Streaming Sparse Principal Component Analysis

Wenzhuo Yang A0096049@NUS.EDU.SG
Department of Mechanical Engineering, National University of Singapore, Singapore 117576

Huan Xu MPEXUH@NUS.EDU.SG
Department of Mechanical Engineering, National University of Singapore, Singapore 117576

Abstract

This paper considers estimating the leading k principal components with at most s non-zero attributes from p-dimensional samples collected sequentially in memory limited environments. We develop and analyze two memory and computational efficient algorithms called streaming sparse PCA and streaming sparse ECA for analyzing data generated according to the spike model and the elliptical model respectively. In particular, the proposed algorithms have memory complexity $O(pk)$, computational complexity $O(pk \min\{k, s \log p\})$ and sample complexity $\Theta(s \log p)$. We provide their finite sample performance guarantees, which implies statistical consistency in the high dimensional regime. Numerical experiments on synthetic and real-world datasets demonstrate good empirical performance of the proposed algorithms.

1. Introduction

Principal component analysis (PCA) (Pearson, 1901), arguably the most widely used dimension reduction method, is a fundamental tool in data analysis in a wide range of areas including machine learning, finance, statistics and many others. Standard PCA extracts the principal components (PCs) of a set of samples by computing the leading eigenvectors of the sample covariance matrix. However, in the face of modern high dimensional data with $p \gg n$, PCA is no longer statistically solid. Indeed, Johnstone & Lu (2009) showed that the consistency of PCA depends crucially on the limiting value of p/n: the angle between the PCA estimate and the true leading PC does not converge to zero unless p/n goes to zero.

To address this inconsistency issue and encourage more interpretable solutions, most of previous works focus on the sparse setting assuming that the leading PCs are *sparse*, i.e., only a few of their attributes are non-zero. In this setting, many variants of sparse PCA have been developed (e.g., Zou et al., 2006; d'Aspremont et al., 2007; Shen & Huang, 2008; Journee & Y. Nesterov, 2008; Birnbaum et al., 2013; Vu et al., 2013; Yuan & Zhang, 2013; Wang et al., 2014). For example, Zou et al. (2006) proposed to use a regression-type optimization problem based on the elastic-net to compute sparse PCs. d'Aspremont et al. (2007) considered a convex semidefinite program formulation for sparse PCA. Yuan & Zhang (2013) and Ma (2013) proposed the TPower method and the iterative thresholding sparse PCA respectively, which are essentially modified variants of the classical power method. Vu et al. (2013) proposed the Fantope projection selection method (FPS) which is a convex relaxation formulation of sparse principal subspace estimation based on a semidefinite program.

Yet, it can be hard to apply these sparse PCA methods to real large scale data, as 1) they need to explicitly compute the sample covariance matrix or store all the samples, which means that at least $O(p \min\{p, n\})$ storage is required; 2) their computational cost may become prohibitive when the dimensionality is high. For example, FPS is solved via an ADMM algorithm that requires to perform spectral decomposition on a $p \times p$ matrix *in each iteration*.

For non-sparse PCA, many computation/memory efficient algorithms have been proposed in recent years. For example, Warmuth & Kuzmin (2008) proposed a multiplicative update algorithm called online PCA under the streaming data model – i.e., the samples are received sequentially. Although each update of online PCA can be calculated efficiently, it still requires $O(p^2)$ storage. Brand (2002) and Arora et al. (2012) developed two variants of PCA – incremental PCA and the stochastic power method – both of which have low memory and computational complexity. Yet, they only showed the empirical performance and provided no theoretical performance guarantees. Recently, Mitliagkas et al. (2013) proposed streaming PCA

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

with memory complexity $O(pk)$ and sample complexity $\Theta(p \log p)$. However, similar to PCA, all these method are inconsistent in the high dimensional regime since sparsity is not exploited. Indeed, how to design a computation- and memory-efficient *sparse* PCA method remains unsolved. Mairal et al. (2010) proposed an online learning algorithm for sparse coding that leads to an online sparse PCA algorithm. Although this algorithm requires only $O(pk)$ memory, its computational cost is high due to solving the elastic-net problem in each iteration.

Another important issue is the sub-Gaussianality assumption. Many sparse PCA methods are theoretically analyzed under the spike model (e.g., Amini & Wainwright, 2009; Vu et al., 2013; Yuan & Zhang, 2013; Vu & Lei, 2012; Shen et al., 2013; Mitliagkas et al., 2013; Cai et al., 2014). The limitation of the spike model is it requires subgaussian data and noise, and hence can not model heavy-tail distributions. To relax this assumption, Han & Liu (2013b) used the semiparametric transelliptical family to model data and proposed the transelliptical component analysis (TCA) based on the the marginal Kendall's tau statistic for estimating the leading eigenvectors of the correlation matrix. In their following-up work (Han & Liu, 2013a), they developed the elliptical component analysis (ECA) based on the multivariate Kendall's tau statistic for estimating the leading eigenvectors of the covariance matrix under the elliptical family (Fang et al., 1990) – a semiparametric generalization of the Gaussian family that can model heavy-tail distributions and nontrivial tail dependence between variables (Hult & Lindskog, 2002). Although TCA and ECA have beautiful theoretical guarantees, they require at least $O(p^2n^2 + p^3)$ computation and $O(p^2)$ memory due to the calculation of the marginal/multivariate Kendall's tau matrix and the spectral decomposition, which makes them unsuitable for large-scale applications.

In this paper, to address the issues discussed above, we propose two variants of sparse PCA – streaming sparse PCA and streaming sparse ECA for estimating the leading PCs of samples drawn from the spike model and the elliptical model, respectively. Our theoretical analysis shows that both algorithms have memory complexity $O(pk)$, computational complexity $O(pk \min\{k, s \log p\})$ and sample complexity $\Theta(s \log p)$, and are consistent in the high dimensional regime.

Notation. Matrices and column vectors are denoted by upper-case and lower-case boldface letters, respectively. For matrix $\mathbf{X}$, we use $\|\mathbf{X}\|_0$ and $\|\mathbf{X}\|_2$ to denote the number of non-zero entries and the spectral norm of $\mathbf{X}$, respectively. The first k singular values of $\mathbf{X}$ are denoted by $\lambda_1(\mathbf{X}), \cdots, \lambda_k(\mathbf{X})$, and the i^{th} column and the i^{th} row of $\mathbf{X}$ are denoted by $\mathbf{X}_i$ and $\mathbf{X}_{(i)}$ respectively. For a square matrix $\mathbf{S}$, its main diagonal is denoted by $\mathrm{diag}(\mathbf{S})$.

2. Problem Setup

We consider the streaming data model where one receives sample $\mathbf{x}_t$ at time t for $t = 1, 2, \cdots$, and $\mathbf{x}_t$ vanishes after it is collected unless it is stored in the memory. Our goal is to extract the leading k PCs of the received data. We consider the following two models that generate the samples:

Spike model. Sample $\mathbf{x}_t$ is generated according to $\mathbf{x}_t = \mathbf{A}\mathbf{z}_t + \mathbf{w}_t$ where $\mathbf{z}_t \in \mathbb{R}^d$ is an i.i.d. sample of the random vector $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_d)$, $\mathbf{w}_t \in \mathbb{R}^p$ is an i.i.d. Gaussian noise drawn from $\mathcal{N}(0, \sigma^2\mathbf{I}_p)$, and matrix $\mathbf{A} \in \mathbb{R}^{p\times d}$ is deterministic but unknown. Let $\mathbf{\Sigma}$ be the covariance of $\mathbf{x}_t$, i.e., $\mathbf{\Sigma} \triangleq \mathbb{E}[\mathbf{x}_t\mathbf{x}_t^\top] = \mathbf{A}\mathbf{A}^\top + \sigma^2\mathbf{I}_p$.

Elliptical model. Sample $\mathbf{x}_t$ is generated according to the elliptical distribution $EC_p(\mu, \mathbf{\Sigma}, \xi)$, i.e., $\mathbf{x}_t = \boldsymbol{\mu} + \xi_t\mathbf{A}\mathbf{z}_t$, where $\mathbf{z}_t \in \mathbb{R}^d$ is a sample of $\mathbf{z}$ which is a uniform random vector on the unit sphere, ξ_t is a sample of ξ which is a scalar random variable (with unknown distribution) independent of $\mathbf{z}$, $\boldsymbol{\mu} \in \mathbb{R}^p$ is a fixed vector, and $\mathbf{A} \in \mathbb{R}^{p\times d}$ is a deterministic matrix satisfying $\mathbf{A}\mathbf{A}^\top = \mathbf{\Sigma}$. We only consider the case that ξ has a continuous distribution.

As our algorithms are designed for the sparse setting, we assume that the projection matrix $\mathbf{\Pi} \triangleq \mathbf{U}_k\mathbf{U}_k^\top$ onto the subspace spanned by the eigenvectors $\mathbf{U}_k \in \mathbb{R}^{p\times k}$ of $\mathbf{\Sigma}$ corresponding to its k largest eigenvalues satisfies $\|\mathrm{diag}(\mathbf{\Pi})\|_0 \leq s$, where s indicates the sparsity which is less than the dimension p and the number of samples n.

3. Algorithm

We now present the details of streaming sparse PCA and streaming sparse ECA for analyzing high dimensional data. Similar to streaming PCA (Mitliagkas et al., 2013), our algorithms are block-wise stochastic power methods that update the estimated PCs once a block of samples are received. The difference between our algorithms and streaming PCA is that we propose to apply a "row truncation" operator as shown in Algorithm 1 to maintain the sparsity of the estimated PCs to achieve consistency in the high dimensional regime, which truncates the row vectors of a matrix to zero except the ones with the largest l_2-norms.

Algorithm 1 Row Truncation Operator

Input: Matrix $\mathbf{X} \in \mathbb{R}^{p\times k}$ and parameter s.
Procedure:
1) Compute $v_i = \|\mathbf{X}_{(i)}\|_2$ for $i = 1, \cdots, p$;
2) Sort $\{v_i\}$ and select the largest s ones. Let $\mathcal{I}$ be the selected indices;
3) Compute $\tilde{\mathbf{X}}$ where $\tilde{\mathbf{X}}_{(i)} = \mathbf{X}_{(i)}$ if $i \in \mathcal{I}$ or 0 otherwise, and return $\tilde{\mathbf{X}}$.

We first present two streaming sparse PCA algorithms, i.e.,

Algorithm 2 and 3, for analyzing data generated according to the spike model. Algorithm 2 accepts a sequence of samples and estimates the leading k sparse PCs simultaneously. The samples are divided into blocks with equal-size B. In each block, the estimated PCs are updated via the power method update followed by the row truncation operation and the QR decomposition. Clearly, its memory complexity is $O(pk)$ since only the estimated PCs need to be stored in the memory. Its computational complexity for each iteration is $O(p(k^2+\log p)+pkB)$ since the computation of the power method update requires $O(pkB)$ operations, the row truncation operation needs $O(pk+p\log p)$ operations and the QR decomposition requires $O(pk^2)$ operations. In the next section, we show that block size $B=\Theta(s\log p)$ which implies that streaming sparse PCA has a much lower computational complexity $O(pk\min\{k,s\log p\})$ than the methods proposed by Vu et al. (2013); d'Aspremont et al. (2007) when s is much smaller than p.

Algorithm 2 Streaming SPCA via Row Truncation

Input: Samples $\{\mathbf{x}_1,\mathbf{x}_2,\cdots\}$, number of steps T, block size B, parameter γ and initial solution $\mathbf{Q}_0\in\mathbb{R}^{p\times k}$.
Procedure:
for $\tau=0$ to $T-1$ **do**
1) Initialize $\tilde{\mathbf{S}}_{\tau+1}=0$;
for $t=B\tau+1$ to $B(\tau+1)$ **do**
2) $\tilde{\mathbf{S}}_{\tau+1}=\tilde{\mathbf{S}}_{\tau+1}+\frac{1}{B}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{Q}_\tau$;
end for
3) $\mathbf{S}_{\tau+1}=\text{Truncate}(\tilde{\mathbf{S}}_{\tau+1},\gamma)$;
4) QR-decomposition: $\mathbf{S}_{\tau+1}=\mathbf{Q}_{\tau+1}\mathbf{R}_{\tau+1}$;
end for
5) Return $\mathbf{Q}_T$.

Intuitively, Algorithm 2 works when matrix $\mathbf{U}_k\in\mathbb{R}^{p\times k}$ – which consists of the leading k PCs – is row sparse, or equivalently, the projection matrix $\mathbf{\Pi}$ onto the subspace spanned by the leading k PCs satisfies $\|\text{diag}(\mathbf{\Pi})\|_0\le s$. If this "row sparse" assumption is not satisfied, e.g., the leading k PCs are all sparse but their supports are nearly disjoint, one can compute the PCs iteratively via the iterative deflation method (d'Aspremont et al., 2007; Mackey, 2008), as shown in Algorithm 3.

Our streaming sparse PCA algorithm has the following advantages compared with streaming PCA and TPower: 1) As we show in the next section, the streaming sparse PCA is consistent in the high dimensional regime where p is much larger than n – a regime where streaming PCA is known to be inconsistent. 2) TPower is not designed to handle the streaming data model. It needs to store all the samples or explicitly compute the empirical covariance matrix, which requires at least $O(p\min\{p,n\})$ storage. This can be problematic in applications involving large data. In contrast, our methods only require $O(pk)$ storage. 3) When

Algorithm 3 Streaming SPCA via Iterative Deflation

Input: Parameters $B_1,\cdots,B_k,T_1,\cdots,T_k,\gamma_1,\cdots,\gamma_k$ and samples $\{\mathbf{x}_1,\mathbf{x}_2,\cdots\}$.
Procedure:
1) Let $\bar{n}=0$;
for $i=1$ to k **do**
2) Initialize $\mathbf{q}_0^{(i)}$;
4) Run Algorithm 2 with $\{\mathbf{y}_{\bar{n}},\cdots,\mathbf{y}_{\bar{n}+T_iB_i}\}$, T_i, B_i, $\mathbf{q}_0^{(i)}$, γ_i and $k=1$. For $t=\bar{n},\cdots,\bar{n}+T_iB_i$, $\mathbf{y}_t$ is defined as $\mathbf{y}_t\triangleq\mathbf{x}_t-\sum_{j=0}^{i-1}\mathbf{q}^{(j)}\mathbf{q}^{(j)\top}\mathbf{x}_t$. The output of Algorithm 2 is denoted by $\mathbf{q}^{(i)}$;
5) Set $\bar{n}=\bar{n}+T_iB_i$;
end for
6) Return $\mathbf{q}^{(1)},\cdots,\mathbf{q}^{(k)}$.

the row sparse assumption holds, Algorithm 1 can compute the leading k PCs *simultaneously*, but TPower can only extract them one by one via the iterative deflation method.

For elliptically distributed data, Han & Liu (2013a) proposed the ECA algorithm based on the multivariate Kendall's tau statistic. In particular, let $\mathbf{x}$ be a random vector following the elliptical distribution $EC_p(\mu,\mathbf{\Sigma},\xi)$ and $\tilde{\mathbf{x}}$ be an independent copy of $\mathbf{x}$. The multivariate Kendall's tau matrix is defined as

$$\mathbf{K}\triangleq\mathbb{E}\left[\frac{(\mathbf{x}-\tilde{\mathbf{x}})(\mathbf{x}-\tilde{\mathbf{x}})^\top}{\|\mathbf{x}-\tilde{\mathbf{x}}\|_2^2}\right]. \tag{1}$$

It is known that the eigenspace of $\mathbf{K}$ is identical to that of $\mathbf{\Sigma}$. Based on this fact, Han & Liu (2013a) proposed to recover the eigenspace of a second order U-statistic estimator of $\mathbf{K}$ which is defined as

$$\hat{\mathbf{K}}_U\triangleq\frac{2}{n(n-1)}\sum_{i'<i}\frac{(\mathbf{x}_i-\mathbf{x}_{i'})(\mathbf{x}_i-\mathbf{x}_{i'})^\top}{\|\mathbf{x}_i-\mathbf{x}_{i'}\|_2^2},$$

where $\{\mathbf{x}_1,\cdots,\mathbf{x}_n\}$ are n independent samples of $\mathbf{x}$. Note that $\hat{\mathbf{K}}_U$ is indeed the empirical covariance matrix of $\{\frac{\mathbf{x}_i-\mathbf{x}_{i'}}{\|\mathbf{x}_i-\mathbf{x}_{i'}\|_2}\}_{i'<i}$. $\hat{\mathbf{K}}_U$ is not suitable for the streaming data model because its computation requires to store all the samples and is quite time-consuming when n is large. So instead of $\hat{\mathbf{K}}_U$, we consider the following estimator:

$$\hat{\mathbf{K}}\triangleq\frac{2}{n}\sum_{i=1}^{n/2}\frac{(\mathbf{x}_{2i-1}-\mathbf{x}_{2i})(\mathbf{x}_{2i-1}-\mathbf{x}_{2i})^\top}{\|\mathbf{x}_{2i-1}-\mathbf{x}_{2i}\|_2^2}. \tag{2}$$

Without loss of generality, we assume that n is an even number. This leads to streaming sparse ECA proposed in Algorithm 4. In comparison with ECA, streaming sparse ECA is able to deal with the streaming data model and requires only $O(pk)$ storage. We remark that the main difference between Algorithms 2 and 4 is that Algorithm 4 computes the empirical covariance of $\{\frac{\mathbf{x}_{2t-1}-\mathbf{x}_{2t}}{\|\mathbf{x}_{2t-1}-\mathbf{x}_{2t}\|_2}\}_{t=B\tau+1}^{B(\tau+1)}$,

Algorithm 4 Streaming ECA via Row Truncation

Input: Samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots\}$, number of steps T, block size B, parameter γ and initial solution $\mathbf{Q}_0 \in \mathbb{R}^{p\times k}$.

Procedure:

for $\tau = 0$ to $T-1$ **do**

1) Initialize $\tilde{\mathbf{S}}_{\tau+1} = 0$;

for $t = B\tau + 1$ to $B(\tau+1)$ **do**

2) $\mathbf{y}_t = (\mathbf{x}_{2t-1} - \mathbf{x}_{2t})/\|\mathbf{x}_{2t-1} - \mathbf{x}_{2t}\|_2$;

3) $\tilde{\mathbf{S}}_{\tau+1} = \tilde{\mathbf{S}}_{\tau+1} + \frac{1}{B}\mathbf{y}_t\mathbf{y}_t^\top\mathbf{Q}_\tau$;

end for

4) $\mathbf{S}_{\tau+1} = \text{Truncate}(\tilde{\mathbf{S}}_{\tau+1}, \gamma)$;

5) QR-decomposition: $\mathbf{S}_{\tau+1} = \mathbf{Q}_{\tau+1}\mathbf{R}_{\tau+1}$;

end for

6) Return $\mathbf{Q}_T$.

rather than the empirical covariance matrix of $\{\mathbf{x}_t\}_{t=B\tau+1}^{B(\tau+1)}$, to handle elliptically distributed data.

4. Performance Guarantees

To measure the accuracy of the output, we use the following distance function (Mitliagkas et al., 2013). For two orthogonal matrix $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{p\times k}$, the distance between $\mathbf{U}$ and $\mathbf{V}$ is defined based on the largest principal angle between the column spaces of $\mathbf{U}$ and $\mathbf{V}$:

$$\text{dist}(\mathbf{U}, \mathbf{V}) = \text{dist}(\text{span}(\mathbf{U}), \text{span}(\mathbf{V})) = \|\mathbf{U}_\perp^\top\mathbf{V}\|_2, \quad (3)$$

where $\mathbf{U}_\perp$ is an orthogonal basis of the perpendicular subspace to the one spanned by the columns of $\mathbf{U}$.

4.1. Streaming Sparse PCA

We first show performance guarantees of streaming sparse PCA, i.e., Algorithm 2 and 3, which handles data generated under the spike model. Recall that the covariance matrix is denoted by $\mathbf{\Sigma}$. Let $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ be the singular value decomposition of $\mathbf{\Sigma}$, $\mathbf{U}_k$ be a submatrix of $\mathbf{U}$ consisting of the first k columns of $\mathbf{U}$ and λ_k be the k^{th} largest eigenvalue of $\mathbf{\Sigma}$. Theorem 1 shows sufficient conditions for Algorithm 2 to obtain a solution of accuracy ϵ. We assume that the initial solution $\mathbf{Q}_0$ satisfies $\|\mathbf{U}_{k,\perp}^\top\mathbf{Q}_0\| > \epsilon$.

Theorem 1. *For $\eta > 0$, $0 < \epsilon < 1$, and $\gamma \geq s$, let $\mu \triangleq \frac{(k+1)\lambda_{k+1}+2\eta\lambda_k}{\lambda_k}$, $f(\mu,\eta,k) \triangleq \max\{\frac{(2+\sqrt{2})\mu}{\sqrt{k}}, \frac{\eta}{k}\}$. If the initial solution $\mathbf{Q}_0$ satisfies that*

$$\nu \triangleq \|\mathbf{U}_{k,\perp}^\top\mathbf{Q}_0\|_2 < \frac{1-\mu^2}{\sqrt{1-\mu^2}+(\mu+1)f(\mu,\eta,k)}, \quad (4)$$

and the following two inequalities hold

$$T \geq \frac{\log(\epsilon/\nu)}{\log\left[\mu/\left(\sqrt{1-\nu^2}-f(\mu,\eta,k)\nu\right)\right]},$$
$$B \geq \frac{ck^2\lambda_1^2[(s+2\gamma)\log p + \log T]}{\epsilon^2\eta^2\lambda_k^2},$$

where c is a universal constant, then with probability at least $1 - s^{-10}$ the output $\mathbf{Q}_T$ of Algorithm 2 satisfies $\|\mathbf{U}_{k,\perp}^\top\mathbf{Q}_T\|_2 \leq \epsilon$.

Remark 1. From Theorem 1, we observe that: 1) Algorithm 2 succeeds as long as $\lambda_k > (k+1)\lambda_{k+1}$ since there exists η, e.g., $\eta = \frac{\lambda_k-(k+1)\lambda_{k+1}}{4\lambda_k}$ so that $\mu < 1$. 2) $\frac{(k+1)\lambda_{k+1}}{\lambda_k}$ affects the convergence rate and the block size B. A smaller $\frac{(k+1)\lambda_{k+1}}{\lambda_k}$ leads to a smaller μ and a larger η, which implies faster convergence and less samples required. 3) The upper bound related to the initial solution $\mathbf{Q}_0$ mainly depends on μ, which goes to 0 as $\mu \to 1$ and approaches to $\frac{k}{\eta}$ as $\mu \to 0$. In other words, a more accurate initial solution is required when μ is larger. 4) The block size B should be at least $\Theta((s+2\gamma)\log p + \log T)$. Typically, if $s \leq \gamma \leq 2s$, our algorithm can succeed when $B = \Theta(s\log p + \log T)$. Notice that this is typically much smaller than streaming PCA which requires the block size $B = \Theta(p\log p)$.

Remark 2. When $k = d$, η can be set to $\frac{\lambda_d-(d+1)\sigma^2}{4\lambda_d}$ so that $\mu = \frac{1}{2}\left[1 + \frac{(d+1)\sigma^2}{\lambda_d}\right]$ and $f(\mu,\eta,k) = \frac{2+\sqrt{2}}{\sqrt{d}}\mu$. Thus in this case, Algorithm 2 succeeds as long as σ^2 – the covariance of the noise – is less than $\frac{\lambda_d}{d+1}$.

Remark 3. When $k = 1$, η can be set to $\frac{\lambda_1-2\lambda_2}{4\lambda_1}$ so that $\mu = \frac{1}{2} + \frac{\lambda_2}{\lambda_1}$ and $f(\mu,\eta,k) = (2+\sqrt{2})\mu$. Notice that Algorithm 2 now becomes a block-wise stochastic version of TPower. When $\gamma = s$, Yuan & Zhang (2013) proved that TPower succeeds when $\lambda_1 > \sqrt{5}\lambda_2$, while our analysis leads to a slightly better result of $\lambda_1 > 2\lambda_2$.

The following theorem provides the performance guarantee of Algorithm 3 that extracts PCs via iterative deflation. Note that the errors of the first $t-1$ estimated PCs $\mathbf{q}^{(1)}, \cdots, \mathbf{q}^{(t-1)}$ may propagate to the estimate of $\mathbf{q}^{(t)}$.

Theorem 2. *Let $\eta > 0$, $0 < \epsilon < \frac{\sqrt{2}}{2}$, $\gamma_i \geq s$ for $i = 1, \cdots, k$. Let $\{\epsilon_1, \epsilon_2, \cdots, \epsilon_k\}$ be such that $\epsilon_k = \epsilon, \epsilon_{k-1} = \frac{\eta\lambda_k\epsilon_k}{20\lambda_1 k}, \epsilon_{k-2} = \frac{\eta\lambda_{k-1}\epsilon_{k-1}}{20\lambda_1(k-1)}, \cdots, \epsilon_1 = \frac{\eta\lambda_2\epsilon_2}{40\lambda_1}$. For the i^{th} iteration, let $\mu_i \triangleq \frac{2\lambda_{i+1}+2\eta\lambda_i}{\lambda_i}$, $f(\mu_i,\eta) \triangleq \max\{(2+\sqrt{2})\mu_i, \eta\}$. If the initial solution $\mathbf{q}_0^{(i)}$ satisfies*

$$\nu_i \triangleq \sqrt{1-|\mathbf{u}_i^\top\mathbf{q}_0^{(i)}|^2} < \frac{1-\mu_i^2}{\sqrt{1-\mu_i^2}+(\mu_i+1)f(\mu_i,\eta)},$$

and the following two inequalities hold

$$T_i \geq \frac{\log(\epsilon_i/\nu_i)}{\log\left[\mu_i/\left(\sqrt{1-{\nu_i}^2}-f(\mu_i,\eta)\nu_i\right)\right]},$$
$$B_i \geq \frac{c\lambda_1^2[(s+2\gamma_i)\log p + \log(kT_i)]}{\epsilon_i^2\eta^2\lambda_i^2},$$

where c is a universal constant, then $|\mathbf{u}_i^\top\mathbf{q}^{(i)}| \geq \sqrt{1-\epsilon_i^2}$ holds with probability at least $1 - s^{-10}$.

Let $\mathbf{Q} \triangleq [\mathbf{q}^{(1)}, \cdots, \mathbf{q}^{(k)}]$, one can easily verify that Theorem 2 implies $\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}\|_2 \leq 2\epsilon$. In the experiments, we empirically show that Algorithm 3 can generates good estimates when k is relatively small, e.g., $k = 8$.

4.2. Streaming Sparse ECA

We now present the performance guarantee of streaming sparse ECA. We start with the following theorem which states that the population multivariate Kendall's tau statistic $\mathbf{K}$ and the scatter matrix $\mathbf{\Sigma}$ under the elliptical model share the same eigenspace.

Theorem 3. *(Marden, 1999; Croux et al., 2002; Han & Liu, 2013a) Let $EC_p(\mu, \mathbf{\Sigma}, \xi)$ be a continuous elliptical distribution and $\mathbf{K}$ be the population multivariate Kendall's tau statistic. Then if rank$(\mathbf{K}) = q$ and $\lambda_j(\mathbf{\Sigma}) \neq \lambda_k(\mathbf{\Sigma})$ for any $j \neq k \in \{1, \cdots, q\}$, we have*

$$\mathbf{u}_j(\mathbf{\Sigma}) = \mathbf{u}_j(\mathbf{K}) \text{ and } \lambda_j(\mathbf{K}) = \mathbb{E}\left[\frac{\lambda_j(\mathbf{\Sigma}) y_j^2}{\sum_{i=1}^q \lambda_i(\mathbf{\Sigma}) y_i^2}\right],$$

where $\mathbf{u}_j(\cdot)$ is the eigenvector corresponding to the j^{th} largest eigenvalue and $\mathbf{y} \triangleq (y_1, \cdots, y_q)^\top \sim \mathcal{N}(0, \mathbf{I}_q)$.

This theorem states that when $\mathbf{\Sigma}$ has distinct eigenvalues, $\mathbf{\Sigma}$ and $\mathbf{K}$ have the same eigenspace with the same descending order of the eigenvalues. Based on this, our streaming sparse ECA utilizes $\hat{\mathbf{K}}$ defined in Equation (2) to recover the eigenspace of $\mathbf{\Sigma}$. We here abuse the notations and let $\lambda_k(\mathbf{K})$ be the k^{th} largest eigenvalue of $\mathbf{K}$ and $\mathbf{U}_k$ be the matrix consisting of the eigenvectors of $\mathbf{K}$ corresponding to its k largest eigenvalues, and suppose that $\|\text{diag}(\mathbf{U}_k \mathbf{U}_k^\top)\|_0 \leq s$. The following theorem states the theoretical guarantee of the streaming sparse ECA.

Theorem 4. *For $\eta > 0$, $0 < \epsilon < 1$, and $\gamma \geq s$, let $\mu \triangleq \frac{(k+1)\lambda_{k+1} + 2\eta\lambda_k}{\lambda_k}$, $f(\mu, \eta, k) \triangleq \max\{\frac{(2+\sqrt{2})\mu}{\sqrt{k}}, \frac{\eta}{k}\}$. If the initial solution $\mathbf{Q}_0$ satisfies that*

$$\nu \triangleq \|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_0\|_2 < \frac{1 - \mu^2}{\sqrt{1 - \mu^2} + (\mu + 1) f(\mu, \eta, k)},$$

and the following two inequalities hold

$$T \geq \frac{\log(\epsilon/\nu)}{\log\left[\mu / \left(\sqrt{1 - \nu^2} - f(\mu, \eta, k)\nu\right)\right]},$$

$$B \geq \frac{ck^2 \left(1 + \lambda_1(\mathbf{K})\right)^2 \left[(s + 2\gamma)\log p + \log T\right]}{\epsilon^2 \eta^2 \lambda_k(\mathbf{K})^2},$$

where c is a universal constant, then with probability at least $1 - s^{-10}$ the output $\mathbf{Q}_T$ of Algorithm 4 satisfies $\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_T\|_2 \leq \epsilon$.

Notice that all the bounds in Theorem 4 relates to the eigenvalues of $\mathbf{K}$. The connection between the eigenvalues of $\mathbf{K}$ and $\mathbf{\Sigma}$ has been shown in (Han & Liu, 2013a), namely, $\lambda_j(\mathbf{K}) = \Theta(\lambda_j(\mathbf{\Sigma})/\text{tr}(\mathbf{\Sigma}))$ or each j, when $\|\mathbf{\Sigma}\|_F \log p = \text{tr}(\mathbf{\Sigma}) \cdot o(1)$, e.g., the condition number of $\mathbf{\Sigma}$ is upper bounded by a constant.

5. Initialization

As shown in Theorem 1, 2 and 4, both of streaming sparse PCA and ECA require an initial solution whose estimation error is bounded by Equation (4) which involves the intrinsic properties of $\mathbf{\Sigma}$ and the number of PCs one wants to extract. Therefore, how to find such an initial solution is an important issue. Yuan & Zhang (2013) proposed to adaptively select the desired sparsity γ in TPower, i.e., one can run TPower with a relatively large γ to generate the initial solution and then rerun the algorithm with a smaller γ, while (Han & Liu, 2013a) computed the initial solution by FPS (Vu et al., 2013). In streaming PCA, Mitliagkas et al. (2013) showed that the initial solution could be generated by randomly drawing k vectors from the standard Gaussian distribution.

Due to limitations of space, we mainly discuss the initialization issue for streaming sparse PCA. This can be easily generalized to streaming sparse ECA. The idea is to run streaming PCA (streaming sparse PCA with $\gamma = p$) on several blocks of the collected samples and then apply the truncation operation on its output to generate the initial solution. This is summarized in Algorithm 5.

Algorithm 5 Finding Initial Solution

Input: Samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots\}$, block size B, parameter $\bar{\theta}$ and γ.

Procedure:

1) Run streaming PCA on several blocks so that the output $\hat{\mathbf{Q}}_0$ satisfies $\|\mathbf{U}_{k,\perp}^\top \hat{\mathbf{Q}}_0\|_2 \leq \bar{\theta}$;
2) Initialize $\tilde{\mathbf{S}}_0 = 0$;
for $t = 1$ to B **do**
  2) $\tilde{\mathbf{S}}_0 = \tilde{\mathbf{S}}_0 + \frac{1}{B}\mathbf{x}_t \mathbf{x}_t^\top \hat{\mathbf{Q}}_0$;
end for
3) $\mathbf{S}_0 = \text{Truncate}(\tilde{\mathbf{S}}_0, \gamma)$;
4) QR-decomposition: $\mathbf{S}_0 = \mathbf{Q}_0 \mathbf{R}_0$;
5) Return $\mathbf{Q}_0$.

Theorem 5 provides the performance guarantee of Algorithm 5. Note that the block size B should be $\Omega(p)$ to achieve consistency. However, we observe from experiments empirically this algorithm is able to generate acceptable results even when B is much smaller than p, especially when the covariance of noise σ^2 is small.

Theorem 5. *Fix accuracy ϵ with $0 < \epsilon < 1$, and let $\bar{\theta} = \sqrt{\frac{\epsilon^2}{\epsilon^2 + 8(3+2\sqrt{2})(1 + \frac{k\lambda_{k+1}}{\lambda_k})^2}}$, then $\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_0\|_2 \leq \epsilon$ holds*

with probability at least $1-d^{-10}$ *if* $\gamma \geq s$, $\|\mathbf{U}_{k,\perp}^\top \hat{\mathbf{Q}}_0\|_2 \leq \bar{\theta}$ *and* $B \geq \frac{ck^2[\lambda_1(\mathbf{A})^4 d+(2\lambda_1(\mathbf{A})+\sigma)^2\sigma^2 p]}{(\lambda_k(\mathbf{A})^2+\sigma^2)^2\bar{\theta}^2}$, *where* c *is a universal constant.*

6. Proof Sketch

We now sketch the proofs of the theorems presented in Section 4. We start with Theorem 1. The key ingredient of our proofs is to build a connection between the estimation error on the τ^{th} iteration $\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2$ and that on the $\tau+1^{th}$ iteration $\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_{\tau+1}\|_2$. If this error decreases on each iteration, then one can compute the number of iterations needed for the estimation error to be less than ϵ.

In the rest of this section, the empirical covariance of samples $\{\mathbf{x}_{B\tau+1}, \cdots, \mathbf{x}_{B(\tau+1)}\}$ is denoted by $\hat{\mathbf{\Sigma}}_\tau$, and the row supports of $\mathbf{U}_k$, $\mathbf{Q}_\tau$ and $\mathbf{Q}_{\tau+1}$ are represented by $\mathcal{S}$, $\mathcal{F}_\tau$ and $\mathcal{F}_{\tau+1}$, respectively. We let $\mathcal{F} \triangleq \mathcal{S} \cup \mathcal{F}_\tau \cup \mathcal{F}_{\tau+1}$ and let $\mathbf{X}(i,j)$ be the $(i,j)^{th}$ entry of matrix $\mathbf{X}$. For a $p \times p$ squared matrix, e.g., $\hat{\mathbf{\Sigma}}_\tau$, let $\hat{\mathbf{\Sigma}}_{\tau,\mathcal{F}}$ denote the matrix whose $(i,j)^{th}$ entry equals $\hat{\mathbf{\Sigma}}_\tau(i,j)$ if the row index i and the column index j are both in $\mathcal{F}$, and 0 otherwise. For a $p \times k$ matrix, e.g., $\tilde{\mathbf{S}}_{\tau+1}$, let $\tilde{\mathbf{S}}_{\tau+1,\mathcal{F}}$ denote the matrix whose $(i,j)^{th}$ entry equals $\tilde{\mathbf{S}}_{\tau+1}(i,j)$ if the row index $i \in \mathcal{F}$, and 0 otherwise. Then one can easily verify that $\tilde{\mathbf{S}}_{\tau+1,\mathcal{F}} = \hat{\mathbf{\Sigma}}_{\tau,\mathcal{F}}\mathbf{Q}_\tau$ and $\mathbf{S}_{\tau+1} = \text{Truncate}(\tilde{\mathbf{S}}_{\tau+1,\mathcal{F}}, \gamma)$. Let $\tilde{\mathbf{S}}_{\tau+1,\mathcal{F}} = \mathbf{Q}_{\tau+1,\mathcal{F}}\mathbf{R}_{\tau+1,\mathcal{F}}$ be the QR decomposition of $\tilde{\mathbf{S}}_{\tau+1,\mathcal{F}}$. In order to establish a relationship between $\mathbf{Q}_\tau$ and $\mathbf{Q}_{\tau+1}$, we first connect $\mathbf{Q}_\tau$ to $\mathbf{Q}_{\tau+1,\mathcal{F}}$:

Lemma 1. *Let* $\xi = \sup_{\mathcal{F}:|\mathcal{F}|\leq s+2\gamma} \|\hat{\mathbf{\Sigma}}_{\tau,\mathcal{F}} - \mathbf{\Sigma}_{\tau,\mathcal{F}}\|_2$, *then if* $\tilde{\mathbf{S}}_{\tau+1,\mathcal{F}}$ *has full column rank, we have*

$$\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_{\tau+1,\mathcal{F}}\|_2^2 \leq \frac{[\lambda_{k+1}\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2 + \xi]^2}{[\lambda_{k+1}\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2 + \xi]^2 + [\lambda_k\sqrt{1-\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2^2} - \xi]^2}.$$

We now connect $\mathbf{Q}_{\tau+1,\mathcal{F}}$ to $\mathbf{Q}_{\tau+1}$ based on the perturbation analysis of the QR decomposition and the error analysis of the row truncation operation.

Lemma 2. *Let* $\mathbf{Q}_{\tau+1,\mathcal{F},\perp}$ *be an orthonormal matrix such that matrix* $[\mathbf{Q}_{\tau+1,\mathcal{F}}, \mathbf{Q}_{\tau+1,\mathcal{F},\perp}]$ *is orthogonal and let* $\xi \triangleq \sup_{\mathcal{F}:|\mathcal{F}|\leq s+2\gamma} \|\hat{\mathbf{\Sigma}}_{\tau,\mathcal{F}} - \mathbf{\Sigma}_{\tau,\mathcal{F}}\|_2$, *then if* $\gamma \geq s$ *and*

$$\frac{\sqrt{k}[\lambda_{k+1}\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2 + \xi]}{\lambda_k\sqrt{1-\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2^2}} < \frac{1}{c},$$

we have

$$\|\mathbf{Q}_{\tau+1,\mathcal{F},\perp}^\top(\mathbf{Q}_{\tau+1} - \mathbf{Q}_{\tau+1,\mathcal{F}})\|_2 \leq \frac{k[\lambda_{k+1}\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2 + \xi]}{\lambda_k\sqrt{1-\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2^2} - c\sqrt{k}[\lambda_{k+1}\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2 + \xi]},$$

where $c = 2+\sqrt{2}$.

We then prove that $\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_{\tau+1}\|_2$ can be upper bounded by $\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_{\tau+1,\mathcal{F}}\|_2 + \|\mathbf{Q}_{\tau+1,\mathcal{F},\perp}^\top(\mathbf{Q}_{\tau+1} - \mathbf{Q}_{\tau+1,\mathcal{F}})\|_2$. Therefore, by combining the two lemmas above, let $\nu \triangleq \|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_0\|_2$, we can show that the following holds with probability at least $1 - \frac{s^{-10}}{T}$,

$$\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_{\tau+1}\|_2 \leq \frac{\mu\|\mathbf{U}_{k,\perp}^\top \mathbf{Q}_\tau\|_2}{\sqrt{1-\nu^2} - f(\mu,\eta,k)\nu},$$

if $B \geq \frac{ck^2\lambda_1^2[(s+2\gamma)\log p + \log T]}{\epsilon^2\eta^2\lambda_k^2}$ and Inequality (4) holds. Thus Theorem 1 is established.

Theorem 2 can be proved by analyzing the error propagation in the iterative deflation, combining with the results derived from Theorem 1 with $k=1$. For Theorem 4, as we have discussed in Section 3, streaming sparse ECA uses $\hat{\mathbf{K}}$ as the estimator of the multivariate Kendall's tau matrix instead of the empirical covariance of the received samples. Therefore, Theorem 4 can be established following the proofs discussed above, together with an upper bound of $\sup_{\mathcal{F}:|\mathcal{F}|\leq s+2\gamma} \|(\hat{\mathbf{K}}_\tau - \mathbf{K}_\tau)_\mathcal{F}\|_2$ shown below.

Theorem 6. *Let* $\mathbf{x}_1, \cdots, \mathbf{x}_n$ *be* n *independent observations of random vector* $\mathbf{x} \sim EC_p(\mu, \mathbf{\Sigma}, \xi)$. *Let* $\mathbf{K}$ *be multivariate Kendall's tau matrix of* $\mathbf{x}$ *and* $\hat{K}$ *be the empirical estimation of* $\mathbf{K}$ *which is defined as Equation* (2). *Then there exists a universal constant* c *such that the following holds with probability at least* $1 - \frac{s^{-10}}{T}$,

$$\sup_{\|\mathbf{v}\|_2=1, \|\mathbf{v}\|_0=s} |\mathbf{v}^\top(\hat{\mathbf{K}} - \mathbf{K})\mathbf{v}| \leq c\left(\min\left\{\frac{4\lambda_1(\mathbf{K})}{q\lambda_q(\mathbf{K})}, 1\right\} + \|\mathbf{K}\|_2\right)\sqrt{\frac{s\log p + \log T}{n}},$$

for parameter T, *where* $q = rank(\mathbf{K})$.

Theorem 6 implies that $\sup_{\mathcal{F}:|\mathcal{F}|\leq s+2\gamma} \|(\hat{\mathbf{K}}_\tau - \mathbf{K}_\tau)_\mathcal{F}\|_2 \leq c\,(1+\|\mathbf{K}\|_2)\sqrt{\frac{(s+2\gamma)\log p + \log T}{n}}$ with high probability. Then by embedding this inequality into Lemma 1, Lemma 2, we obtain Theorem 4.

7. Experiments

We investigate the performance of our algorithms on a variety of simulated and real-world datasets. All the algorithms mentioned below are implemented in Python. The experiments are conducted on a desktop PC with an i7 3.4GHz CPU and 4G memory.

7.1. Synthetic Data

We first illustrate the empirical performance of our streaming sparse PCA (Algorithm 2) with synthetic datasets. For

the spike model, we consider two data generating schemes. The first one, which is similar to (Yuan & Zhang, 2013), is that matrix $\mathbf{A} \in \mathbb{R}^{p\times 2}$ satisfies that $\mathbf{A}\mathbf{A}^\top = \lambda_1 \mathbf{v}_1\mathbf{v}_1^\top + \lambda_2 \mathbf{v}_2\mathbf{v}_2^\top$, where $\mathbf{v}_1 \in \mathbb{R}^p$ and $\mathbf{v}_2 \in \mathbb{R}^p$ are two sparse vectors satisfying that

$$v_{1i} = \begin{cases} \frac{1}{\sqrt{10}} & 1 \le i \le 10 \\ 0 & \text{otherwise,} \end{cases} \quad v_{2i} = \begin{cases} \frac{1}{\sqrt{10}} & 11 \le i \le 20 \\ 0 & \text{otherwise,} \end{cases}$$

and $\lambda_1 = 5, \lambda_2 = 3$. The second one constructs matrix $\mathbf{A} \in \mathbb{R}^{p\times d}$ by randomly generating two orthogonal matrices $\mathbf{U} \in \mathbb{R}^{p\times d}$ and $\mathbf{V} \in \mathbb{R}^{d\times d}$ such that $\|\mathrm{diag}(\mathbf{U}\mathbf{U}^\top)\|_0 = s$, and then setting $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$ where $\mathbf{S} \in \mathbb{R}^{d\times d}$ is a diagonal matrix whose diagonal entries are computed according to the chi-square density $f(x) = \frac{x^{-\frac{1}{2}}e^{-\frac{1}{2}x}}{\sqrt{2}\Gamma(\frac{1}{2})}$, i.e., $S_{ii} = f(\frac{i}{20})$ for $i = 1, \cdots, d$. In the experiments, we repeat each test 20 times and report the average results.

In the first experiment, we make a comparison between streaming sparse PCA, streaming PCA, FPS and online sparse PCA (Mairal et al., 2010) for a relative small p. We use three measurements to evaluate their performance, namely, the subspace distance defined in Equation (3), the expressed variance in (Xu et al., 2013) defined as $\sum_{i=1}^k \mathbf{q}_i^\top \mathbf{A}\mathbf{A}^\top \mathbf{q}_i / \sum_{i=1}^k \lambda_i(\mathbf{A}\mathbf{A}^\top)$, and the sparsity defined as $\sum_{i=1}^k |\{j : |q_{ij}| > t\}|$, where $\{\mathbf{q}_1, \cdots, \mathbf{q}_k\}$ are their estimation and t is a threshold which is set to 0.001. Figure 1(a) shows the results in the first scheme, where the leading PC is extracted. We observe that streaming sparse PCA performs similarly to online sparse PCA and outperforms FPS and streaming PCA. The running time of streaming sparse PCA is much less than that of FPS and online sparse PCA because FPS needs to solve a SDP problem via an ADMM algorithm and online sparse PCA needs to solve the elastic-net (Zou & Hastie, 2005) in each iteration, whereas our algorithm only requires the truncation operation and the QR decomposition. Figure 1(b) presents the results in the second scheme, where we extract the leading ten PCs. It can be observed that FPS and streaming sparse PCA have better performance than the other methods. Notice that streaming sparse PCA runs more than 100 times faster than FPS and FPS needs to store the $p \times p$ covariance matrix while streaming sparse PCA only maintains a $p \times k$ matrix.

In the second experiment, we mainly compare streaming PCA and streaming sparse PCA in the high dimensional regime since the other two methods are too slow in this setup. The estimation error is measured by the subspace distance. The samples are generated according to the second scheme described above and we extract the leading 10 PCs. Figure 2(a) shows their estimation errors when $n = 1000$, $B = 100$ and p varies from 1000 to 50000. Clearly, streaming sparse PCA succeeds in recovering the sparse PCs but streaming PCA fails when p is large, which

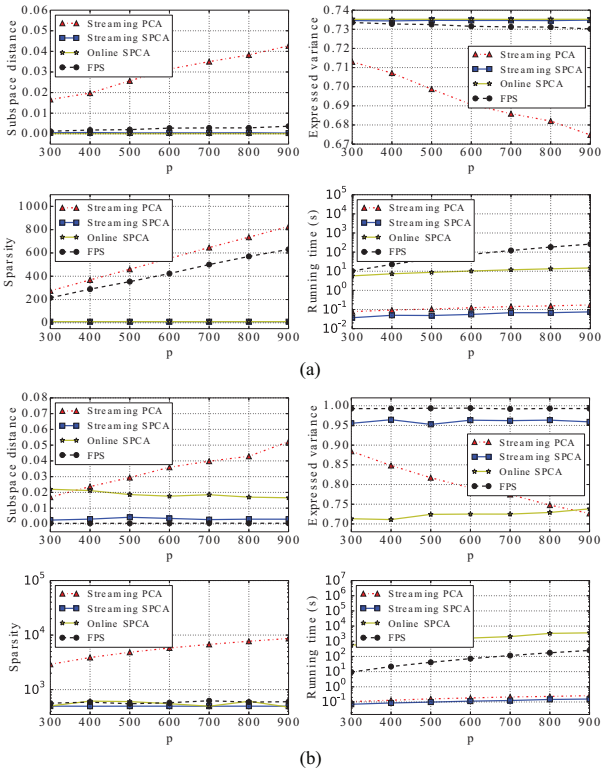

Figure 1. (a) Scheme 1 with $n = 1000$, $\sigma^2 = 0.5$, $B = 100$, $\gamma = 10$ and $k = 1$. (b) Scheme 2 with $n = 1000$, $d = 10$, $\sigma^2 = 0.2$, $B = 100$, $s = 50$, $\gamma = 50$ and $k = 10$.

is consistent with the theoretical results shown in Theorem 1. Figure 2(b) presents the number of samples required for recovering the leading 10 PCs of accuracy $\epsilon \le 0.03$ where $B = 100$, which shows that streaming PCA requires much more samples than streaming sparse PCA to achieve the same accuracy. Figure 2(c) shows the effect of block size B on their performance, where $n = 1000$ and $p = 5000$. When B is too small, e.g., less than 60, streaming sparse PCA does not guarantee the convergence to the true PCs, which agrees with Theorem 1. Figure 2(d) demonstrates the probability of success of streaming sparse PCA, measured by the fraction of the trials in which the estimation error is less than 0.05. We here set $n = 1000$ and $B = 100$. We can observe that the tolerance of σ^2 decreases as p increases when B is fixed. This is because both of p and σ^2 affect the lower bound of B as shown in Theorem 1.

We now compare our streaming sparse PCA/ECA with ECA (Han & Liu, 2013a) under the elliptical model. In the following experiments, the samples are independently drawn from $EC_p(0, \mathbf{\Sigma}, \xi)$. Here, $\mathbf{\Sigma}$ is constructed according to $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^\top + \mathbf{I}_p$ where $\mathbf{A}$ is generated following the first scheme described above, and ξ follows the chi-distribution with degree of freedom p or the F-distribution with degrees of freedom p and 1. We estimate the leading

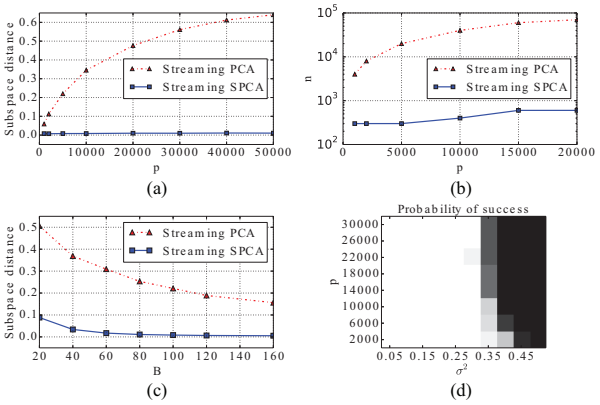

Figure 2. (a), (b) and (c) show the comparison between streaming PCA and streaming sparse PCA, where $\sigma^2 = 0.2$, $k = 10$ and $\gamma = s = 100$. (d) presents the effect of σ^2 on the performance of streaming sparse PCA.

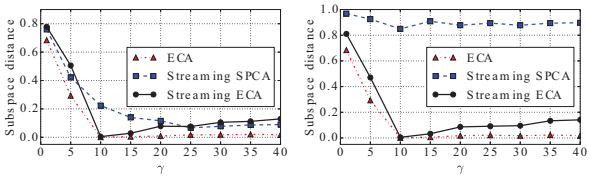

Figure 3. Comparison between ECA, streaming sparse ECA, and streaming sparse PCA. Random variable ξ follows (Left) the chi-distribution χ_p, and (Right) the F-distribution $F(d, 1)$.

eigenvector of $\mathbf{\Sigma}$. Figure 3 plots the subspace distances between their estimation and the true PC against parameter γ for these three methods, where we set $p = 500$, $n = 600$ and $B = 100$, which shows that streaming sparse ECA performs similarly to ECA and is better than streaming sparse PCA especially when γ is close to s. Typically, streaming sparse ECA is much faster than ECA when p or n is large, because ECA computes the second order U-statistic estimator and uses FPS to construct a good initial solution. In this experiment, on the average, ECA needs 30s to generate its solution while streaming sparse ECA only requires 30 milliseconds. Figure 4 shows the comparison between the "streaming" versions of PCA, ECA and sparse PCA in the high dimensional setting where $p = 10000$, $n = 2000$ and $B = 100$. Clearly, streaming sparse ECA outperforms the other two methods in the elliptical model.

7.2. Real-world Datasets

We use two large datasets, the NIPS paper dataset and the NYTimes news articles dataset, both available from the UCI Machine Learning Repository (Bache & Lichman), to compare the empirical performance of streaming sparse PCA, streaming PCA and large-scale sparse PCA (Zhang & El Ghaoui, 2011). Both datasets record word occurrences in the form of bags-of-words. The NIPS dataset contains 1500 articles and a dictionary of 12419 words. The NYTimes dataset contains 300000 articles and a dictionary of 102660 words. We evaluate the performance by the fraction of explained variance which is defined as $\mathrm{tr}(\mathbf{U}^\top \mathbf{X}\mathbf{X}^\top \mathbf{U})/\mathrm{tr}(\mathbf{X}\mathbf{X}^\top)$ where $\mathbf{X} \in \mathbb{R}^{p\times n}$ is the sample matrix and $\mathbf{U} \in \mathbb{R}^{p\times k}$ is the output. Parameters B and γ in streaming sparse PCA are set to 300 and 500, respectively. The leading k PCs are computed via the iterative deflation. Figure 5 plots the explained variance against k. We observe that streaming sparse PCA performs similarly to streaming PCA and is better than large-scale sparse PCA, in terms of the expressed variance. One advantage of streaming sparse PCA is that it is computationally efficient and guarantees the sparsity of its solution, i.e., $\gamma = 500$, without much loss of performance compared to streaming PCA.

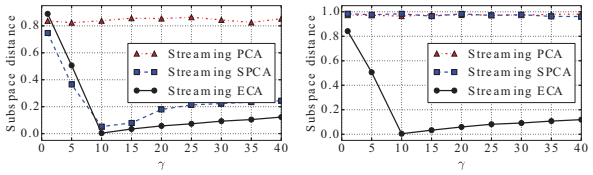

Figure 4. Comparison between streaming PCA and streaming sparse PCA/ECA. Random variable ξ follows (Left) the chi-distribution χ_p, and (Right) the F-distribution $F(d, 1)$.

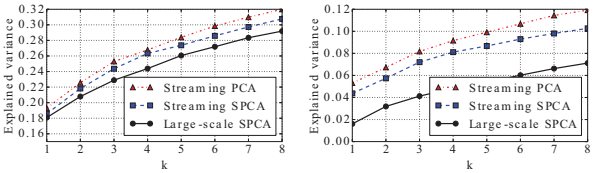

Figure 5. (Left) NIPS dataset. (Right) NYTimes dataset.

8. Conclusion

In this paper, we propose streaming sparse PCA/ECA for dimensionality reduction of high dimensional data generated according to the spike model and the elliptical model, and establish finite sample performance guarantees. The experiments validate that they perform better and are more computationally efficient than the other alternatives to PCA.

Acknowledgments

This work is partially supported by the Ministry of Education of Singapore AcRF Tier Two grants R265000443112 and R265000519112, and A*STAR Public Sector Funding R265000540305.

References

Amini, A. A. and Wainwright, M. J. High-dimensional analysis of semidefinite relaxiation for sparse principal components. *The Annals of Statistics*, 37(5B):2877–2921, 2009.

Arora, R., Cotter, A., Livescu, K., and Srebro, N. Stochastic optimization for PCA and PLS. In *Allerton Conference*, 2012.

Bache, K. and Lichman, M. UCI machine learning repository. URL http://archive.ics.uci.edu/ml.

Birnbaum, A., Johnstone, I. M., Nadler, B., and Paul, D. Minimax bounds for sparse PCA with noisy high-dimensional data. *The Annals of Statistics*, 41(3):1055–1084, 2013.

Brand, M. Incremental singular value decomposition of uncertain data with missing values. In *ECCV*, 2002.

Cai, T., Ma, Z., and Wu, Y. Optimal estimation and rank detection for sparse spiked covariance matrices. *arXiv:1305.3235*, 2014.

Croux, C., Ollila, E., and Oja, H. Sign and rank covariance matrices: statistical properties and application to principal components analysis. *Statistics for Industry and Technology*, pp. 257–269, 2002.

d'Aspremont, A., El Ghaoui, L., Jordan, M. I., and Lanckriet, G. R. A direct formulation for sparse PCA using semidefinite programming. *SIAM review*, 49(3):434–448, 2007.

Fang, K., Kotz, S., and Ng, K. *Symmetric Multivariate and Related Distributions*. Chapman & Hall, 1990.

Han, F. and Liu, H. ECA: High dimensional elliptical component analysis in non-Gaussian distributions. *arXiv:1310.3561*, 2013a.

Han, F. and Liu, H. Optimal rates of convergence for latent generalized correlation matrix estimation in transelliptical distribution. *arXiv:1305.6916v3*, 2013b.

Hult, H. and Lindskog, F. Multivariate extremes, aggregation and dependence in elliptical distributions. *Advances in Applied probability*, 34(3):587–608, 2002.

Johnstone, I. M. and Lu, A. Y. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693, 2009.

Journee, M. and Y. Nesterov, Peter Richtarik, R. Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, pp. 517–553, 2008.

Ma, Z. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.

Mackey, L. Deflation methods for sparse PCA. In *NIPS*, 2008.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

Marden, J. Some robust estimates of principal components. *Statistics and Probability Letters*, 43(4):349–359, 1999.

Mitliagkas, I., Caramanis, C., and Jain, P. Memory limited, streaming PCA. In *NIPS*, 2013.

Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559C–572, 1901.

Shen, D., Shen, H., and Marron, J.S. Consistency of sparse PCA in high dimension, low sample size contexts. *Journal of Multivariate Analysis*, 115:115–317, 2013.

Shen, H. and Huang, J. Z. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99:1015–1034, 2008.

Vu, V. Q. and Lei, J. Minimax rates of estimation for sparse PCA in high dimensions. In *AISTATS*, 2012.

Vu, V. Q., Cho, J., Lei, J., and Robe, K. Fantope projection and selection: A near-optimal convex relaxation of sparse PCA. In *NIPS*, 2013.

Wang, Z., Lu, H., and Liu, H. Nonconvex statistical optimization: minimax optimal sparse PCA in polynomial time. In *NIPS*, 2014.

Warmuth, M. K. and Kuzmin, D. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9:2287–2320, 2008.

Xu, H., Caramanis, C., and Mannor, S. Outlier-robust PCA: the high-dimensional case. *IEEE Transactions on Information Theory*, 59(1):546–572, 2013.

Yuan, X. and Zhang, T. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14:899–925, 2013.

Zhang, Y. and El Ghaoui, L. Large-scale sparse principal component analysis with application to text data. In *NIPS*, 2011.

Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

Zou, H., Hastie, T., and Tibshirani, R. Sparse principal component analysis. In *JCGS*, pp. 265–286, 2006.