JMLR: Workshop and Conference Proceedings vol 40:1–12, 2015

Online PCA with Spectral Bounds

Zohar Karnin ZKARNIN@YAHOO-INC.COM
Yahoo Labs, Haifa, Israel

Edo Liberty EDO@YAHOO-INC.COM
Yahoo Labs, New York, NY

Abstract

This paper revisits the online PCA problem. Given a stream of n vectors $x_t \in \mathbb{R}^d$ (columns of X) the algorithm must output $y_t \in \mathbb{R}^\ell$ (columns of Y) before receiving x_{t+1}. The goal of online PCA is to simultaneously minimize the target dimension ℓ and the error $\|X - (XY^+)Y\|^2$. We describe two simple and deterministic algorithms. The first, receives a parameter Δ and guarantees that $\|X - (XY^+)Y\|^2$ is not significantly larger than Δ. It requires a target dimension of $\ell = O(k/\varepsilon)$ for any k, ε such that $\Delta \geq \varepsilon\sigma_1^2 + \sigma_{k+1}^2$, with σ_i being the i'th singular value of X. The second receives k and ε and guarantees that $\|X-(XY^+)Y\|^2 \leq \varepsilon\sigma_1^2+\sigma_{k+1}^2$. It requires a target dimension of $O(k \log n/\varepsilon^2)$. Different models and algorithms for Online PCA were considered in the past. This is the first that achieves a bound on the spectral norm of the residual matrix.

Keywords: Online, PCA, SVD, Principal Component Analysis, Dimension Reduction

1. Introduction

Principal Component Analysis (PCA) is an algebraic technique used for countless purposes, across multiple fields of study. Its importance for scientific computing, statistics, engineering and computer science cannot be overstated. Among others, it is used for statistical inference, dimension reduction, factor analysis, signal processing, topic modeling, and visualization. A convenient definition of it, for our setup, is achieved by viewing it as an optimization problem in the context of dimension reduction. PCA can be seen as minimizing an objective function describing a reconstruction error. Given a matrix $X \in \mathbb{R}^{d\times n}$ with n columns consisting of d-dimensional vectors, compute a matrix $Y \in \mathbb{R}^{k\times n}$ whose columns reside in a low dimensional space $k \ll d$ minimizing

$$\|X - (XY^+)Y\|_F^2 \quad \text{or} \quad \|X - (XY^+)Y\|^2 .$$

Here, and throughout, A^+ stands for the Moore Penrose inverse or pseudo-inverse of A, $\|A\|_F = (\sum_{ij} A_{ij}^2)^{1/2}$ its Frobenius norm and $\|A\| = \max_{x\neq 0} \|Ax\|/\|x\|$ its spectral norm. Recall that if σ_i denotes the i'th singular value of A then $\|A\| = \sigma_1$ and $\|A\|_F = \sqrt{\sum_i \sigma_i^2}$. It is well known that a truncated Singular Value Decomposition (SVD) of X can solve both problems simultaneously. Namely, let Q denote the matrix whose columns are the k left singular vectors of X corresponding to its largest singular values. Then, setting $Y = Q^T X$ simultaneously gives the optimal solution for both objective functions. Given the importance of this problem, a significant amount of research was dedicated to reducing the complexity of obtaining a good approximation to Q in one pass by Frieze et al. (1998); Drineas and Kannan (2003); Deshpande and Vempala (2006); Sarlós (2006); Rudelson and Vershynin (2007); Liberty et al. (2007); Liberty (2013); Ghashami and Phillips (2014b). Yet, even when Q is computed (or approximated) in one pass, a second pass is needed to produce the reduced dimension matrix Y, that is, to compute $y_t \leftarrow Q^T x_t$. Here x_t and y_t correspond to the columns of X and Y respectively.

1.1. Online PCA with Frobenius Norm Bounds

Several authors investigated online PCA with respect to the Frobenius norm of the residual. Recently, Arora et al. (2013); Mitliagkas et al. (2013) and Balsubramani et al. (2013) investigated the stochastic model where x_t are assumed to be drawn from the same (unknown) distribution. This is a natural assumption in machine learning, for example, but uncommon in numerical linear algebra and in the literature of online algorithms as a whole. Warmuth and Kuzmin (2007) and Nie et al. (2013) considered the general adversarial case but their definition of online PCA is a quite different than ours. At each point in time they commit to a rank k projection P_t before observing x_t. Their cost function incurs a cost of $\|(I - P_t)x_t\|^2$. Unfortunately, this kind of result cannot be converted to one that outputs y_t (along the way) with reconstruction guarantees. In particular, their setting is on the one hand easier than ours as they do not need to commit to a single reconstruction matrix, and on the other hand more difficult as they need to commit to P_t before seeing the current vector x_t. These differences lead to the use very different tools and methods compared to our paper. Sarlós (2006) and later Clarkson and Woodruff (2009) show that minimizing $\|X - (XY^+)Y\|_F^2$ can be done online in a surprisingly simple manner. Let $S \in \mathbb{R}^{d\times\ell}$ be a matrix with i.i.d. gaussian distributed entries. Then setting $y_t = S^T x_t$ yields a $1 + \varepsilon$ multiplicative approximation to the optimal value of $\|X - (XY^+)Y\|_F^2$ for some $\ell \in O(k/\varepsilon)$ with constant probability. Recently, Boutsidis et al. (2015) considered minimizing $\min_\Phi \|X - \Phi Y\|_F^2$ online where Φ is restricted to being an isometry[1]. Notice that $\Phi' = XY^+$ is the minimizer of the above expression among all matrices of appropriate dimensions but it is not an isometry in general. Hence, the requirement for Φ being an isometry introduces a new challenge. There are good reasons for preferring an isometric registration matrix Φ but this discussion goes beyond the scope of this paper. Boutsidis et al. (2015) show that one can obtain an approximate solution online and *deterministically* with an additive error of $\varepsilon\|X\|_F^2$, compared to the offline optimal solution of SVD with dimension k and a target dimension of $\tilde{O}(k/\varepsilon^2)$.

1.2. Our Contribution: Online PCA with Spectral Norm Bounds

To the best of our knowledge, this paper is the first to consider online PCA with respect to the spectral norm

$$\|X - (XY^+)Y\|^2 .$$

As stated above, the exact solution to this problem can be found (offline) by a partial SVD. However, while the exact minimizer of $\|X-(XY^+)Y\|_F^2$ is also the minimizer of $\|X-(XY^+)Y\|^2$, the same cannot be said about their approximate solutions. To make this point clear, consider an input matrix X whose first k singular values are equal to 1 and the rest are equal to $1/2$. We denote by σ_i the i'th singular value of X sorted in descending magnitude order. For this matrix

$$\min_Y \|X - (XY^+)Y\|_F^2 = \sum_{i=k+1}^{d} \sigma_i^2 = (d-k)/4 .$$

1. An isometry or an isometric matrix Φ is a matrix such that $\Phi^T\Phi = I$ or alternatively, $\forall z \|\Phi z\| = \|z\|$.

On the other hand, for any matrix Y,

$$\|X - (XY^{+})Y\|_F^2 \leq \|X\|_F^2 = (d-k)/4 + k \; .$$

Here, *any* solution Y is $1+\varepsilon$ approximation so long as $d \geq 5k/\varepsilon$. This is not the case when considering the spectral norm. The optimal Y perfectly captures the signal and

$$\min_Y \|X - (XY^{+})Y\|^2 = \sigma_{k+1}^2 = 1/4 \; .$$

In sharp contrast to the above, obtaining Y such that $\|X - (XY^{+})Y\|^2 \leq 1/4 + \varepsilon$ is far from trivial. It does not hold for a random Y and it does not hold for $Y = S^T X$ where S is random as in Sarlós (2006) and Clarkson and Woodruff (2009).

One might argue that such matrices are uncommon or that they are unreasonable inputs for PCA. We argue that both statements are incorrect. Consider X such that $X = S + N$ where S corresponds to a low dimensional signal and N to (roughly) isotropic noise. PCA can approximately recover S from X if the singular values of S are above $\|N\|$. Note that the spectrum of X is potentially very similar to that of the hard example above. This is, practically, the working model for statistical signal processing or factor analysis, and it is in this context that PCA excels as an analytic tool.

We propose two algorithms, each tailored to a slightly different scenario. The first scenario is the *fixed error setting* (Section 3), we are given as input a fixed bound Δ on the required spectral norm of the error matrix. Our goal is to provide reduced dimension vectors y_t such that $\|X - (XY^{+})Y\|^2$ meets the error requirement while requiring a small target dimension. The second scenario is the *adaptive error setting* (Section 4); we are given $\varepsilon > 0$ and k, the target dimension of the offline optimal solution (SVD) we wish to compete with. Our objective is to use a small as possible target dimension while keeping the spectral norm of the error bounded by $\sigma_{k+1}^2 + \varepsilon\sigma_1^2$.

Our algorithms operate online; they receive the vectors $x_t \in \mathbb{R}^d$ one by one in an arbitrary order and *deterministically* yield $y_t \in \mathbb{R}^\ell$ before receiving x_{t+1}. In the *fixed error setting* the target dimension of our algorithm is bounded by $O(k/\varepsilon)$ for any k for which $\sigma_{k+1}^2 < \Delta$ and corresponding $\varepsilon = (\Delta - \sigma_{k+1}^2)/\sigma_1^2$. In the *adaptive error setting* the target dimension is $O(\log(n)k/\varepsilon^2)$ in the worst case, but can potentially improve up to $O(k/\varepsilon)$ given a crude estimate of $\sigma_{k+1}^2 + \varepsilon\sigma_{k+1}^2$ or of $\sigma_1^2/\sigma_{k+1}^2$. In both settings the algorithm returns an *isometric* registration matrix U.

Our algorithm is inspired by that of Boutsidis et al. (2015) and should be considered a direct continuation of their work. Much like theirs, our algorithm works with an ever growing orthogonal basis U, and a new direction u_i is added once enough energy is observed in that direction. In fact, although it is not proven in their paper, their algorithm can also be adapted to provide spectral norm bounds. Even so, the properties of our algorithm make it preferable to that of Boutsidis et al. (2015) for several reasons. First, in order to reduce computational resources, both algorithms require a covariance sketch. Our algorithm can provably operate with any covariance sketch while the latter is limited to Frequent-Directions (See Liberty (2013)). This both simplifies the proof and enables a wider range of different implementations. Second, it sketches the original matrix X (and not its residual) which potentially reduces the sketching running time by utilizing the sparsity of x_t. Finally, it requires no special algorithmic handling of large normed vectors which used to be somewhat of a delicate issue implementation-wise.

1.3. Covariance Sketches

Let $A_{t_1:t_2}$ stand for the matrix whose columns are $a_{t_1}, \dots, a_{t_2}$ where a_t are the columns of A. A covariance sketch of a matrix A with an error bound ρ is an algorithm that receives the columns of

A one by one and maintains a sketch matrix B such that

$$\max_{t} \ \ \|A_{1:t}A_{1:t}^T - B_t B_t^T\| \le \rho \tag{1}$$

where B_t stands for the state of the sketch at time t. Note that one can trivially keep $A_{1:t}$ as its own "sketch" with error bound $\rho = 0$. This will trivially require $O(d)$ time to update and $\Theta(dn)$ space. One could also keep the covariance of A exactly with error bound $\rho = 0$. This brings the memory requirement down to $\Theta(d^2)$ but increases the update time to $\Theta(d^2)$, potentially.

There are several, much more efficient sketching techniques with $\rho > 0$, and any of them would suffice for our analysis to go through. To understand the guarantees offered we provide two examples. The most efficient algorithm in terms of space is Frequent Directions (See Liberty (2013); Ghashami and Phillips (2014a); Ghashami et al. (2015)). It requires $O(d\|A\|_F^2/\rho)$ space and $O(d\|A\|_F^2/\rho)$ floating point operations to add a vector to the sketch. In terms of update time the most efficient sketch is column sampling based on the work of Frieze et al. (1998); Drineas and Kannan (2003); Rudelson and Vershynin (2007). It exhibits update time proportional to the number of non zeros in the added vector. A somewhat relaxed but sufficient bound on its space requirement is $O(d\|A\|_F^4/\rho^2)$. As a remark, sampling is straight forward to implement efficiently (see the last appendix of Achlioptas et al. (2013)) and a natural choice in practice.

2. Fixed Error: A Conceptual Algorithm

Algorithm 1 is conceptually very simple. It is given as input a parameter Δ and ensures that the spectral norm of the error matrix does not significantly exceed Δ. Our guarantees regarding the target dimension, denoted by ℓ, are given with respect to the minimal k such that $\sigma_{k+1}^2 \le \Delta$. Algorithm 1 is provably correct but is wasteful with computational resources. Specifically it must maintain the entire history $X_{1:t}$ throughout the algorithm. Nevertheless, the reader is encouraged to keep this algorithm in mind as the blueprint for its modified and more efficient counterpart. The proof of its correctness is deferred to Section 3 because Algorithm 1 is identical to Algorithm 2 with the substitution of $B_t = X_{1:t}$ and $\rho = 0$.

Algorithm 1 Online PCA, Fixed error: Conceptual algorithm

input: X, Δ
$U \leftarrow$ all zeros matrix
for $x_t \in X$ **do**
  while $\|(I - UU^T)X_{1:t}\|^2 \ge \Delta$
    Add the top left singular vector of $(I - UU^T)X_{1:t}$ to U
  yield $y_t = U^T x_t$
end for

3. Fixed Error: A Space Efficient Algorithm

In this section we present Algorithm 2, tailored for the fixed error setting. In order to avoid keeping the matrix $X_{1:t}$ in memory, Algorithm 2 uses covariance sketching (see section 1.3). We denote by ρ the sketching approximation guarantee as detailed in Equation (1). We use E_t for the sketching error matrix $E_t = X_{1:t}X_{1:t}^T - B_t B_t^T$. Recall that the guarantees of the sketch producing B dictate that $\|E_t\| \leq \rho$ for all t. Note that one can store the covariance matrix $X_{1:t}X_{1:t}^T$ exactly and gain $\rho = 0$ at the cost of using $\Theta(d^2)$ space.

Algorithm 2 Online PCA, Fixed error: a space efficient algorithm

input: X, Δ
$U \leftarrow$ all zeros matrix
$B \leftarrow$ a covariance sketch with precision ρ
for $x_t \in X$ **do**
  Add x_t to the sketch B
  while $\|(I - UU^T)B\|^2 \geq \Delta$
    Add the top left singular vector of $(I - UU^T)B$ to U
  yield $y_t = U^T x_t$
end for

We denote by U_t and B_t the values taken by the matrices U, B at the end of iteration t in Algorithm 2. That is, U_t is the matrix used for computing $y_t = U_t^T x_t$. In particular as n denotes the length of the stream, U_n is the state of U at the end of the stream. We denote by u_i the i'th column of the matrix U and t_i the time of its insertion. That is, for $t < t_i$ the i'th column of U_t is equal to zero and for $t \geq t_i$ it is u_i.

Lemma 1 *Let ℓ denote the number of vectors u added by algorithm 2. Let σ_i be the singular values of X in descending magnitude order. Then for any $k \leq \ell$, assuming $\Delta > \sigma_{k+1}^2 + \rho$, it holds that*

$$\ell \leq \frac{k(\sigma_1^2 - \sigma_{k+1}^2)}{\Delta - \rho - \sigma_{k+1}^2}$$

Proof First, notice that $\|u_i^T X\|^2 \geq \Delta - \rho$. To verify that,

$$\begin{aligned} \|u_i^T X\|^2 &= \|u_i^T X_{1:t_i}\|^2 + \|u_i^T X_{t_i+1:n}\|^2 \geq u_i^T X_{1:t_i} X_{1:t_i}^T u_i \\ &= u_i^T B_{t_i} B_{t_i}^T u_i - u_i^T E_{t_i} u_i \overset{(i)}{\geq} \|u_i^T (I - U_{t_i-1} U_{t_i-1}^T) B_{t_i}\|^2 - \rho \geq \Delta - \rho \end{aligned}$$

Inequality (i) follows from $\|E_{t_i}\| \leq \rho$ and the columns of U being orthonormal. This orthonormality is guaranteed from the construction of U, as each new column u_i is the top singular vector, and in particular in the column span, of $(I - U_{t_i-1}U_{t_i-1}^T)B_{t_i}$. Summing the inequality $\Delta - \rho \leq \|u_i^T X\|^2$ over the ℓ different vectors u_i we obtain:

$$\ell(\Delta - \rho) \leq \sum_{i=1}^{\ell} \|u_i^T X\|^2 = \|U_n^T X\|_F^2 \leq \sum_{i=1}^{\ell} \sigma_i^2 \leq k\sigma_1^2 + (\ell - k)\sigma_{k+1}^2$$

Rearranging the inequality above completes the proof. ■

Lemma 2 *Let R be the residual matrix whose t'th column is $r_t = x_t - U_t U_t^T x_t$. Then*

$$\|X - (XY^+)Y\|_2^2 \leq \|R\|_2^2 \,.$$

Proof First, trivially $\|X - (XY^{+})Y\|_2^2 \leq \|X - U_nY\|_2^2$. Second, notice that $X - U_nY = R$. The t'th columns of $X - U_nY$ is equal to $x_t - U_nU_t^Tx_t = x_t - U_tU_t^Tx_t = r_t$. The fact that $U_nU_t^T = U_tU_t^T$ is because $U_nU_t^T = \sum_{i=1}^{\ell_t} u_iu_i^T + \sum_{i=\ell_t+1}^{\ell_n} u_i \cdot 0_d^T = U_tU_t^T$. Here, ℓ_t is the number of vectors u added by time t and 0_d is the all zeros vector in dimension d. ■

Given the above lemma we proceed to bound the norm of R. To do so we consider the vectors of U_n and their completion to an orthonormal basis spanning the d-dimensional space. In the following we present observations showing that (1) $\|u^TR\|$ is bounded for each of these vectors, and that (2) every pair $u^TR, (u')^TR$ is almost orthogonal.

Observation 3 *At the conclusion of every time step, we have $\|(I - U_tU_t^T)B_t\|^2 \leq \Delta$. This is an immediate consequence of the algorithm exiting the inner "while" loop at every time step.*

Observation 4 *For any vector $u_i \in U$ we have $\|u_i^TR\|^2 \leq \Delta + \rho$*

$$\begin{aligned} \|u_i^TR\|^2 &= \|u_i^TR_{1:t_i-1}\|^2 = \|u_i^TX_{1:t_i-1}\|^2 \leq \|u_i^TB_{t_i-1}\|^2 + \rho \\ &= \|u_i^T(I - U_{t_i-1}U_{t_i-1}^T)B_{t_i-1}\|^2 + \rho \leq \Delta + \rho \end{aligned}$$

Similarly, for any unit vector $u_\perp$ which is perpendicular to U_n we have $\|u_\perp^TR\|^2 \leq \Delta + \rho$.

Lemma 5 *Let u_i be a vector in U and let $u_\perp$ be a vector orthogonal to u_i, then*

$$u_i^TRR^Tu_\perp \leq \left(\rho + \max_t \|x_t\|^2\right) \|u_\perp\| \ .$$

Proof

$$\begin{aligned} u_i^TRR^Tu_\perp &= u_i^TX_{1:t_i-1}X_{1:t_i-1}^Tu_\perp = u_i^TX_{1:t_i}X_{1:t_i}^Tu_\perp - u_i^Tx_{t_i}x_{t_i}^Tu_\perp \\ &= u_i^TB_{t_i}B_{t_i}^Tu_\perp + u_i^TE_{t_i}u_\perp - u_i^Tx_{t_i}x_{t_i}^Tu_\perp = \sigma^2u_i^Tu_\perp + u_i^T(E_{t_i} - x_{t_i}x_{t_i}^T)u_\perp \\ &= u_i^T(E_{t_i} - x_{t_i}x_{t_i}^T)u_\perp \leq \left(\|E_{t_i}\| + \|x_{t_i}x_{t_i}^T\|\right)\|u_\perp\| \leq \left(\rho + \max_t \|x_t\|^2\right)\|u_\perp\| \end{aligned}$$

■

We are now ready to bound the spectral norm of R.

Lemma 6 $\|R\|_2^2 \leq \Delta + \rho + 2\sqrt{\ell}\left(\rho + \max_t \|x_t\|^2\right)$

Proof Denote by z the top left singular vector of R and for notational convenience set $u_{\ell+1}$ to be a unit vector in the same direction as $(I - U_nU_n^T)z$ (if $(I - U_nU_n^T)z = 0$ then $u_{\ell+1}$ can be set as an arbitrary unit vector orthogonal to U_n). Since z is supported by $u_1, \ldots, u_{\ell+1}$ we can write $z = \sum_{i=1}^{\ell+1} \alpha_iu_i$. Since the u vectors are orthonormal and z is a unit vector we have $\sum_i \alpha_i^2 = 1$. Using the observations above, it is possible to compute $\|z^TR\|^2$ directly.

$$\begin{aligned} z^T RR^T z &= \sum_i \alpha_i^2 \|u_i^T R\|^2 + \sum_{i=1}^{\ell+1} \sum_{j\neq i} \alpha_i \alpha_j u_i^T RR^T u_j \\ &= \Delta + \rho + 2\sum_{i=1}^{\ell} \alpha_i u_i^T RR^T (\sum_{j>i} \alpha_j u_j) \\ &\leq \Delta + \rho + 2\left(\rho + \max_t \|x_t\|^2\right) \sum_{i=1}^{\ell} |\alpha_i| \\ &\leq \Delta + \rho + 2\sqrt{\ell}\left(\rho + \max_t \|x_t\|^2\right) \end{aligned}$$

Inequality (2) is due to Lemma 5 and the fact that $\sum_{j>i} \alpha_j u_j$ is a vector of norm at most 1. Inequality (2) is due to $\|a\|_1 \leq \sqrt{\ell}\|a\|_2$ for any vector of dimension ℓ. ■

Theorem 7 *Let X and Δ be the inputs for Algorithm 2. Consider any k for which $\sigma_{k+1}^2 \leq \Delta$ and set ε such that $\varepsilon\sigma_1^2 + \sigma_{k+1}^2 = \Delta$. Assume $\text{poly}(k, 1/\varepsilon) \cdot \max \|x_t\|^2 = o(\sigma_1^2)$.[2] There exists a class of covariance sketches for which Algorithm 2 outputs Y such that*

1. *The target dimension ℓ is at most $2k/\varepsilon$.*
2. *Either the algorithm uses $O(d^2)$ memory and outputs Y such that:*

$$\|X - (XY^+)Y\|^2 \leq \|X - U_n^T Y\|^2 \leq \Delta + o(\sigma_1^2)$$

3. *Or, the algorithm uses $O(dk/\varepsilon + drk^{1/2}/\varepsilon^{3/2})$ memory and outputs Y such that:*

$$\|X - (XY^+)Y\|^2 \leq \|X - U_n^T Y\|^2 \leq \Delta + \varepsilon\sigma_1^2 + o(\sigma_1^2)$$

   Where $r = \|X\|_F^2/\|X\|^2$ is the numeric rank of X.[3]

Proof From Lemma 1 if the covariance sketch is of accuracy $\rho \leq \varepsilon\sigma_1^2$ we get that $\ell \leq 2k/\varepsilon$. From Lemma 6 if the exact covariance is kept in space $O(d^2)$ then $\rho = 0$. Using the assumption that $\text{poly}(k, 1/\varepsilon) \cdot \max \|x_t\|^2 = o(\sigma_1^2)$ completes the second claim. For $\rho > 0$, combining Lemma 6 and Lemma 2 we get that

$$\|X - (XY^+)Y\|_2^2 \leq \|R\|_2^2 \quad \leq \quad \Delta + (1 + 2\sqrt{\ell})\rho + 2\sqrt{\ell}\max_i \|x_i\|^2 \leq \Delta + \varepsilon\sigma_1^2 + o(\sigma_1^2)$$

if $\rho \leq \sigma_1^2(\varepsilon/3\sqrt{\ell})$ which is possible by a Frequent-Directions sketch (see section 1.3) of size $O(drk^{1/2}/\varepsilon^{3/2})$. ■

2. Notice that σ_1^2 is linear in n while in any reasonable setting $\max_t \|x_t\|^2$ is bounded by a constant. Even if $\max_t \|x_t\|^2$ grows asymptotically like $\sqrt{n}$, still, the assumption will hold for some $n \geq \text{poly}(k, 1/\varepsilon)$.
3. The numeric rank of a matrix is a stable version of its algebraic rank. It is lower bounded by 1 and upper bounded by the algebraic rank. Yet in many practical cases where X reflects data with some structure to it, the numeric rank of X is significantly smaller than its algebraic counterpart.

Notice that the running time of Algorithm 2 was not discussed. This is because, as written, the algorithm requires computing the spectral norm of the matrix $(I - UU^T)B$ at every iteration. This operation is a computational bottleneck. Other than this operation, the required running time is dominated by the time required by the covariance sketch and the time required to compute $y_t = U^T x_t$. Using a simple "trick", this norm computation can be avoided in the vast majority of the iterations. This leads to a running time that is dominated by covariance sketching and by the mapping $y_t = U^T x_t$. Nevertheless, in this section, we chose to present the simpler version to make the presentation more palpable.

In the following section we present Algorithm 3 which operates in the adaptive error setting. It shows how to avoid checking the spectral norm of the matrix $(I - UU^T)B$ in most iterations. The same technique can be easily adapted for Algorithm 2, leading to an asymptotic running time of $T_{\text{sketch}}(X, \rho) + O(\ell \cdot \text{nnz}(X) + \ell^2 d \log(\ell))$. Here, $T_{\text{sketch}}(X, \rho)$ is the time required for sketching X to within accuracy ρ and $\text{nnz}(X)$ is the number of non-zero entries in X. The additive term of $\ell^2 d \log(\ell)$ reflects the time spent on computing top singular vectors of $(I - UU^T)B$, as this can occur at most ℓ times. Typically, $\ell \cdot \text{nnz}(X)$ is the dominating term of the above expression.

4. Adaptive Error: A Time Efficient Algorithm

In this section we present Algorithm 3 which does not require as input a pre specified fixed error bound Δ. Instead, one can specify an integer k and scalar $\varepsilon > 0$ and obtain Y such that $\|X - (XY^+)Y\|^2 \leq \sigma_{k+1}^2 + O(\varepsilon\sigma_1^2)$. We present an additional modification that allows a more efficient running time. The bottleneck in terms of running time for Algorithm 2 is the need to check in each iteration the norm of $\|(I - UU^T)B\|$. Our modification allows us to compute the norm only after seeing a substantial amount of energy since the last time it was computed. In other words, there is a computationally attractive way to maintain a loose upper bound for $\|(I - UU^T)B\|$ that allows us to compute the actual value only $o(n)$ times throughout the execution of the algorithm. Since our input only includes k and ε, our challenge is to find the 'correct' value for Δ they correspond to. This is done via a doubling trick. Based on $\|x_1\|^2$ we compute an initial value for Δ and exponentially increase it until reaching the desired value.

Denote by Δ_t the value taken by Δ at the end of iteration t. We follow the outline of the analysis of Section 3 and begin with a proof that $\|(I - U_t U_t^T)B_t\|^2$ is always bounded from above by (roughly) Δ despite the fact that we do not compute it in every iteration.

Lemma 8 *At the conclusion of every time step, we have* $\|(I - U_t U_t^T)B_t\|^2 \leq \Delta_t + (2 + \varepsilon)\rho + \max_t \|x_t\|^2$.

Proof The statement clearly holds in iterations where the condition inside the 'if' statement is held. Consider an iteration t where we did not enter the if statement. Let $t' < t$ be the last iteration where we did enter the if statement, where $t' = 0$ if no such iteration exists. For some unit vector $v \in \mathbb{R}^d$

Algorithm 3 Online PCA, Adaptive error: a time efficient algorithm

input: X, k, ε
$U \leftarrow$ all zeros matrix
$\ell \leftarrow \lceil k/\varepsilon \rceil$
$B \leftarrow$ a covariance sketch with guaranteed precision ρ
$\omega \leftarrow 0$
$\Delta \leftarrow 2\sqrt{\ell}\|x_1\|^2$
for $x_t \in X$ **do**
  Add x_t to the sketch B
  $\omega \leftarrow \omega + \|(I - UU^T)x_t\|^2$
  if $\omega > \varepsilon(\Delta + \rho)$ **then**
    while $\|(I - UU^T)B\|^2 \geq \Delta(1-\varepsilon)$ **do**
      Add the top left singular vector of $(I - UU^T)B$ to U
      If $|U|$ increased by ℓ vectors since last update of Δ, $\Delta \leftarrow \Delta \cdot (1+\varepsilon)$
    end while
    $\omega \leftarrow 0$
  end if
  yield $y_t = U^T x_t$
end for

such that $U_t^T v = 0$ we have that

$$\begin{aligned} \|(I - U_t U_t^T)B_t\|^2 &= \|v^T B_t\|^2 \leq \|v^T X_{1:t}\|^2 + \rho \\ &\leq \|v^T X_{1:t'}\|^2 + \|(I - U_t U_t^T)X_{t'+1:t}\|_F^2 + \rho \\ &\leq \|v^T B_{t'}\|^2 + \varepsilon(\Delta_t + \rho) + \|x_t\|^2 + 2\rho \\ &\overset{(i)}{\leq} \|(I - U_{t'} U_{t'}^T)B_{t'}\|^2 + \varepsilon\Delta_t + \|x_t\|^2 + (2+\varepsilon)\rho \\ &\leq \Delta_t + (2+\varepsilon)\rho + \|x_t\|^2 \ . \end{aligned}$$

Inequality (i) is since $U_t = U_{t'}$ as we did not enter the while statement between iterations t' and t. ■

Let Δ_n denote the final value taken by Δ. Given Lemma 8 it is an easy exercise to prove analogous results to those of Lemmas 2 and 6 in Section 3. These are expressed w.r.t Δ_n, the largest value taken by Δ. Formally, we have that

$$\|X - (XY^+)Y\|^2 \leq \Delta_n + \left(\varepsilon + 3 + 2\sqrt{\bar{\ell}}\right)\left(\rho + \max_t \|x_t\|^2\right)$$

with $\bar{\ell}$ being the target dimension, i.e. the total number of vectors eventually added to U. In the following Lemma we provide the bound on Δ_n, leading to a bound on $\bar{\ell}$.

Lemma 9 *It holds that*

$$\Delta_n \leq \max\left\{\sqrt{\ell}\|x_1\|^2, (1+\varepsilon)\frac{\sigma_{k+1}^2 + \rho + \varepsilon\sigma_1^2}{1-\varepsilon}\right\} \leq \sigma_{k+1}^2 + 5\varepsilon\sigma_1^2 + \sqrt{\ell}\|x_1\|^2 + 2\rho$$

for $\varepsilon \leq 0.5$ and $\bar{\ell} \leq \ell \log_{1+\varepsilon}(\Delta_n/\Delta_0)$. Assuming that $\sum_{t=1}^{n} \|x_t\|^2/\sqrt{\ell}\|x_1\|^2 = n^{O(1)}$, which in particular occurs when all numbers are expressed with an accuracy of up to $O(\log(n))$ bits, we have $\bar{\ell} = O(\ell \log(n)/\varepsilon)$.

Proof Let Δ_t be first value taken by Δ for which

$$\Delta_t > \frac{\sigma_{k+1}^2 + \rho + \varepsilon\sigma_1^2}{1-\varepsilon}$$

Let u be a vector added to U during the time where $\Delta = \Delta_t$, and let t_u be the iteration number in which u was inserted. We have,

$$\begin{aligned} \|u^T X\|^2 &= \|u^T X_{1:t_u}\|^2 + \|u^T X_{t_u+1:n}\|^2 \geq u^T X_{1:t_u} X_{1:t_u}^T u \\ &= u^T B_{t_u} B_{t_u}^T u - u^T E_{t_i} u \geq \|u^T (I - U_{t_u-1} U_{t_u-1}^T) B_{t_u}\|^2 - \rho \\ &\geq \Delta(1-\varepsilon) - \rho \end{aligned}$$

Denote by $u_1, \ldots, u_{\ell'}$ the vectors inserted to U during the time periods in which $\Delta = \Delta_t$. We obtain, by summing the inequality $\Delta(1-\varepsilon) - \rho \leq \|u_i^T X\|^2$ over the ℓ' different vectors u_i, that

$$\ell'(\sigma_{k+1}^2 + \varepsilon\sigma_1^2) < \ell'(\Delta(1-\varepsilon) - \rho) \leq \sum_{i=1}^{\ell'} \|u_i^T X\|^2 \leq \sum_{i=1}^{\ell'} \sigma_i^2 \leq k\sigma_1^2 + (\ell' - k)\sigma_{k+1}^2$$

Hence $\ell' < k/\varepsilon \leq \ell$, and it must be the case that $\Delta_n = \Delta_t$. Since the updates are multiplicative with a factor of $1+\varepsilon$ the bound for Δ_n follows. The inequality $\bar{\ell} \leq \ell + \ell \log_{1+\varepsilon}(\Delta_n/\Delta_0)$, with $\Delta_0 = \sqrt{\ell}\|x_1\|^2$ being the initial value of Δ, is trivial due to the algorithm structure. A (very) crude upper bound for Δ_n is $\|X\|_F^2$, and Δ_n/Δ_0 is bounded by $\sum_{t=1}^{n} \|x_t\|^2/\sqrt{\ell}\|x_1\|^2$. According to our assumption we have that the mentioned quantity is upper bounded by n^c for a small constant c (most likely $c = 1 + o(1)$). The claim immediately follows. ■

We are now done with the analysis of the quality of the output of the algorithm. The remaining difference in the analysis of the improved algorithm and that of Section 3 is that of the running time. We prove in the following that we do not need to compute the spectral norm of $(I - UU^T)B$ too many times, hence the amortized update time is dominated by that of the sketching procedure.

Lemma 10 *In Algorithm 3, after entering the code inside the if statement at most $\ell d/\varepsilon$ times the value of Δ increases. In other words, the number of times the condition of the 'if' statement is true is at most $O(\ell d \log(n)/\varepsilon^2)$.*

Proof Consider a time t in which $\Delta = \Delta_t$. Let $t' > t$ be an index of an iteration such that between time t and t' we entered the if statement d/ε times. To prove the claim it suffices to show that we must have added a vector to U between times t and t'. Indeed, if this is the case then after $\ell d/\varepsilon$ times of entering the if statement we insert ℓ directions to U and Δ is increased.

Let $t_1, \ldots, t_m$ for $m \geq d/\varepsilon$ be the iterations in which we entered the if statement after time t. For t_i, either a direction entered U between times t and t_i or $\|(I - U_t U_t^T) X_{t:t_i}\|_F^2 \geq i \cdot \varepsilon(\Delta + \rho)$. Hence, if we did not enter any direction to U at time t_m we must have

$$\|(I - U_t U_t^T) X_{1:t_m}\|^2 \geq \|(I - U_t U_t^T) X_{t:t_m}\|^2 \geq$$

$$\|(I - U_{t_m-1}U_{t_m-1}^T)X_{t:t_m}\|_F^2/d \geq \Delta + \rho$$

It follows that $\|(I - U_{t_m-1}U_{t_m-1}^T)B_{t_m}\|^2 > \Delta$ and a direction entered U at time $t_m \leq t'$. ■

Theorem 11 *Combining the above we get the following: assume Algorithm 3 received as input parameters k, ε and a sketching algorithm with guarantee ρ, update time $T_{\text{sketch}}(X, \rho)$ and a memory requirement of $S_{\text{sketch}}(X, \rho)$. We have*

1. *The target dimension of the sketch is $O(k \log(n)/\varepsilon^2)$.*
2. *The running time is bounded by $T_{\text{sketch}}(X, \rho) + O(\text{nnz}(X)k \log(n)/\varepsilon^2) + O\left(kd \log(n)/\varepsilon^3\right)$ where $\text{nnz}(X)$ is the number of non-zero entries in X. For sufficiently large n (as common in streaming scenarios) this quantity is in fact $T_{\text{sketch}}(X, \rho) + O(\text{nnz}(X)k \log(n)/\varepsilon^2)$.*
3. *The space requirement of the algorithm is $S_{\text{sketch}}(X, \rho) + O(kd \log(n)/\varepsilon^2)$.*
4. *The error of the output is bounded by*

$$\|X - (XY^+)Y\|^2 \leq \sigma_{k+1}^2 + 5\varepsilon\sigma_1^2 + O\left(\sqrt{k \log(n)/\varepsilon^2}\left(\rho + \max_t \|x_t\|^2\right)\right) = \sigma_{k+1}^2 + O(\varepsilon\sigma_1^2)\,.$$

Possible improvements: Having prior knowledge about the matrix, two potential improvements can be made. In some cases we have a crude approximation for $\Delta^* = \sigma_{k+1}^2 + \varepsilon\sigma_1^2$. By this we mean having knowledge of a scalar Δ_0 such that $\Delta_0 \leq \Delta^*$ but $\Delta_0 \geq \Delta^*/c$, for some large constant c. If this happens to be the case we can initialize Δ to be Δ_0 and the $\log(n)$ terms in the above theorems become $\log(c)$. The second improvement can be made when we have some lower bound $1 < \kappa \leq \sigma_1^2/\sigma_{k+1}^2$. First, notice that typically it makes sense to have an input k for which $\sigma_{k+1}^2 \ll \sigma_1^2$, hence κ can be potentially large. When having knowledge of such a parameter we can set the multiplicative update of Δ to grow by $1 + \varepsilon\kappa$ rather than $1 + \varepsilon$. The results stated above regarding the error guarantee remain the same as long as $\kappa \leq \sigma_1^2/\sigma_{k+1}^2$; however, the running time, memory complexity and target dimension are decrease by a factor of $\max\{1/\varepsilon, \kappa\}$. To conclude, in an optimistic, yet not unlikely scenario where we have knowledge of $\Delta_0 = \Omega(\Delta^*)$ and $\kappa = \Omega(\varepsilon^{-1})$ we get a target dimension of $O(k/\varepsilon)$.

References

Dimitris Achlioptas, Zohar Karnin, and Edo Liberty. Near-optimal entrywise sampling for data matrices. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 1565–1573, 2013.

Raman Arora, Andy Cotter, and Nati Srebro. Stochastic optimization of pca with capped msg. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1815–1823. 2013.

Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental pca. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3174–3182. 2013.

Christos Boutsidis, Dan Garber, Zohar Karnin, and Edo Liberty. Online principal components analysis. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 887–901, 2015.

Kenneth L Clarkson and David P Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 205–214. ACM, 2009.

Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *APPROX-RANDOM*, pages 292–303, 2006.

Petros Drineas and Ravi Kannan. Pass efficient algorithms for approximating large matrices. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 223–232, 2003.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 370–, 1998.

Mina Ghashami and Jeff M. Phillips. Relative errors for deterministic low-rank matrix approximations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 707–717, 2014a.

Mina Ghashami and Jeff M. Phillips. Relative errors for deterministic low-rank matrix approximations. In *SODA*, 2014b.

Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions : Simple and deterministic matrix sketching. *CoRR*, abs/1501.01711, 2015.

Edo Liberty. Simple and deterministic matrix sketching. In *KDD*, pages 581–588, 2013.

Edo Liberty, Franco Woolfe, Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences,*, 104(51):20167–20172, December 2007.

Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory limited, streaming PCA. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2886–2894. 2013.

Jiazhong Nie, Wojciech Kotłowski, and Manfred K. Warmuth. Online PCA with optimal regrets. In *ALT*, pages 98–112, 2013.

Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *J. ACM*, 54(4), July 2007. ISSN 0004-5411.

Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS*, pages 143–152, 2006.

Manfred K. Warmuth and Dima Kuzmin. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension, 2007.