}^k$. Hence by the JL Lemma (Claim A.2) applied to $\widetilde{Y}, u$, and$(n, d) = (\gamma d, k)$ with high probability we have

$$|\frac{1}{k}\|\sum_{i=1}^{k} y_i\|^2 - \gamma d| \lesssim \sqrt{\gamma d \log d} + \log d$$

or

$$|\Delta_{w^*}^2 - \gamma k| \lesssim k\sqrt{\frac{\gamma \log d}{d}} + \frac{k}{d}\log d = o_d(1)$$

Thus

$$\frac{1}{d}\|Xw^*\|^2 \ge C_{\gamma,k} - o_d(1).$$

□

We now show that every $w \perp y_1$ has smaller $\|Xw\|^2$.

Lemma 4.3. *Suppose that $k(1-\gamma) \geq 1+\gamma$.*
With high probability in d, every unit vector w has

$$\frac{1}{d}\|Xw\|^2 \leq C_{\gamma,k} + O(\sqrt{\lambda}) - \frac{\gamma}{2k} + 2\frac{|\langle w, y_1\rangle|}{\sqrt{kd}}$$

Proof. We would like to bound the terms in (17).

We first consider $\|X'w\|$. The maximum singular value of X is approximately $\sqrt{\lambda d} + \sqrt{d}$. In particular, by [FS10] (see Lemma A.1), the maximum singular value is at most $2\sqrt{\lambda d} + \sqrt{d}$ with high probability. Suppose that happens.

For any fixed unit vector u independent of X', as a distribution over X', Claim A.2 says that $\|X'u\|^2$ has expectation $n = \lambda d$ and with high probability,

$$|\|X'u\|^2 - \lambda d| \lesssim \sqrt{\lambda d \log d} + \log d \ll d.$$

Thus

$$\frac{1}{d}\|X'u\|^2 \leq \lambda + o_d(1)$$

with high probability. Suppose this happens to both the unit vector in direction $\overline{x}$ and the one in direction $\sum_i y_i$. Then

$$\begin{aligned}\frac{1}{\sqrt{d}}\|X'w\| &\leq \frac{1}{\sqrt{d}}\left(|a|\|X'\frac{\overline{x}}{\|\overline{x}\|}\| + |b|\|X'\frac{\sum y_i}{\|\sum y_i\|}\| + |c|\|X'w'\|\right)\\ &\leq |a|\sqrt{\lambda} + |b|\sqrt{\lambda} + o_d(1) + |c|(2\sqrt{\lambda}+1)\\ &\leq O(\sqrt{\lambda}) + |c|\end{aligned}$$

and hence

$$\frac{1}{d}\|X'w\|^2 \leq c^2 + O(\sqrt{\lambda}) \tag{18}$$

We next consider $\|Yw\|$. $\widetilde{Y}$ is a $k \times (\gamma d)$ matrix with independent ± 1 entries, so by [FS10] (see Lemma A.1), with high probability its top singular value is $\sqrt{k} + \sqrt{\gamma d} + o(\sqrt{d}) \leq (\sqrt{\gamma} + o(1))\sqrt{d}$, so the same is true for Y. Since $Y\overline{x} = 0$, we have

$$\frac{1}{d}\|Yw\|^2 \leq \frac{1}{d}\|Y\|^2(1-a^2) \leq (1-a^2)(\gamma + o_d(1)). \tag{19}$$

For large enough d we have that this $o_d(1) \leq \sqrt{\lambda}$.

Plugging into (17) we have

$$\begin{aligned}\frac{1}{d}\|Xw\|^2 &\leq ka^2(1-\gamma) + 2ab\sqrt{1-\gamma}\Delta_w + c^2 + O(\sqrt{\lambda}) + (1-a^2)\gamma\\ &= \gamma + O(\sqrt{\lambda}) + c^2 + (k(1-\gamma)-\gamma)a^2 + ab\cdot 2\sqrt{1-\gamma}\Delta_w\end{aligned}$$

Since $a^2+b^2=1-c^2$, by Claim A.4 this satisfies

$$\frac{1}{d}\|Xw\|^2 \le O(\sqrt{\lambda}) + c^2 + \gamma + (1-c^2)\underbrace{\frac{1}{2}\left(k(1-\gamma)-\gamma+\sqrt{(k(1-\gamma)-\gamma)^2+4(1-\gamma)\Delta_w^2}\right)}_{C}$$

Now, $C \ge k(1-\gamma)-\gamma \ge 1$, so this expression is maximized when $c=0$, giving:

$$\frac{1}{d}\|Xw\|^2 \le O(\sqrt{\lambda}) + \underbrace{\frac{1}{2}\left(k(1-\gamma)+\gamma+\sqrt{(k(1-\gamma)-\gamma)^2+4(1-\gamma)\Delta_w^2}\right)}_{C'} \tag{20}$$

We now relate the term C' to $C_{\gamma,k}$. First,

$$(k(1-\gamma)-\gamma)^2+4(1-\gamma)\Delta_w^2 = \left[(k(1-\gamma))^2+4(1-\gamma)\gamma k\right] - 6\gamma(1-\gamma)k+\gamma^2+4(1-\gamma)\Delta_w^2.$$

Since $\sqrt{a+b} \le \sqrt{a}+\frac{b}{2\sqrt{a}}$ for $a>0$ and any b, this means

$$\begin{aligned} C' &\le \frac{\gamma}{2} + C_{\gamma,k} + \frac{4(1-\gamma)\Delta_w^2 - 6\gamma(1-\gamma)k+\gamma^2}{4k(1-\gamma)} \\ &= C_{\gamma,k} - \gamma + \frac{\Delta_w^2}{k} + \frac{\gamma^2}{4k(1-\gamma)} \end{aligned}$$

Thus plugging back into (20),

$$\frac{1}{d}\|Xw\|^2 \le O(\sqrt{\lambda}) + C_{\gamma,k} - \gamma + \frac{\Delta_w^2}{k} + \frac{\gamma^2}{4k(1-\gamma)} \tag{21}$$

We now relate Δ_w^2 to $\langle w, y_1\rangle$. We know that

$$|\langle w, \sum y_i\rangle| = |\langle w, \sum_{i>1} y_i\rangle + \langle w, y_1\rangle| \le \|\sum_{i>1} y_i\| + |\langle w, y_1\rangle|$$

and (as in the JL lemma, Claim A.2), with high probability $\|\sum_{i>1} y_i\| = \sqrt{(k-1)\gamma d} + o(\sqrt{d})$. Thus

$$|\Delta_w| = \frac{1}{\sqrt{d}}|\langle w, \sum y_i\rangle| \le \sqrt{(k-1)\gamma} + o(1) + \frac{|\langle w, y_1\rangle|}{\sqrt{d}}$$

and so

$$\Delta_w^2 \le (k-1)\gamma + 2\sqrt{(k-1)\gamma}\frac{|\langle w, y_1\rangle|}{\sqrt{d}} + \frac{\langle w, y_1\rangle^2}{d} + o(1)$$

Since $\frac{|\langle w, y_1\rangle|}{\sqrt{d}} \le \sqrt{\gamma}$, this means

$$\Delta_w^2 \le (k-1)\gamma + (2\sqrt{(k-1)\gamma}+\gamma)\frac{|\langle w, y_1\rangle|}{\sqrt{d}} + o(1) \le (k-1)\gamma + 2\sqrt{k}\frac{|\langle w, y_1\rangle|}{\sqrt{d}} + o(1).$$

Plugging back into (21), we have

$$\frac{1}{d}\|Xw\|^2 \le O(\sqrt{\lambda}) + C_{\gamma,k} - \frac{\gamma}{k}\left(1 - \frac{\gamma}{4(1-\gamma)}\right) + 2\frac{|\langle w, y_1\rangle|}{\sqrt{kd}}$$

or

$$\frac{1}{d}\|Xw\|^2 \le C_{\gamma,k} + O(\sqrt{\lambda}) - \frac{\gamma}{2k} + 2\frac{|\langle w, y_1\rangle|}{\sqrt{kd}}$$

as desired. □

Lemma 4.4. *Suppose that $k(1-\gamma) \geq 1+\gamma$, $\gamma < \frac{1}{2}$, and $\lambda \leq c(\frac{\gamma}{k})^2$ and $\varepsilon \leq c\frac{\gamma}{k^2}$ for a sufficiently small constant c. For sufficiently large $d > \text{poly}(k/\gamma)$, any ε-approximate PCA solution w must have $\frac{|\langle w, y_1\rangle|}{\sqrt{\gamma d}} \geq \frac{\sqrt{\gamma}}{8\sqrt{k}}$ with high probability.*

Proof. Let the top singular vector of X be v^*. Then any ε-approximate PCA solution w has $w = \sqrt{1-a}v^* + aw'$ for a unit vector $w' \perp v^*$ and $0 \leq a \leq \varepsilon$. Hence

$$\|Xw\| \geq \sqrt{1-a}\|Xv^*\| - a\|Xw'\| \geq (\sqrt{1-a} - a)\|X\| \geq (1 - \frac{3}{2}a)\|X\| \geq (1 - \frac{3}{2}\varepsilon)\|X\|.$$

By Lemma 4.2, this means

$$\frac{1}{d}\|Xw\|^2 \geq (1 - \frac{3}{2}\varepsilon)^2 C_{\gamma,k} - o(1).$$

Now, $C_{\gamma,k} \leq k$, so

$$\frac{1}{d}\|Xw\|^2 \geq C_{\gamma,k} - 3\varepsilon k - o(1)$$

By Lemma 4.3, this means

$$O(\sqrt{\lambda}) - \frac{\gamma}{2k} + 2\frac{|\langle w, y_1\rangle|}{\sqrt{kd}} \geq -3\varepsilon k - o(1)$$

or

$$2\frac{|\langle w, y_1\rangle|}{\sqrt{kd}} \geq \frac{\gamma}{2k} - 3\varepsilon k - o(1) - O(\sqrt{\lambda})$$

For $\varepsilon < \frac{\gamma}{24k^2}$ and $\sqrt{\lambda} < c\frac{\gamma}{k}$ for sufficiently small constant c this gives

$$2\frac{|\langle w, y_1\rangle|}{\sqrt{kd}} \geq \frac{\gamma}{4k}$$

and hence

$$\frac{|\langle w, y_1\rangle|}{\sqrt{\gamma d}} \geq \frac{\sqrt{\gamma}}{8\sqrt{k}}$$

□

Lemma 4.5. *For $\gamma \leq \frac{1}{4}$, $\lambda \leq \frac{1}{10}$, and sufficiently large $d > O(k)$, the spectral gap R is at least $\frac{k}{4}$ with high probability.*

Proof. By Lemma 4.2, with high probability $\frac{1}{d}\|X\|^2 \geq C_{\gamma,k} - o_d(1) \geq (1-\gamma)k - o_d(1) \geq k/2$.

The second eigenvalue λ_2 of $\Sigma = X^T X$ satisfies

$$\begin{aligned}
\lambda_2 &= \min_{v} \max_{\substack{v' \perp v \\ \|v\|=1}} \|Xv'\|^2 \\
&\leq \max_{\substack{v' \perp \overline{x} \\ \|v\|=1}} \|Xv'\|^2 \\
&= \max_{\substack{v' \perp \overline{x} \\ \|v\|=1}} \left(\|X'v'\|^2 + \|Yv'\|^2\right) \\
&\leq \|X'\|^2 + \|Y\|^2
\end{aligned}$$

By [FS10] (see Lemma A.1), with high probability, $\|X'\| \leq \sqrt{n} + \sqrt{d} + o(\sqrt{d})$ and $\|Y\| \leq \sqrt{k} + \sqrt{\gamma d} + o(\sqrt{d})$, so

$$\frac{1}{d}\lambda_2 \leq (1 + \sqrt{\lambda} + o(1))^2 + (\sqrt{\gamma} + o(1))^2 \leq 1 + 2\sqrt{\lambda} + \lambda + \gamma + o(1) \leq 2.$$

Hence with high probability, the spectral ratio

$$R = \frac{\|\Sigma\|}{\lambda_2} = \frac{\|X\|^2}{\lambda_2} \geq \frac{k}{4}.$$

□

4.2 Streaming lower bound

Theorem 1.3 (Lower bound)**.** *There exists a universal constant $C > 1$ such that: for any $R > 1$, $\frac{1}{CR^2}$-approximate PCA on streams with spectral gap R requires at least $\frac{d^2}{CR^3}$ bits of space for sufficiently large $d > \mathrm{poly}(R)$.*

The polynomial dependence on R in our proof has not been optimized.

Proof. Suppose that we have such an ε-approximate streaming PCA algorithm. We set up a two player one-way communication protocol. Let $A_1 \in \{-1,1\}^{n\times(1-\gamma)d}$ and $A_2 \in \{-1,1\}^{n\times\gamma d}$ be chosen uniformly at random. Let $A = [A_1, A_2] \in \{-1,1\}^{n\times d}$ be their concatenation.

In this protocol, Alice receives $A = [A_1, A_2]$ and Bob receives A_1. Alice feeds A to the streaming algorithm, reaching some stream state S, which she sends to Bob. Bob uses A_1 and S to construct an approximation $\widehat{A}$ to A_2 in the following fashion. For each row $i \in [n]$, Bob chooses $k-1$ vectors $u_1, \ldots, u_{k-1}$ that match the ith row of A_1 on the first $(1-\gamma)d$ coordinates, and are independently uniformly drawn from $\{-1,1\}^n$ on the remaining γd coordinates. Bob sets the streaming algorithm's state to S, inserts $u_1, \ldots, u_{k-1}$, and computes the algorithm's approximate PCA solution $\widehat{v}_i$. He does this for each $i \in [n]$, constructing a matrix $\widehat{V} \in \mathbb{R}^{n\times d}$. Let $\widehat{V}_2 \in R^{n\times\gamma d}$ be the last γd columns of $\widehat{V}$. We will show that $I(A_2; \widehat{V}) \gtrsim d^2$ for appropriate choice of parameters.

Note that when Bob produces $\widehat{v}_i$, the streaming algorithm has effectively seen the stream A followed by $k-1$ vectors that match the ith row of A. Up to reordering of rows, this is distributed identically to PartialDuplicate for $n' = n-1$. Reordering the rows, of course, does not change the covariance matrix.

We choose $\gamma = \frac{1}{4}$, $k = 4R$, $n = \lambda d$ for $\lambda = c(\gamma/k)^2$ and $\varepsilon = c\frac{\gamma}{k^2}$ for the constant c in Lemma 4.4, and require $d > \mathrm{poly}(k/\gamma)$ for a sufficiently large polynomial such that Lemma 4.4 applies.

By Lemma 4.5, with high probability the stream has spectral gap at least $k/4 \geq R$. Therefore the streaming algorithm's PCA solution should be ε-approximate with at least $2/3$ probability. But then by Lemma 4.4, where we pick y_1 to have the ith row a_i of A_2, we have

$$\frac{|\langle \widehat{v}_i, a_i\rangle|}{\sqrt{\gamma d}} \geq \frac{\sqrt{\gamma}}{8\sqrt{k}}$$

with at least $2/3 - \frac{1}{\mathrm{poly}(d)} > 1/2$ probability. Then Lemma A.3 says that

$$I(\widehat{V}; A_2) \geq \Omega((\frac{\sqrt{\gamma}}{8\sqrt{k}})^2 \cdot n \cdot \gamma d) - n = \Omega(\frac{\gamma^2 nd}{k}) - n = d^2 \cdot \Omega(\frac{\lambda}{k}) = d^2 \cdot \Omega(1/R^3).$$

Now, $\widehat{V}$ is independent of A_2 conditioned on (S, A_1) so by the data processing inequality,

$$I(\widehat{V}; A_2) \leq I(A_1, S; A_2) \leq I(A_1; A_2) + I(S; A_2 \mid A_1) \leq 0 + H(S).$$

Thus, if the state S contains $|S|$ bits, we have

$$\Omega(d^2/R^3) \leq H(S) = H(|S|) + H(S \mid |S|) \leq \mathbb{E}[|S|] + H(|S|)$$

Now, for any random variable X over positive integers,

$$\begin{aligned} H(X) &= \sum_{i=1}^{\infty} p(i) \log \frac{1}{p(i)} \\ &= \left(\sum_{i:p(i)\le 2^{-i}} p(i) \log \frac{1}{p(i)} \right) + \left(\sum_{i:p(i)>2^{-i}} p(i) \log \frac{1}{p(i)} \right) \\ &\le \left(\sum_{i:p(i)\le 2^{-i}} 2^{-i} \cdot i \right) + \left(\sum_{i:p(i)>2^{-i}} ip(i) \right) \\ &= 2 + \mathbb{E}[X] \end{aligned}$$

so $\Omega(d^2/R^3) \le 2\,\mathbb{E}[|S|] + 2$, or

$$\mathbb{E}[|S|] \ge \Omega(d^2/R^3).$$

Thus the streaming algorithm must store $\Omega(d^2/R^3)$ bits on average after Alice has finished feeding in her part of the stream. □

Acknowledgments

We appreciate the many helpful comments of anonymous reviewers on a prior version of this paper.

Bibliography

[ACLS12] Raman Arora, Andrew Cotter, Karen Livescu, and Nathan Srebro. Stochastic optimization for PCA and PLS. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 861–868. IEEE, 2012.

[AL17] Zeyuan Allen-Zhu and Yuanzhi Li. First efficient convergence for streaming k-PCA: a global, gap-free, and near-optimal rate. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 487–492. IEEE, 2017.

[BDF13] Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental PCA. *Advances in neural information processing systems*, 26, 2013.

[BDWY16] Maria-Florina Balcan, Simon Shaolei Du, Yining Wang, and Adams Wei Yu. An improved gap-dependency analysis of the noisy power method. In *Conference on Learning Theory*, pages 284–309. PMLR, 2016.

[BWZ16] Christos Boutsidis, David P Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 236–249, 2016.

[CW09] Kenneth L Clarkson and David P Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 205–214, 2009.

[FS10] Ohad N Feldheim and Sasha Sodin. A universality result for the smallest eigenvalues of certain sample covariance matrices. *Geometric And Functional Analysis*, 20(1):88–123, 2010.

[GLPW16] Mina Ghashami, Edo Liberty, Jeff M Phillips, and David P Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

[HNWTW21] De Huang, Jonathan Niles-Weed, Joel A Tropp, and Rachel Ward. Matrix concentration for products. *Foundations of Computational Mathematics*, pages 1–33, 2021.

[HNWW21] De Huang, Jonathan Niles-Weed, and Rachel Ward. Streaming k-PCA: Efficient guarantees for Oja's algorithm, beyond rank-one updates. In *Conference on Learning Theory*, pages 2463–2498. PMLR, 2021.

[HP14] Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27, 2014.

[JJK+16] Prateek Jain, Chi Jin, Sham M Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming PCA: Matching matrix bernstein and near-optimal finite sample guarantees for Oja's algorithm. In *Conference on learning theory*, pages 1147–1164. PMLR, 2016.

[Lib13] Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–588, 2013.

[LSW21] Robert Lunde, Purnamrita Sarkar, and Rachel Ward. Bootstrapping the error of oja's algorithm. *Advances in Neural Information Processing Systems*, 34:6240–6252, 2021.

[MCJ13] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory limited, streaming PCA. *Advances in neural information processing systems*, 26, 2013.

[Oja82] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

[RV10] Mark Rudelson and Roman Vershynin. Non-asymptotic theory of random matrices: extreme singular values. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*, pages 1576–1602. World Scientific, 2010.

[TYUC17] Joel A Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Practical sketching algorithms for low-rank matrix approximation. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1454–1485, 2017.

[Upa18] Jalaj Upadhyay. Fast and space-optimal low-rank factorization in the streaming model with application in differential privacy. *NeurIPS*, 2018.

[Woo14a] David Woodruff. Low rank approximation lower bounds in row-update streams. *Advances in neural information processing systems*, 27, 2014.

[Woo14b] David P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

A Utility lemmas for the Lower Bound

We use the following bound on the maximum singular value of an iid subgaussian matrix:

Lemma A.1 (Feldheim and Sodin [FS10], see also (2.4) of [RV10])**.** *Let A be an $n \times N$ random matrix with independent subgaussian entries of zero mean and variance 1, for $n \leq N$. There exists a universal constant $c > 0$ such that*

$$\Pr[\|A\| \geq \sqrt{n} + \sqrt{N} + \tau\sqrt{N}] \lesssim e^{-cn\tau^{3/2}}$$

for any $\tau > 0$.

The following is essentially a restatement of the JL lemma for ± 1 matrices:

Claim A.2. *Let $u \in \mathbb{R}^d$ be a unit vector, and $X \in \{-1, 1\}^{n \times d}$ independently and uniformly. Then*

$$\mathbb{E}[\|Xu\|^2] = n$$

and with $1 - \delta$ probability

$$|\|Xu\|^2 - n| \lesssim \sqrt{n \log \frac{1}{\delta}} + \log \frac{1}{\delta}.$$

Proof. Let $z = Xu$. The coordinates z_i are independent, mean zero, variance 1, and subgaussian with variance parameter 1. The expectation bound is trivial: sum the variance over n independent coordinates. For concentration, each coordinate z_i^2 is a squared subgaussian, and hence subgamma with (σ, c) parameters $(O(1), O(1))$. Then $\sum_i z_i^2$ is subgamma with parameters $(O(\sqrt{n}), O(1))$. Hence with probability $1 - \delta$ we have

$$|\|Xu\|^2 - n| \lesssim \sqrt{n \log \frac{1}{\delta}} + \log \frac{1}{\delta}.$$

□

Lemma A.3. *Let $X \in \{-1, 1\}^{n \times d}$ be uniformly distributed, and let $Y \in \mathbb{R}^{n \times d}$ have rows of norm at most 1 such that each row $i \in [n]$ has $|\langle x_i, y_i \rangle| > a\sqrt{d}$ with at least 50% probability, for $a > 0$. Then*

$$I(X; Y) \geq \Omega(a^2 nd) - n.$$

Proof. For any row y, when $x \in \{-1, 1\}^d$ uniformly at random, $\langle x, y \rangle$ is subgaussian with variance parameter $\|y\|^2 \leq 1$, so

$$\Pr[|\langle x, y \rangle| > a\sqrt{d}] \leq 2e^{-a^2 d/2},$$

so the number of x with $|\langle x, y \rangle| > a\sqrt{d}$ is at most $2^{(1-\Omega(a^2))d}$. Let $b \in \{0, 1\}^n$ denote the indicator vector with $b_i = 1$ if $|\langle x_i, y_i \rangle| > a\sqrt{d}$ and $b_i = 0$ otherwise.

For any Y, b, let $S_{Y,b} \subseteq \{-1, 1\}^{n \times d}$ be the set of possible X that satisfy the inner product condition $|\langle x_i, y_i \rangle| > a\sqrt{d}$ for all rows $i \in [n]$ with $b_i = 1$. Each row with $b_i = 1$ has at most $2^{(1-\Omega(a^2))d}$ values of x_i in the support, so

$$|S_{Y,b}| \leq 2^{nd - \Omega(a^2 \|b\|_1 d)}.$$

We have $\mathbb{E}[\|b\|_1] \geq \frac{n}{2}$, so

$$H(X \mid Y) \leq H(X \mid Y, b) + H(b) \leq (\mathop{\mathbb{E}}_{Y,b} \log |S_{Y,b}|) + n \leq (1 - \Omega(\frac{1}{2}a^2))nd + n$$

so

$$I(X; Y) = H(X) - H(X \mid Y) \geq \Omega(a^2 nd) - n.$$

□

Claim A.4. *Let $A, B > 0$. Then*

$$Aa^2 + Bab \leq \frac{a^2+b^2}{2}(A + \sqrt{A^2+B^2}),$$

with equality if $\frac{a^2}{a^2+b^2} = \frac{1+\sqrt{\frac{A^2}{A^2+B^2}}}{2}$.

Proof. Just ask a computer. By hand, though: the equations are homogeneous, so WLOG we can assume $a^2 + b^2 = 1$. We then maximize over $a \in [0, 1]$. Taking the derivative, the maximum is achieved when

$$2Aa + B(\sqrt{1-a^2} - \frac{a^2}{\sqrt{1-a^2}}) = 0$$

or

$$\begin{aligned} 2Aa\sqrt{1-a^2} &= B(2a^2-1) \\ 4A^2a^2(1-a^2) &= B^2(4a^4 - 4a^2 + 1) \\ a^4(4B^2+4A^2) - a^2(4A^2+4B^2) + B^2 &= 0 \\ a^2 &= \frac{1 \pm \sqrt{\frac{A^2}{A^2+B^2}}}{2} \end{aligned}$$

the first squaring preserved equality only when $a^2 \geq \frac{1}{2}$, so the optimum is at

$$a^2 = \frac{1+\sqrt{\frac{A^2}{A^2+B^2}}}{2}.$$

Then

$$\begin{aligned} Aa^2 + Ba\sqrt{1-a^2} &= A\frac{1+\sqrt{\frac{A^2}{A^2+B^2}}}{2} + B\sqrt{\frac{1+\sqrt{\frac{A^2}{A^2+B^2}}}{2}\frac{1-\sqrt{\frac{A^2}{A^2+B^2}}}{2}} \\ &= A\frac{1+\sqrt{\frac{A^2}{A^2+B^2}}}{2} + B\sqrt{\frac{\frac{B^2}{A^2+B^2}}{4}} \\ &= \frac{1}{2}(A + \sqrt{A^2+B^2}). \end{aligned}$$

□