A Unified Framework for Outlier-Robust PCA-like Algorithms

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

We propose a unified framework for making a wide range of PCA-like algorithms – including the standard PCA, sparse PCA and non-negative sparse PCA, etc. – robust when facing a constant fraction of arbitrarily corrupted outliers. Our analysis establishes solid performance guarantees of the proposed framework: its estimation error is upper bounded by a term depending on the intrinsic parameters of the data model, the selected PCA-like algorithm and the fraction of outliers. Our experiments on synthetic and real-world datasets demonstrate that the outlier-robust PCA-like algorithms derived from our framework have outstanding performance.

1. Introduction

Principal component analysis (PCA) (Pearson, 1901), arguably the most widely applied dimension reduction method, plays a significant role in data analysis in a broad range of areas including machine learning, statistics, finance, biostatistics and many others. The standard PCA performs the spectral decomposition of the sample covariance matrix, selects the eigenvectors corresponding to the largest eigenvalues, and then constructs a low dimensional subspace based on the selected eigenvectors. It is well known that standard PCA, depending on different applications, may suffer from three weaknesses (Montanari & Richard, 2014; Xu et al., 2013; Johnstone & Lu, 2009): 1) PCA is notoriously fragile to outliers – indeed, its performance can significantly degrade in the presence of even few corrupted samples, due to the quadratic error criterion used; 2) PCA cannot utilize additional information of the principal components: e.g., in certain applications, it is known that the principal components should lie in the positive orthant; 3) its output may lack interpretability since it does not encourage sparse solutions.

Many efforts have been made to mitigate these weaknesses of PCA. In recent years, numerous robust PCA algorithms have been proposed to address the first issue (Devlin et al., 1981; Xu & Yuille, 1995; Yang & Wang, 1999; la Torre & Black, 2003; Dasgupta, 2003; Xu et al., 2013; Feng et al., 2012). Among them, Xu et al. (2013) successfully tackles the case where a *constant fraction* of samples are corrupted in the *high dimensional regime*. Their proposed method, termed HR-PCA (which stands for High-dimensional Robust PCA), is tractable, easily kernelizable, and is able to robustly estimate the principal components even in the face of a constant fraction of outliers and very low signal-to-noise ratio. To overcome the computational issue of HR-PCA, Feng *et al.* (Feng et al., 2012) proposed a deterministic approach (DHR-PCA) that dramatically reduces the computational work. However, neither HR-PCA nor DHR-PCA deals with the last two weaknesses mentioned above.

To address the second weakness, Montanari & Richard (2014) recently proposed a new algorithm called *non-negative PCA* which handles the case that the principal components are known to lie in the positive orthant, and showed that near-optimal non-negative principal components can be extracted in nearly linear time. But similar to the standard PCA, this algorithm is sensitive to outliers. Indeed, the estimated principal components can be far from the true ones in the face of even few outliers.

To address the third weakness, previous works focus on a class of methods called sparse PCA that adapt the standard PCA so that only a few of attributes of the resulting principle components are non-zero (e.g., Vu et al., 2013; Zou et al., 2006; Shen & Huang, 2008; Journee & Y. Nesterov, 2008; Birnbaum et al., 2013; Vu & Lei, 2013; d'Aspremont et al., 2007; Thiao et al., 2010). Some of these methods are based on non-convex optimization formulations (Jolliffe et al., 2003; Moghaddam et al., 2005) while others use ℓ_1-norm regularization (Zou et al., 2006). Recently, Vu *et al.* (Vu et al., 2013) proposed FPS – a convex relaxation for-

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

mulation of sparse principal subspace estimation based on a semi-definite program with a *Fantope* constraint and established theoretical guarantees in the outlier-free regime. Yet, one severe drawback of most sparse PCA algorithms is that the output can be sensitive to the existence of even few outliers. This is clearly undesirable, as in real-world applications, the existence of outliers is ubiquitous. Recently, several robust sparse PCA have been proposed (Croux et al., 2013; Wang & Cheng, 2012; Hubert et al., 2014) to handle outliers, but all of them are only evaluated by experiments and have no theoretical performance guarantees.

This paper is the first attempt to theoretically address these issues of PCA simultaneously. In specific, we propose a general framework for a wide range of PCA-like algorithms to make them provably *robust* to *a constant fraction* of *arbitrary* outliers. Our framework is inspired by HR-PCA (Xu et al., 2013; Feng et al., 2012), but overcomes the drawbacks of HR-PCA and has the capability of converting a non-robust PCA-like algorithm such as non-negative PCA (Montanari & Richard, 2014), sparse PCA (Vu et al., 2013; Papailiopoulos et al., 2013) or non-negative sparse PCA (Asteris et al., 2014), into its outlier-robust variant.

The analysis of our proposed framework is novel and different from that of HR-PCA. We analyze its performance using two performance metrics: the subspace distance and the expressed variance. We show that the subspace distance between its estimated principal components and the ground-truth under the spiked model can be upper bounded by a term depending on the parameters of the spike model, the selected PCA-like algorithm and the fraction of outliers. The analysis of subspace distance in the presence of outliers is new to the best of our knowledge. Moreover, while the analysis of expressed variance for HR-PCA exists in literature, our analysis of the expressed variance of this framework is more general, in that it shows that maximal robustness can be achieved for a wide range of PCA-like algorithms besides HR-PCA. Our numerical experiments results show that when outliers exist, the outlier-robust PCA-like algorithms developed from our framework outperform their non-robust counterparts considerably.

Notation. We use lower-case boldface letters to denote column vectors and upper-case boldface letters to denote matrices. In this paper, $\|\mathbf{M}\|_2$ is the spectral norm, $\|\mathbf{M}\|_*$ is the nuclear norm, $\|\mathbf{M}\|_1$ is the element-wise ℓ_1 norm, $\|\mathbf{M}\|_\infty$ is the element-wise infinity norm, and $\|\mathbf{M}\|_F$ is the Frobenius norm. We use $\|\mathbf{M}\|_0$ to denote the number of non-zero entries in $\mathbf{M}$, and use subscript $(\cdot)$ to represent order statistics of a random variable. For example, let $v_1, \cdots, v_n \in \mathbb{R}$, then $v_{(1)}, \cdots, v_{(n)}$ is a permutation of $v_1, \cdots, v_n$ in a non-decreasing order. For matrix $\mathbf{X}$, the first k singular values of $\mathbf{X}$ are denoted by $\lambda_1(\mathbf{X}), \cdots, \lambda_k(\mathbf{X})$.

2. Algorithms

In this section, we present our framework for outlier-robust PCA-like algorithms. We first describe the problem setup and necessary assumptions, and then show the details of the algorithm along with the key intuition underlying it.

2.1. Problem Setup

Suppose there are n samples $\mathcal{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_n \in \mathbb{R}^p\}$ which consist of t authentic samples $\mathbf{z}_1, \cdots, \mathbf{z}_t \in \mathbb{R}^p$ and $n-t$ outliers $\mathbf{o}_1, \cdots, \mathbf{o}_{n-t} \in \mathbb{R}^p$. The outliers are *arbitrary*. We denote the fraction of outliers by $\rho = (n-t)/n$ and assume that $\rho < 0.5$. The authentic samples $\mathbf{z}_i$ are generated according to $\mathbf{z}_i = \mathbf{A}\mathbf{x}_i + \mathbf{n}_i$ where $\mathbf{x}_i \in \mathbb{R}^d$ are i.i.d. samples of a random vector $\mathbf{x}$ with mean 0 and variance $\mathbf{I}_d$ and $\mathbf{n}_i$ are independent realizations of standard Gaussian $\mathcal{N}(0, \mathbf{I}_p)$. The matrix $\mathbf{A} \in \mathbb{R}^{p\times d}$ and the distribution of $\mathbf{x}$ (denoted by ν) are unknown. The covariance of $\mathbf{z}$ is denoted by $\mathbf{\Sigma}$. Since $\mathbf{z} = \mathbf{A}\mathbf{x}+\mathbf{n}$, $\mathbf{\Sigma} = \mathbb{E}[\mathbf{z}\mathbf{z}^\top] = \mathbf{A}\mathbf{A}^\top + \mathbf{I}_p$. We denote the one-dimensional marginal of ν along direction $\mathbf{v} \in \mathcal{S}_d$ by $\bar{\nu}_\mathbf{v}$, and assume that $\bar{\nu}_\mathbf{v}(\{0\}) < 0.5$ for all $\mathbf{v} \in \mathcal{S}_d$ and it is sub-Gaussian, i.e., there exists $\theta > 0$ such that $\bar{\nu}_\mathbf{v}((-\infty, x] \cup [x, +\infty)) \leq \exp(1 - x^2/\theta)$ for all $x > 0$. Clearly, both assumptions are satisfied if ν is Gaussian.

We make the following two assumptions: 1) $\mathbf{A}$ is full row rank and $n > d$. This essentially means the intrinsic dimension of the authentic samples (ignoring the noise) is indeed d. 2) The projection $\mathbf{\Pi}(k) = \mathbf{U}(k)\mathbf{U}(k)^\top$ onto the subspace spanned by the eigenvectors $\mathbf{U}(k)$ of $\mathbf{\Sigma}$ corresponding to its k largest eigenvalues satisfies $\|\mathbf{\Pi}(k)\|_0 \leq \beta^2$, where $\|\mathbf{\Pi}(k)\|_0$ is the number of nonzero entries of $\mathbf{\Pi}(k)$. Our goal is to approximately recover $\mathbf{\Pi}(k)$ even though the samples contain a non-negligible fraction of arbitrary outliers. For convenience, we let $\mathbf{\Pi} \triangleq \mathbf{\Pi}(k)$ in the following sections. Throughout the paper, "with high probability" means with probability at least $1 - c\max\{p^{-10}, n^{-10}\}$ for some constant c.

2.2. General Formulation of PCA-like Algorithms

Many kinds of PCA-like algorithms have been proposed in recent decades, e.g., sparse PCA (Zou et al., 2006; Papailiopoulos et al., 2013), non-negative PCA (Montanari & Richard, 2014), etc., which play a significant role in machine learning, computer vision, statistics and data analysis. In this section, we consider a general formulation as shown below for a wide range of these algorithms:

$$\max_{\mathbf{X}\in\mathcal{C}} \quad \langle \hat{\mathbf{\Sigma}}, \mathbf{X} \rangle - \mu\|\mathbf{X}\|_1, \tag{1}$$

where $\hat{\mathbf{\Sigma}}$ is the empirical sample covariance matrix, $\mathcal{C}$ includes the constraints imposed on $\mathbf{X}$, and μ is the weight of the regularization term. Typically, μ is less than a certain

universal constant. To see that this formulation can model most PCA-like algorithms proposed in literature, let k be the number of the principal components one wants to extract and $\mathcal{F}(k)$ be the set $\{\mathbf{X} : 0 \preceq \mathbf{X} \preceq \mathbf{I}_p, \mathrm{tr}(\mathbf{X}) = k\}$ which includes the matrices that lie in the convex hull of all feasible projection matrices. Thus, the following algorithms are all equivalent to Formulation (1) for appropriate $k, \mathcal{C}$ and μ:

1. Standard PCA (Pearson, 1901): $k = d, \mathcal{C} = \mathcal{F}(k)$ and $\mu = 0$;
2. Non-negative PCA (Montanari & Richard, 2014): $k = 1, \mathcal{C} = \{\mathbf{u}\mathbf{u}^\top : \|\mathbf{u}\|_2 \leq 1, \mathbf{u} \geq \mathbf{0}\}$ and $\mu = 0$;
3. Sparse PCA (Papailiopoulos et al., 2013): $k = 1, \mathcal{C} = \{\mathbf{u}\mathbf{u}^\top : \|\mathbf{u}\|_0 \leq \gamma, \|\mathbf{u}\|_2 \leq 1\}$ and $\mu = 0$;
4. Fantope projection and selection (FPS) (Vu et al., 2013): $k = d, \mathcal{C} = \mathcal{F}(k)$ and $\mu \asymp \sqrt{\frac{\log p}{n}}$;
5. Non-negative sparse PCA (Asteris et al., 2014): $k = 1, \mathcal{C} = \{\mathbf{u}\mathbf{u}^\top : \|\mathbf{u}\|_0 \leq \gamma, \|\mathbf{u}\|_2 \leq 1, \mathbf{u} \geq \mathbf{0}\}$ and $\mu = 0$;
6. Large-scale sparse PCA (Zhang & El Ghaoui, 2011): $k = 1, \mathcal{C} = \{\mathbf{X} : \mathbf{X} \succeq 0, \mathrm{tr}(\mathbf{X}) = 1\}$, $\mu > 0$.

Since the feasible set $\mathcal{C}$ in (1) may be non-convex, the global optimum of (1) may not be achievable. Therefore, there are two important issues: 1) whether a PCA-like algorithm can probably find an optimal or near-optimal solution of (1), and 2) whether its solution converges to the ground truth. We call the PCA-like algorithms that can find optimal or near-optimal solutions of (1) "workable" algorithms, formally defined as:

Definition 1. *A PCA-like algorithm is (η, γ)-workable if there exist positive numbers $\eta \leq 1$ and $\gamma \leq p$ such that with high probability its output $\hat{\mathbf{X}}$ satisfies $\|\hat{\mathbf{X}}\|_0 \leq \gamma^2$ and*

$$\langle \hat{\mathbf{\Sigma}}, \hat{\mathbf{X}} \rangle - \mu \|\hat{\mathbf{X}}\|_1 \geq (1 - \eta) \left[\langle \hat{\mathbf{\Sigma}}, \mathbf{\Pi} \rangle - \mu \|\mathbf{\Pi}\|_1 \right].$$

Note that η indicates the accuracy of the solution $\hat{\mathbf{X}}$, e.g., $\eta = 0$ means $\hat{\mathbf{X}}$ is optimal, while $\eta = 0.5$ means the cost value corresponding to $\hat{\mathbf{X}}$ is half of the optimum. Parameter γ bounds the sparsity of $\hat{\mathbf{X}}$. For the first five algorithms mentioned above, previous works have proved that all of these algorithms are workable. In particular, $\eta = 0, \gamma = p$ for standard PCA and FPS, $0 < \eta < 1, \gamma = p$ for non-negative PCA, and $0 < \eta < 1, \gamma \ll p$ for sparse PCA and non-negative sparse PCA. For large-scale sparse PCA, no performance guarantees are known, but our experiments show that this algorithm can still be put into our framework to achieve robustness.

2.3. Outlier-Robust PCA-like Algorithm

Our framework is inspired by HR-PCA (Xu et al., 2013). Therefore, before presenting its details, we briefly explain the intuition behind HR-PCA. HR-PCA iteratively performs PCA to compute principal components (PCs) and then randomly removes one point with a probability proportional to its magnitude after projected on the found PCs. HR-PCA works for the following intuitive reasons. In each iteration, a PC is computed either due to true samples which implies it is a "good" direction; or due to large outliers in which case the random removal scheme will remove an outlier with high probability. Thus, for at least one iteration, the algorithm will find a good direction, say $\mathbf{w}^t$. Among all the directions found in the algorithm, the final output of HR-PCA is the one with the largest *Robust Variance Estimator* (RVE). RVE measures the projection variance of the $(n-\hat{t})$-smallest points: A large RVE means that that many of the points have a large variance in this direction, while a small RVE indicates otherwise. This makes sure that the final output is close to $\mathbf{w}^t$, and hence a good direction. A variant of HR-PCA is called deterministic HR-PCA or DHR-PCA (Feng et al., 2012). Instead of removing one point, DHR-PCA decreases the weights of all samples according to their magnitudes after projected on the found PCs in each iteration to reduce computational cost.

HR-PCA and DHR-PCA only focus on making standard PCA robust to outliers but say nothing about whether it is possible to improve the robustness of non-negative PCA or sparse PCA. In this paper, we propose a more general framework as shown in Algorithm 1 for developing outlier-robust PCA-like algorithms. In Algorithm 1, the weighted covariance matrix acts as a robust covariance estimator (Rousseeuw, 1985; Rousseeuw & Driessen, 1998; Croux & Haesbroeck, 2000), and $\overline{V}_{\hat{t}}(\mathbf{X})$ is the *Robust Variance Estimator* which is defined as $\overline{V}_{\hat{t}}(\mathbf{X}) \triangleq \frac{1}{\hat{t}} \sum_{i=1}^{\hat{t}} \langle \mathbf{y}\mathbf{y}^\top, \mathbf{X} \rangle_{(i)}$, where $\mathbf{y} \in \mathcal{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$. Intuitively, the term $\langle \mathbf{y}\mathbf{y}^\top, \mathbf{X} \rangle$ imitates the magnitude of $\mathbf{y}$ after it is projected on the column subspace of $\mathbf{X}$, so this RVE measures the projection variance similar to the one in HR-PCA. As we show below, a PCA-like algorithm becomes outlier-robust if it is integrated into this general robustness framework. For example, DHR-PCA can be easily deduced from this framework by solving the standard PCA in Step 3.

3. Theoretical Guarantees

We now present the performance guarantees of Algorithm 1 with a (η, γ)-workable PCA-like algorithm. Typically, there are two ways to measure the performance of PCA-like algorithms (Xu et al., 2013; Vu et al., 2013). The first one, termed the *subspace distance* (S.D.), measures the distance between the subspace spanned by the estimated PCs and the subspace spanned by the true PCs. The second one,

Algorithm 1 Outlier-Robust PCA-like Algorithm

Input: Contaminated sample-set $\mathcal{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$, k, T, $\hat{t}$, μ.

Procedure:

1) Initialize: $s = 0$, Opt $= 0$; $\hat{\mathbf{y}}_i = \mathbf{y}_i$ and $\alpha_i = 1$ for $i = 1, \cdots, n$;

while $s \leq T$ **do**

2) Compute the weighted empirical covariance matrix $\hat{\mathbf{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} \alpha_i \hat{\mathbf{y}}_i \hat{\mathbf{y}}_i^\top$;

3) Solve the PCA-like problem 1 and denote the output by $\hat{\mathbf{X}}$;

4) If $\overline{V}_{\hat{t}}(\hat{\mathbf{X}}) >$ Opt, let Opt $= \overline{V}_{\hat{t}}(\hat{\mathbf{X}})$ and $\mathbf{X}^* = \hat{\mathbf{X}}$, where $\overline{V}_{\hat{t}}(\hat{\mathbf{X}}) \triangleq \frac{1}{\hat{t}}\sum_{i=1}^{\hat{t}} \langle \mathbf{y}\mathbf{y}^\top, \hat{\mathbf{X}} \rangle_{(i)}$;

5) Update $\alpha_i = (1 - \frac{\langle \mathbf{y}_i\mathbf{y}_i^\top, \hat{\mathbf{X}} \rangle}{\max_{\{i|\alpha_i \neq 0\}} \langle \mathbf{y}_i\mathbf{y}_i^\top, \hat{\mathbf{X}} \rangle})\alpha_i$;

6) $s = s + 1$;

end while

7) Perform SVD on $\mathbf{X}^*$ and denote the top k eigenvectors by $\mathbf{w}_1^*, \cdots, \mathbf{w}_k^*$;

8) return $\mathbf{w}_1^*, \cdots, \mathbf{w}_k^*$ and $\mathbf{X}^*$.

termed the *expressed variance* (E.V.), measures the portion of the signal $\mathbf{A}\mathbf{x}$ being expressed by the estimated principle components. Formally, we have:

Definition 2. *Let $\mathbf{M}_1, \mathbf{M}_2$ be two symmetric matrices and $\mathcal{M}_1, \mathcal{M}_1$ be their respective k-dimensional principal subspaces, then the subspace distance is S.D.* $\triangleq \sin\Theta(\mathcal{M}_1, \mathcal{M}_1)$.

Definition 3. *The expressed variance of $\mathbf{w}_1, \cdots, \mathbf{w}_k$ is defined as E.V.* $\triangleq \frac{\sum_{i=1}^{k} \mathbf{w}_i^\top \mathbf{A}\mathbf{A}^\top \mathbf{w}_i}{\sum_{i=1}^{k} \lambda_i(\mathbf{A}\mathbf{A}^\top)}$.

Notice that a smaller S.D. or a larger E.V. indicates a more desirable solution. Also, S.D. ≥ 0 and E.V. ≤ 1 with equality achieved when the vectors $\mathbf{w}_1, \cdots, \mathbf{w}_k$ span the same space as the true PCs. Thus, to provide performance guarantees of the proposed algorithms, we lower bound the expressed variance as well as upper bound the subspace distance for the output. This is different from (Xu et al., 2013) and (Feng et al., 2012) which only analyzed the expressed variance (of HR-PCA and DHR-PCA respectively).

To analyze the performance of Algorithm 1, the following "tail weight" function the first appeared in (Xu et al., 2013) is required.

Definition 4. *((Xu et al., 2013)) For any $\gamma \in [0, 1]$ and $\mathbf{v} \in \mathcal{S}_d$, let $\delta_\gamma \triangleq \min\{\delta \geq 0 | \bar{\nu}_{\mathbf{v}}([-\delta, \delta]) \geq \gamma\}$ and $\gamma_\nu^- = \bar{\nu}_{\mathbf{v}}((-\delta, \delta))$. Then the "tail weight" functions $\mathcal{V}_{\mathbf{v}}$ is defined as follows:*

$$\mathcal{V}_{\mathbf{v}}(\gamma) \triangleq \lim_{\epsilon \downarrow 0} \int_{-\delta_\gamma + \epsilon}^{\delta_\gamma - \epsilon} x^2 \bar{\nu}_{\mathbf{v}}(dx) + (\gamma - \gamma_\nu^-)\delta_r^2.$$

We define $\mathcal{V}^+(\gamma) \triangleq \sup_{\mathbf{v} \in \mathcal{S}_d} \mathcal{V}_{\mathbf{v}}(\gamma)$ and $\mathcal{V}^-(\gamma) \triangleq \inf_{\mathbf{v} \in \mathcal{S}_d} \mathcal{V}_{\mathbf{v}}(\gamma)$. In the following subsections, we assume that the feasible set $\mathcal{C}$ in (1) is a subset of $\mathcal{F}(k)$ – the convex hull of all the feasible projection matrices. This is not a restrictive condition. Indeed all the algorithms listed in Section 2.2 except large-scale sparse PCA meet this condition.

3.1. Upper Bound of Subspace Distance

We first bound the subspace distance for Algorithm 1. The following lemma relates the subspace distance with the Frobenius norm of $\mathbf{X}^* - \mathbf{\Pi}$ so that we only need to bound $\|\mathbf{X}^* - \mathbf{\Pi}\|_F$.

Lemma 1. *(Vu et al., 2013) If $\mathcal{M}$ is the principal d-dimensional subspace of $\mathbf{\Sigma}$ and $\mathcal{M}^*$ is the principal k-dimensional subspace of $\mathbf{X}^*$, then*

$$\sin\Theta(\mathcal{M}, \mathcal{M}^*) \leq \sqrt{2}\|\mathbf{X}^* - \mathbf{\Pi}\|_F.$$

In the following parts, we let $\delta_k(\mathbf{A}\mathbf{A}^\top) \triangleq \lambda_k(\mathbf{A})^2 - \lambda_{k+1}(\mathbf{A})^2$ and let

$$f(B) = \min\left\{2B\|\mathbf{A}\|_2^2 + c_1\tau, \gamma B\|\mathbf{A}\|_2^2 + c_2\gamma(d\|\mathbf{A}\|_2 + 1)\right\},$$

where $\tau = \max\{\frac{p}{n}, 1\}$ and c is a universal constant. Notice that $f(B)$ is upper bounded by $2B\|\mathbf{A}\|_2^2 + c_1$ when $p = O(n)$ and by $\gamma B\|\mathbf{A}\|_2^2 + c_2\gamma(d\|\mathbf{A}\|_2 + 1)$ when $p = \Omega(n)$ and $\gamma \ll p$ for some constants c_1 and c_2. Therefore, in the high dimensional case where $p \gg n$, when sparse PCA algorithms are applied, i.e., $\gamma \ll p$, $f(B)$ can still be small, compared with $\frac{p}{n}$.

We now provide our first main theorem which states that the output $\hat{\mathbf{X}}$ in Step 3 will be close to the true projection matrix after a certain number of iterations.

Theorem 1. *Suppose that $\rho < 0.5$ and $\log p \leq n$, then there exists a finite number $s \leq n$ such that the output $\mathbf{X}_s$ of the PCA-like algorithm in the s^{th} stage satisfies the following inequality with high probability,*

$$\|\mathbf{X}_s - \mathbf{\Pi}\|_F \leq R(\mu) + \sqrt{k} \min_{1 \geq \kappa > 2\rho} \sqrt{\frac{f(B_1) + \eta\beta B_0}{\delta_k(\mathbf{A}\mathbf{A}^\top)}}, \tag{2}$$

where

$$R(\mu) \triangleq \begin{cases} \frac{8(\gamma[\epsilon_0(\|\mathbf{A}\|_2^2+1)-\mu]_+ + \mu\beta)}{\delta_k(\mathbf{A}\mathbf{A}^\top)}, & \mu \neq 0 \\ \min\left\{\frac{8\epsilon_0\gamma(\|\mathbf{A}\|_2^2+1)}{\delta_k(\mathbf{A}\mathbf{A}^\top)}, 2\sqrt{\frac{\epsilon_1 k(\|\mathbf{A}\|_2^2+1)}{\delta_k(\mathbf{A}\mathbf{A}^\top)}}\right\}, & \mu = 0, \end{cases}$$

$\epsilon_0 = c_0\sqrt{\frac{\log p}{n}}$, $\epsilon_1 = c_1\sqrt{\frac{p}{n}}$, $B_0 = c_2(\|\mathbf{A}\|_2^2 + 1)$, $B_1 = \kappa + 1 - \mathcal{V}^-(1 - \frac{\rho}{\kappa(1-\rho)}) + \epsilon_0 + c_3\left(\frac{d\log^3 n}{n}\right)^{\frac{1}{4}}$, *and c_0, c_1, c_2, c_3 are universal constants.*

Remark. The upper bound of $\|\mathbf{X}_s - \mathbf{\Pi}\|_F$ involves three terms: 1) $R(\mu)$: $R(\mu)$ is related to the weight of the regularization term in (1). A positive μ can encourage sparse solutions. From the formulation of $R(\mu)$, we know that setting μ to $\epsilon_0(\|\mathbf{A}\|_2^2+1)$ when μ is non-zero leads to a tighter bound. 2) $f(B_1)$: B_1 involves ρ – the fraction of outliers, and decreases when ρ decreases. Clearly, $B_1 \to 0$ when ρ, $\frac{d\log^3 n}{n}$ and $\frac{\log p}{n}$ converge to zero. 3) $\eta\beta B_0$: This term contains η, i.e., the accuracy of the selected PCA-like algorithm. When the optimal solution of (1) can be achieved, this term becomes zero.

Theorem 1 tells us that a good solution $\mathbf{X}_s$ can be generated for some iteration s. However, such s is not specified. Thus, one can not take $\mathbf{X}_s$ as the output; instead, one can choose a solution that is close to $\mathbf{X}_s$ as the output. In Algorithm 1, the solution with the maximal RVE is selected as the final output $\mathbf{X}^*$. Other methods can also be applied in practical applications based on specific information. The following theorem provides the estimation error of $\mathbf{X}^*$.

Theorem 2. *Suppose that $\rho < 0.5$ and $\log p \le n$, the following holds with high probability,*

$$\|\mathbf{X}^* - \mathbf{\Pi}\|_F \le \sqrt{\frac{2\left[(dB_2 + kB_4)\lambda_1(\mathbf{A})^2 + kf(B_3)\right]}{\delta_k(\mathbf{A}\mathbf{A}^\top)}}, \tag{3}$$

where B_2 is the right hand side of (2),

$$B_3 = 2 - \mathcal{V}^-(\frac{\hat{t}}{t}) - \mathcal{V}^-(\frac{\hat{t}-\rho n}{t}) + c_0\left(\frac{d\log^3 n}{n}\right)^{\frac{1}{4}},$$

$B_4 = \min\{c_1\sqrt{\frac{p}{n}}, c_2\gamma\sqrt{\frac{\log p}{n}}\}$, *and c_0, c_1, c_2 are universal constants.*

Remark. This upper bound contains three terms: 1) B_2 is the upper bound of $\|\mathbf{X}_s - \mathbf{\Pi}\|_F$ as shown in Theorem 1. 2) B_3 involves ρ and parameter $\hat{t}$, which becomes small when ρ decreases and $\hat{t}$ approaches t. 3) B_4 converges to zero as $\frac{p}{n} \to 0$ or $\gamma\sqrt{\frac{\log p}{n}} \to 0$. To achieve consistency, one should ensure that $\frac{p}{n} \to 0$ for the standard PCA where $\gamma = p$, and $\frac{\gamma^2 \log p}{n} \to 0$ for sparse PCA where $\gamma \ll p$.

The following corollaries provide more interpretable bounds of the subspace distance for the standard PCA, FPS and sparse PCA discussed in Section 2.2.

Corollary 1. *Suppose that $\rho < 0.5$ and $\log p \le n$, then when the PCA-like algorithm is the standard PCA (Pearson, 1901), the following holds with high probability,*

$$\|\mathbf{X}^* - \mathbf{\Pi}\|_F \le \sqrt{\frac{4d(B_2 + B_3 + \epsilon)\lambda_1(\mathbf{A})^2 + cd\tau}{\lambda_d(\mathbf{A})^2}}, \tag{4}$$

where

$$B_2 = 2\sqrt{\frac{\epsilon d(\lambda_1(\mathbf{A})^2+1)}{\lambda_d(\mathbf{A})^2}} + \min_{1\ge\kappa>2\rho}\sqrt{\frac{2dB_1\lambda_1(\mathbf{A})^2 + cd\tau}{\lambda_d(\mathbf{A})^2}},$$

B_1 is defined in Theorem 1, B_3 is defined in Theorem 2, $\epsilon = c_1\sqrt{\frac{p}{n}}$, $\tau = \max\{\frac{p}{n}, 1\}$ and c, c_0, c_1 are universal constants.

The standard PCA imposes no constraint on the sparsity of its solution, so when the ambient dimension p grows faster than the sample number n, the bound in Corollary 1 will go to infinity. One way to encourage sparsity is to impose a "soft" constraint which upper bounds the l_1-norm of the solution, e.g., FPS.

Corollary 2. *Suppose that $\rho < 0.5$ and $\log p \le n$, then when the PCA-like algorithm is FPS (Vu et al., 2013), the following holds with high probability,*

$$\|\mathbf{X}^* - \mathbf{\Pi}\|_F \le \sqrt{\frac{4d(B_2 + B_3 + \epsilon_1)\lambda_1(\mathbf{A})^2 + cd\tau}{\lambda_d(\mathbf{A})^2}}, \tag{5}$$

where

$$B_2 = \frac{\epsilon_0(\lambda_1(\mathbf{A})^2+1)\beta}{\lambda_d(\mathbf{A})^2} + \min_{1\ge\kappa>2\rho}\sqrt{\frac{2dB_1\lambda_1(\mathbf{A})^2 + cd\tau}{\lambda_d(\mathbf{A})^2}},$$

B_1 is defined in Theorem 1, B_3 is defined in Theorem 2, $\epsilon_0 = c_0\sqrt{\frac{\log p}{n}}$, $\epsilon_1 = c_1\sqrt{\frac{p}{n}}$, $\tau = \max\{\frac{p}{n}, 1\}$ and c, c_0, c_1 are universal constants.

Notice that p cannot grow faster than $\frac{n\lambda_d(\mathbf{A})^2}{d}$ due to the existence of outliers, but the first term in B_2 in Corollary 2 involves $\frac{\log p}{n}$ instead of $\frac{p}{n}$, which is much smaller than that in Corollary 1. Thus, the soft constraint is helpful if the true solution is indeed sparse. When the selected PCA-like algorithm has a "hard" constraint on the sparsity, e.g., the ones proposed by (Papailiopoulos et al., 2013) and (Asteris et al., 2014), p can grow much faster than n.

Corollary 3. *Suppose that $\rho < 0.5$ and $\log p \le n$, then when $\gamma \ge \beta$ and the PCA-like algorithm is sparse PCA (Papailiopoulos et al., 2013) or non-negative sparse PCA (Asteris et al., 2014), the following holds with high probability,*

$$\|\mathbf{X}^* - \mathbf{\Pi}\|_F \le \sqrt{\frac{2\left[(dB_2 + \beta B_3 + \beta\epsilon_0)\lambda_1(\mathbf{A})^2 + c\beta(d\lambda_1(\mathbf{A})+1)\right]}{\delta_1(\mathbf{A}\mathbf{A}^\top)}}, \tag{6}$$

where

$$B_2 = \frac{\epsilon_0(\lambda_1(\mathbf{A})^2+1)\beta}{\delta_1(\mathbf{A}\mathbf{A}^\top)} + \sqrt{\gamma}\min_{1\ge\kappa>2\rho}\sqrt{\frac{B_1\lambda_1(\mathbf{A})^2 + c(d\lambda_1(\mathbf{A})+1) + \eta B_0}{\delta_1(\mathbf{A}\mathbf{A}^\top)}}.$$

B_0, B_1 are defined in Theorem 1, B_3 is defined in Theorem 2, $\epsilon_0 = c_0\sqrt{\frac{\log p}{n}}$, and c, c_0 are universal constants.

Recall that $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^\top + I_p$. The bound shown in Corollary 3 can be finite regardless of the magnitude of the existing outliers, e.g., when d, $\beta\sqrt{\frac{\log p}{n}}$, $(B_1 + B_3 + \eta)\gamma$, $\frac{\gamma}{\lambda_1(A)}$ and $\frac{\lambda_1(\mathbf{\Sigma})}{\lambda_1(\mathbf{\Sigma}) - \lambda_2(\mathbf{\Sigma})}$ are bounded from above.

3.2. Lower Bound of Expressed Variance

Xu et al. (2013) and Feng et al. (2012) provided lower bounds of E.V. when the standard PCA is selected in Algorithm 1. We now show that E.V. can be bounded from below when other PCA-like algorithm of form (1) (and workable) are used in Algorithm 1. Let $H^* \triangleq \langle \mathbf{A}\mathbf{A}^\top, \mathbf{X}^* \rangle$ and $\overline{H} \triangleq \langle \mathbf{A}\mathbf{A}^\top, \mathbf{\Pi} \rangle$, then we have the following theorem.

Theorem 3. *Suppose that $\rho < 0.5$. For any κ, there exists a constant c such that the following inequalities hold w.h.p,*

$$
\begin{aligned}
E.V \geq & \frac{(1-\eta)\mathcal{V}^-\left(\frac{\hat{t}}{t} - \frac{\rho}{1-\rho}\right)\mathcal{V}^-\left(1 - \frac{\rho}{\kappa(1-\rho)}\right)}{(1+\kappa)\mathcal{V}^+\left(\frac{\hat{t}}{t}\right)} \\
& - \frac{c}{\mathcal{V}^+(0.5)}\left[\left(\frac{k\min\{\tau, \varsigma\}}{\overline{H}}\right)^{\frac{1}{2}} + \left(\frac{d\log^3 n}{n}\right)^{\frac{1}{4}}\right] \\
& - \frac{2(1-\eta)\mu\beta\sqrt{k}}{\mathcal{V}^+(\frac{\hat{t}}{t})\overline{H}} - \max\{1 - \lambda_k(\mathbf{X}^*), \lambda_{k+1}(\mathbf{X}^*)\},
\end{aligned} \tag{7}
$$

where $\tau = \max\{\frac{p}{n}, 1\}$ and $\varsigma = \max\{\frac{\log p}{n}, 1\}$.

As discussed in Section 2.2, $\mathbf{X}^*$ has the form $\mathbf{X}^* = \mathbf{u}\mathbf{u}^\top$ for the standard PCA, non-negative PCA (Montanari & Richard, 2014), sparse PCA (Papailiopoulos et al., 2013) and non-negative sparse PCA (Asteris et al., 2014), which implies that the last term in (7) vanishes for these four algorithms when $k = 1$. But for FPS (Vu et al., 2013), this term may not be zero. The following lemma shows that it can converge to zero under certain circumstances.

Lemma 2. *Suppose that $\mathcal{S}$ is a sequence of matrices such that for any $\mathbf{S}_n \in \mathcal{S}$, $\mathbf{S}_n \in \mathbb{S}_+^{p\times p}$ and $\lambda_d(\mathbf{S}_n) - \lambda_{d+1}(\mathbf{S}_n) \geq \delta > 0$. Let*

$$
\mathbf{X}_n \triangleq \arg\max_{\mathbf{X}\in\mathcal{F}(d)} \langle \mathbf{S}_n, \mathbf{X} \rangle - \mu_n \|\mathbf{X}\|_1,
$$

then if $\mu_n \to 0$ as $n \to +\infty$ and $pd^{3/2} = o(\frac{1}{\mu_n})$, we have $\lambda_d(\mathbf{X}_n) \to 1$ and $\lambda_{d+1}(\mathbf{X}_n) \to 0$ as $n \uparrow +\infty$.

The following result shows the asymptotic bound of the expressed variance in which we assume that the last term in (7) converges to zero as n goes to infinity. This condition holds for all the algorithms mentioned above.

Theorem 4. *(Asymptotic Bound): Consider a sequence of $\{\mathcal{Y}_i, d_i, n_i, p_i, \mu_i, \beta_i, \gamma_i\}$, where the asymptotic scaling satisfies*

$$
n_i \uparrow +\infty,\ \lim_{i\uparrow+\infty}\frac{\log p_i}{n_i} \leq +\infty,\ \lim_{i\uparrow+\infty}\frac{\min\{p_i/n_i, \gamma_i\}}{\sum_{j=1}^k \lambda_j(\mathbf{A}_i)^2} \downarrow 0,
$$

$$
\frac{n_i}{d_i \log^3 n_i} \uparrow +\infty,\ \frac{d_i}{\sum_{j=1}^k \lambda_j(\mathbf{A}_i)^2} \downarrow 0,\ \mu_i\beta_i \downarrow 0,
$$

Let $\rho^* = \limsup \rho_i \leq 0.5$ and suppose $\hat{t} > 0.5n$, then if $\lambda_k(\mathbf{X}^*) \to 1$ and $\lambda_{k+1}(\mathbf{X}^*) \to 0$ as $n_i \uparrow +\infty$, the following holds in probability when $i \uparrow +\infty$,

$$
\liminf_i E.V \geq (1-\eta)\max_\kappa \frac{\mathcal{V}^-\left(1 - \frac{\rho^*}{(1-\rho^*)\kappa}\right)\mathcal{V}^-\left(\frac{\hat{t}}{t} - \frac{\rho^*}{1-\rho^*}\right)}{(1+\kappa)\mathcal{V}^+\left(\frac{\hat{t}}{t}\right)}.
$$

Furthermore, if $\bar{\mu}_{\mathbf{v}}(\{0\}) = 0$ for all $\mathbf{v} \in \mathcal{S}_d$, then the breakdown point is $\rho^* = 0.5$.

Corollary 4. *Under the settings of the above theorem, the following holds in probability for some constant C when $i \uparrow +\infty$,*

$$
\liminf_i E.V \geq (1-\eta)\left[\frac{\mathcal{V}^-(\frac{\hat{t}}{t}) - C\sqrt{\theta\rho^*\log(1/2\rho^*)}}{\mathcal{V}^+(\frac{\hat{t}}{t})}\right].
$$

3.3. Complexity

Recall that Algorithm 1 is an iterative algorithm that solves a PCA-like algorithm in each iteration. Theoretically, the number of iterations required to generate a good solution is bounded by n. But in practice, one can stop the algorithm at any time as long as the output of the robust variance estimator is good enough. We will show in the experiments that 5-10 iterations are sufficient to achieve a good solution. Since the time and space complexity of Algorithm 1 mainly depends on performing the selected PCA-like algorithm, this means the computational cost of Algorithm 1 is about 5-10 times higher than the non-robust PCA-like algorithm – robustness is not a free lunch, but you don't pay much.

4. Experimental Results

In this section, we show that our framework indeed makes PCA-like algorithms more robust to outliers. We refer to the selected PCA-like algorithm in Step 3 in Algorithm 1 as $\mathcal{A}$ and consider four algorithms induced from our framework: 1) OR-PCA: $\mathcal{A}$ is the standard PCA. OR-PCA has been extensively studied in (Xu et al., 2013). 2) OR-SPCA: $\mathcal{A}$ is FPS (Vu et al., 2013) to encourage sparse solutions. 3) Nonnegative OR-SPCA: $\mathcal{A}$ is non-negative sparse PCA (Asteris et al., 2014). 4) Large-scale OR-SPCA: $\mathcal{A}$ is the algorithm proposed by (Zhang & El Ghaoui, 2011) which

is able to handle high dimensional data. Although this algorithm has no performance guarantees, it does work well in the experiments.

Firstly, we illustrate the performance of OR-PCA and OR-SPCA via numerical results on synthetic and real data. For synthetic data, we generate matrix $\mathbf{A}$ via the following three steps: 1) randomly generate sparse orthogonal matrices $\mathbf{U} \in \mathbb{R}^{p\times d}$ and $\mathbf{V} \in \mathbb{R}^{d\times d}$ such that $\|\mathbf{U}\|_{2,0} = \beta$ where $\|\mathbf{U}\|_{2,0}$ is the number of non-zero rows in $\mathbf{U}$; 2) generate a diagonal matrix $\mathbf{S}$ whose diagonal entries are drawn from (a) the uniform distribution over $[1, 2]$ or (b) the chi-square density $\frac{x^{-0.5}e^{-0.5x}}{\sqrt{2}\Gamma(0.5)}$ where x is chosen from 0.05 to $0.05d$ using step-size 0.05; 3) finally, let $\mathbf{A} = \mathbf{USV}^\top$. The t authentic samples $\mathbf{z}_i$ are generated by the function $\mathbf{z}_i = \mathbf{Ax}_i + \mathbf{n}_i$ where $\mathbf{x}_i \sim \mathcal{N}(0, I_d)$, $\mathbf{n}_i \sim \mathcal{N}(0, \sigma^2 I_p)$. A ρ fraction outliers $\mathbf{o}_i$ are generated with a uniform distribution over $[-c, c]^p$ where c is a constant.

We make a comparison between OR-PCA, OR-SPCA, FPS and ROB-SPCA. ROB-SPCA is developed based on (Hubert et al., 2014), which uses ROBPCA (Hubert et al., 2005) to estimate the robust sample covariance and then applies FPS to compute the principal components. The performance is evaluated by the "expressed variance" and "sparsity". The sparsity is defined by

$$\text{Sparsity} \triangleq |(i, j) : |X_{ij}| > 0.001|/p^2,$$

where $\mathbf{X}$ is the projection matrix generated by each algorithm.

In the first experiment, we compare the performance of each algorithm when ρ varies while the other parameters are fixed. The parameters for generating test data are set as follows: $d = 10$, $\sigma = 0.05$, $\beta = 0.3p$. Parameter T and $\hat{t}$ for OR-PCA and OR-SPCA are set to 10 and ρn, respectively. Parameter μ for FPS and OR-SPCA is $0.2\sqrt{\frac{\log p}{n}}$. For each parameter setup, we report the average results of 10 tests. Figures 1 and 2 show the performance of these four algorithms. Clearly, FPS easily breaks down, even when there exists only a small fraction of outliers. ROB-SPCA breaks down when ρ is larger than 0.25. Actually, most of robust PCA algorithms based on ROBPCA do not work well when the fraction of outliers exceeds 0.25 (Xu et al., 2013). One can also observe that OR-PCA and OR-SPCA are much more robust than the other two algorithms, and OR-SPCA can generate more sparse solutions than OR-PCA without significant decrease in the expressed variance, which implies that our framework has the capability of converting a non-robust SPCA algorithm, e.g., FPS, into a robust one.

In the second experiment, we investigate the number of the iterations required in Algorithm 1 to achieve good performance. We take OR-SPCA as an example. Figure 3 shows

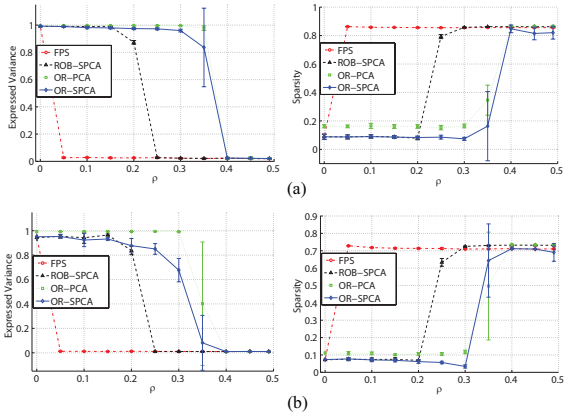

Figure 1. The performance of OR-PCA, OR-SPCA, ROB-SPCA and FPS under (a) $p = 500, n = 300, c = 5$ and (b) $p = 1000, n = 300, c = 5$. The singular values of $\mathbf{A}$ are uniformly drawn from $[1, 2]$.

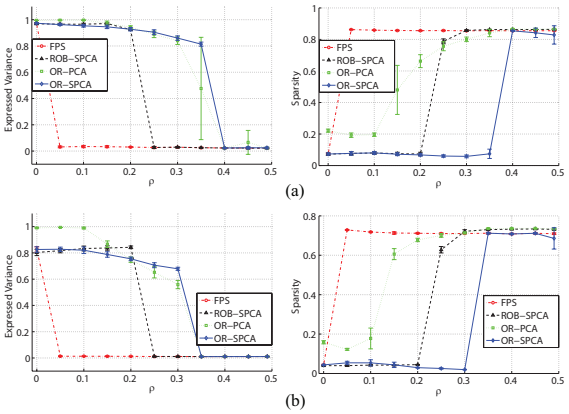

Figure 2. The performance of OR-PCA, OR-SPCA, ROB-SPCA and FPS under (a) $p = 500, n = 300, c = 5$ and (b) $p = 1000, n = 300, c = 5$. The singular values of $\mathbf{A}$ are uniformly drawn from the chi-square density.

the effect of the number of iterations on the expressed variance and sparsity for OR-SPCA under three cases that $p = 600$, $p = 800$ and $p = 1000$, from which we observe that only 5 iterations are required for OR-SPCA to generate acceptable results in all three cases. Empirically, we

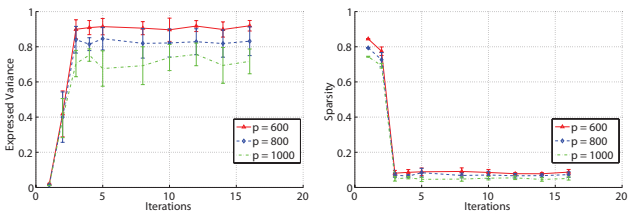

Figure 3. The effect of the number of iterations on the expressed variance and sparsity. n, ρ and c are fixed: $n = 300$, $\rho = 0.1$, $c = 5$.

observe that 5-10 iterations are enough for Algorithm 1 to compute good results in practical applications. Hence in the following experiments on real data, parameter T is set to 10.

In the third experiment, we show the performance of OR-SPCA, OR-PCA and FPS on a real dataset of 600 samples in which 75% of samples are drawn from MNIST (LeCun et al., 1995) and 25% of samples are drawn from the CBCL face image dataset (Sung, 1996). We take the digit images as the authentic samples and the face images as the outliers. Each image in this dataset is converted into a vector with dimension 784. Figure 4 shows the leading ten principal components extracted by FPS, OR-PCA and OR-SPCA. It can be observed that OR-SPCA can generate more interpretable results than OR-PCA, i.e., each PC corresponds to some strokes. Notice that the principal components extracted by OR-SPCA are more reliable than FPS. For example, the third principal component extracted by FPS clearly mixes digits with faces, which is obviously unreliable.

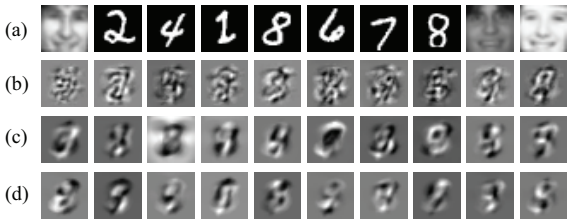

Figure 4. We plot the leading ten PCs extracted by OR-PCA, FPS and OR-SPCA. (a) shows a couple of sample images. (b), (c) and (d) show the results of OR-PCA, FPS and OR-SPCA, respectively.

Secondly, we evaluate the performance of the non-negative OR-SPCA on the real world dataset constructing by mixing 2429 images in the CBCL face image dataset with 125 digit images randomly drawn from the MNIST dataset. We take the face images as the authentic samples and the digit images as the outliers. Each image in this dataset is converted into a vector with dimension 361. We compare non-negative OR-SPCA with non-negative SPCA. Figure 5 shows the sample images and the five leading PCs computed by non-negative SPCA and non-negative OR-SPCA. Clearly, non-negative SPCA fails in the face of these "digit" outliers, while non-negative OR-SPCA can still extract good principal components that are close to the ones generated by applying non-negative SPCA on the clean data, i.e., 2429 face images only.

Finally, we use the NYTimes news article dataset from the UCI Machine Learning Repository (Frank & Asuncion, 2010), which contains 300000 articles and a dictionary of 102660 unique words, to illustrate the performance of Algorithm 1 on large-scale data. 3000 random vectors whose

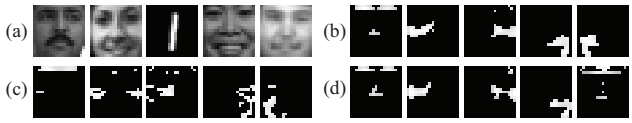

Figure 5. We plot (a) five samples in the dataset, (b) the five leading PCs extracted by non-negative SPCA on the clean data (2429 face images), and the five leading PCs extracted by (c) non-negative SPCA and (d) non-negative OR-SPCA on the dirty data (2429 face images plus 125 outliers).

Table 1. The words associated with the leading two sparse principal components extracted by large-scale SPCA and large-scale OR-SPCA. The ground truth is obtained by performing large-scale sparse PCA on the clean data.

Ground-truth		LS-SPCA		LS-OR-SPCA	
1st PC	2st PC	1st PC	2st PC	1st PC	2st PC
million	point	site	fire	percent	team
percent	play	summer	scientist	company	player
business	team	contract	oil	million	season
company	season	system	prices	market	game
market	game	person	district	money	play

entries are randomly drawn from the uniform distribution with support $[0, 100]$ are added into the NYTimes dataset, which are taken as outliers. We choose large-scale SPCA (LS-SPCA) proposed by (Zhang & El Ghaoui, 2011) as $\mathcal{A}$ and compare the corresponding large-scale OR-SPCA (LS-OR-SPCA) with it. Table 1 provides the leading two sparse PCs in which the first two columns shows the two leading PCs extracted by LS-SPCA on the dataset without outliers, and the other four columns presents the leading PCs extracted by LS-SPCA and LS-OR-SPCA on the dataset with outliers. Clearly, the results of LS-SPCA are meaningless when outliers exist, whereas LS-OR-SPCA can generate quite similar results to the ground-truth where the first PC is about business and the second PC is about sports.

5. Conclusion

In this paper, we proposed a unified framework for making PCA-like algorithms robust to outliers. We provided theoretical performance analysis of the proposed framework using both the subspace distance and the expressed variance metrics. To the best of our knowledge, this is the first attempt to make a wide range of PCA-like algorithms provably robust to any constant fraction of arbitrarily corrupted samples. As an immediate result, our framework leads to robust sparse PCA and robust non-negative sparse PCA with theoretic guarantees – the first of its kind to the best of our knowledge. The experiments show that the outlier-robust PCA-like algorithms derived from our framework outperforms their non-robust version and other alternatives including HR-PCA and ROB-SPCA.

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

Asteris, M., Papailiopoulos, D. S., and Dimakis, A. G. Nonnegative sparse PCA with provable guarantees. In *ICML*, 2014.

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.

Croux, C. and Haesbroeck, G. Principal component analysis based on robust estimators of the covariance or correlation matrix: Influence functions and efficiencies. *BIOMETRIKA*, 87:603–618, 2000.

Croux, C., Filzmoser, P., and Fritz, H. Robust sparse principal component analysis. *Technometrics*, 55(2):202–214, 2013.

Dasgupta, S. Subspace detection: A robust statistics formulation. *In Proceedings of the Sixteenth Annual Conference on Learning Theory*, 2003.

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.

Devlin, S. J., Gnanadesikan, R., and Kettenring, J. R. Robust estimation of dispersion matrices and principal components. *Journal of the American Statistical Association*, 76(374):354–362, 1981.

Feng, J., Xu, H., and Yan, S. Robust PCA in high-dimension: A deterministic approach. In *ICML*, 2012.

Frank, A. and Asuncion, A. UCI machine learning repository. 2010.

Hubert, M., Rousseeuw, P. J., and Branden, K. ROBPCA: A new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005.

Hubert, M., Reynkens, T., and Schmitt, E. Sparse PCA for high-dimensional data with outliers. *Technical Report*, 2014.

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.

Jolliffe, I. T., Trendafilov, N. T., , and Uddin, M. A modified principal component technique based on the Lasso. In *JCGS*, pp. 531–547, 2003.

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.

la Torre, F. De and Black, M. J. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1/2/3):117–142, 2003.

LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., Drucker, H., Guyon, I., Müller, U., Säckinger, E., Simard, P., and Vapnik, V. Comparison of learning algorithms for handwritten digit recognition. In *International Conference on Artificial Neural Networks*, pp. 53–60, 1995.

Moghaddam, B., Weiss, Y., and Avidan, S. Spectral bounds for sparse PCA: Exact and greedy algorithms. In *NIPS*, 2005.

Montanari, A. and Richard, E. Non-negative principal component analysis: Message passing algorithms and sharp asymptotics. In *arXiv:1406.4775*, 2014.

Papailiopoulos, D. S., Dimakis, A. G., and Korokythakis, S. Sparse PCA through low-rank approximations. In *ICML*, 2013.

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

Rousseeuw, P. J. Multivariate estimation with high breakdown point. *Mathematical statistics and applications*, 8, 1985.

Rousseeuw, P. J. and Driessen, K. V. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223, 1998.

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

Sung, K. Learning and example selection for object and pattern recognition. *PhD thesis, MIT*, 1996.

Thiao, M., Dinh, T. P., and Thi, H. A. A DC programming approach for sparse eigenvalue problem. In *ICML*, 2010.

Vu, V. Q. and Lei, Jing. Minimax bounds for sparse PCA with noisy high-dimensional data. *The Annals of Statistics*, 41(6):2703–3110, 2013.

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, L. and Cheng, H. Robust sparse PCA via weighted elastic net. In *Pattern Recognition*, pp. 88–95. Springer, 2012.

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

Xu, L. and Yuille, A. L. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, 6(1): 131–143, 1995.

Yang, T. N. and Wang, S. D. Robust algorithms for principal component analysis. *Pattern Recognition Letters*, 20 (9):927–933, 1999.

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

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