SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2014, Vol. 57 092115:1–092115:14
doi: 10.1007/s11432-013-4970-y

Robust sparse principal component analysis

ZHAO Qian[1], MENG DeYu[1]* & XU ZongBen[1,2]

[1]*Institute for Information and System Sciences, School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an* 710049*, China;*
[2]*Ministry of Education Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an* 710049*, China*

Received April 17, 2013; accepted June 9, 2013; published online June 16, 2014

Abstract The model for improving the robustness of sparse principal component analysis (PCA) is proposed in this paper. Instead of the l_2-norm variance utilized in the conventional sparse PCA model, the proposed model maximizes the l_1-norm variance, which is less sensitive to noise and outlier. To ensure sparsity, l_p-norm ($0 \leqslant p \leqslant 1$) constraint, which is more general and effective than l_1-norm, is considered. A simple yet efficient algorithm is developed against the proposed model. The complexity of the algorithm approximately linearly increases with both of the size and the dimensionality of the given data, which is comparable to or better than the current sparse PCA methods. The proposed algorithm is also proved to converge to a reasonable local optimum of the model. The efficiency and robustness of the algorithm is verified by a series of experiments on both synthetic and digit number image data.

Keywords noise, outlier, principal component analysis, robustness, sparsity

Citation Zhao Q, Meng D Y, Xu Z B. Robust sparse principal component analysis. Sci China Inf Sci, 2014, 57: 092115(14), doi: 10.1007/s11432-013-4970-y

1 Introduction

Principal component analysis (PCA) [1] is one of the most classical and popular methods for data analysis. Briefly speaking, PCA aims to seek a series of directions (principal components, or PCs), along which the variance of the data can be maximally captured. By projected onto the PCs, the dimensionality of the data can be reduced and simultaneously the intrinsic structure of the original data can be captured.

Although the traditional PCA has a lot of advantages, it suffers from the defect that each obtained PC is in general a linear combination of all the variables of original data, i.e. the PC loadings are generally all non-zeroes without any sparsity. However, sparsity always plays a central role in real applications. For example, in biology, each variable of the data corresponds to a certain gene, and thus the sparsity of PCs can facilitate the understanding of the relation between the whole gene microarraies and certain genes. Besides, in financial analysis, the sparsity of PCs implies fewer assets in a portfolio, which can be used to reduce the trading costs.

Consequently, sparse PCA has become a hot issue in data analysis in the last decade, and a variety of methods have been developed. Jolliffe [2] proposed a method for sparse PCA by rotating the PCs

*Corresponding author (email: dymeng@mail.xjtu.edu.cn)

obtained by the traditional PCA, to make the PCs interpretable. Cadima et al. [3] gave an approach which sets the PC loadings within a small absolute value to zeros, leading to sparse PCs. Jolliffe et al. presented SCoTLASS method, which is based on the Lasso [4], to find sparse PCs by enforcing l_1-norm constraints to PC loadings [5]. Zou et al. [6] proposed SPCA algorithm based on elastic-net regression, achieving better results than that of the SCoTLASS. D'Aspremont et al. [7] constructed DSPCA algorithm by relaxing the sparse PCA model to a semidefinite programming problem. Shen et al. [8] developed the approach called sPCA-rSVD (including sPCA-rSVD$_{l_0}$, sPCA-rSVD$_{l_1}$, sPCA-rSVD$_{\mathrm{SCAD}}$), computing sparse PCs by low-rank matrix factorization. Sigg et al. [9] designed EMPCA based on probabilistic PCA model to solve sparse PCA and nonnegative sparse PCA. Journée et al. [10] derived GPower algorithm (including GPower$_{l_0}$, GPower$_{l_1}$, GPower$_{l_0,m}$, GPower$_{l_1,m}$), which formulates sparse PCA as a non-concave function maximization, and solves it by iterative thresholding method. Sriperumbudur et al. [11] developed DCPCA algorithm, which transforms sparse PCA into DC-programming problem. Lu et al. [12] presented an augmented Lagrangian method (ALSPCA), solving sparse PCA by non-smooth optimization technique. Besides, greedy methods for sparse PCA were proposed by Moghaddam et al. (GSPCA [13]) and d'Aspremont et al. (PathSPCA [14]).

All the methods aforementioned involve the maximization of the l_2-norm variance of the input data along the PC direction under sparsity constraints. However, l_2-norm usually leads sparse PCA to lose effectiveness on noise or outlier cases (details will be discussed in the next section). Therefore, constructing the sparse PCA method inhering robustness has become an important issue in current research [15], and is also the focus of this paper. We have constructed a robust sparse PCA model by substituting l_1-norm variance for the l_2-norm variance under l_1 constraint [16]. In this study we further investigate such robust sparse PCA model under more general l_p constraint cases. We also propose a simple yet efficient algorithm to solve the model so constructed. The complexity of the proposed algorithm increases approximately linearly with both of the size and the dimensionality of the data, which is comparable or even better than those of the current sparse PCA methods. The robustness and efficiency of our algorithm is further verified by experiments.

The paper is organized as follows: the robustness problem of the current sparse PCA research is discussed in Section 2. The robust sparse PCA model and the algorithm are also proposed in this section. A series of experimental results are listed in Section 3. A conclusion of the paper and outlook for future research are finally given in Section 4.

2 Robust sparse principal component analysis

2.1 Model formulation

Denote data matrix by $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$, where d and n are the dimensionality and the size of the data respectively. Without loss of generality, we assume that the data have been centralized, i.e. $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i = 0$. The traditional PCA aims to seek an m ($m < d$) dimensional linear subspace, along which the variance of the data is maximally captured. This can be done via the following optimization:

$$\boldsymbol{w}_i^* = \arg\max_{\boldsymbol{w}_i\in\mathbb{R}^d} \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{w}_i = \arg\max_{\boldsymbol{w}_i\in\mathbb{R}^d} \left\|\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{X}\right\|_2^2, \text{ s.t. } \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i = 1,\ \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_j = 0,\ j < i,\ i = 1, \ldots, m, \tag{1}$$

where $\boldsymbol{\Sigma} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \in \mathbb{R}^{d\times d}$ is the data covariance matrix, and $\|\cdot\|_2$ denotes the l_2-norm of vectors. Through solving (1), the basis of the required m dimensional subspace, $\boldsymbol{W}^* = \{\boldsymbol{w}_i^*\}_{i=1}^m$, can then be obtained.

The sparse PCA model is formulated by enforcing sparsity constraint to the traditional PCA model, as expressed in the following:

$$\boldsymbol{w}_i^* = \arg\max_{\boldsymbol{w}_i\in\mathbb{R}^d} \left\|\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{X}\right\|_2^2, \text{ s.t. } \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i = 1,\ \|\boldsymbol{w}_i\|_p^p \leqslant t,\ \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_j = 0,\ j < i,\ i = 1, \ldots, m, \tag{2}$$

where $\|\boldsymbol{w}\|_p$ ($0 \leqslant p \leqslant 1$) denotes the l_p-norm of vector $\boldsymbol{w}$. Specifically, the l_p-norm of a d dimensional

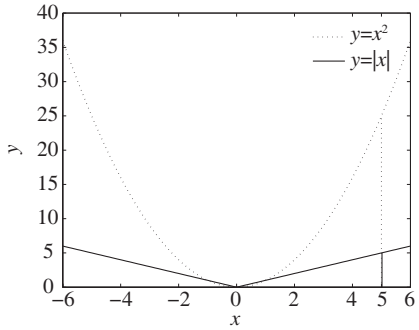

Figure 1 The comparison of the l_2-norm variance and the l_1-norm variance. $\|x\|_2^2$ tends to be much larger than $\|x\|_1$ as the absolute value of x increasing. For example, when $x = 5$, $\|x\|_2^2$ is 4 times larger than $\|x\|_1$.

vector $\boldsymbol{w} = (w^1, w^2, \ldots, w^d)^{\mathrm{T}}$ is defined by

$$\|\boldsymbol{w}\|_p = \begin{cases} \left(\sum\limits_{i=1}^{d} |\boldsymbol{w}^i|^p\right)^{1/p}, & 0 < p \leqslant 1, \\ \mathrm{Card}(\boldsymbol{w}), & p = 0, \end{cases}$$

where $\mathrm{Card}(\boldsymbol{w})$ denotes the number of nonzero elements in vector $\boldsymbol{w}$. Note that the sparse PCA model (2) differs from the traditional PCA model (1) on an additional constraint $\|\boldsymbol{w}_i\|_p^p \leqslant t$ (common choice for p is $p = 0$ or $p = 1$ in previous sparse PCA studies), ensuring the sparsity of the obtained PCs.

The l_2-norm variance $\left\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}\right\|_2^2$, which is employed as the objective in (2), intrinsically causes robustness problem of the model, i.e. the sparse PCA methods constructed on this model are generally sensitive to noises and outliers. There have been various researchers pointing out such robustness issue resulting from the l_2-norm variance [16–22]. A common strategy to alleviate this issue is to substitute the l_1-norm variance $\left\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}\right\|_1$ [17, 18] for the l_2-norm variance. The underlying principle is graphically depicted in Figure 1, which shows two curves of $f(x) = \|x\|_2^2$ and $f(x) = \|x\|_1$ respectively. It is easy to see that when the absolute value of x is large, $\|x\|_2^2$ tends to be much larger than $\|x\|_1$. For the data with heavy noises or outliers, l_2-norm variance always leads to magnified negative influence of these unexpected data and thus invalidate the related methods, while such negative effects tend to be ameliorated by adopting l_1-norm variance as the objective. Based on the analysis, we propose the following robust sparse PCA model:

$$\boldsymbol{w}_i^* = \underset{\boldsymbol{w}_i \in \mathbb{R}^d}{\arg\max} \ \left\|\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{X}\right\|_1, \ \text{s.t. } \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i = 1, \ \|\boldsymbol{w}_i\|_p^p \leqslant t, \ \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_j = 0, \ j < i, \ i = 1, \ldots, m, \tag{3}$$

where $\|\boldsymbol{w}_i\|_p^p \leqslant t$ is the sparsity constraint and $p \in [0, 1]$. This model has been investigated at $p = 1$ case in [16] and the model considered here can be viewed as a more general framework for robust sparse PCA. In the following, the algorithm for this general model are discussed.

Note that the above model is difficult to be solved by the classical optimization techniques due to the non-smooth l_1 and l_p norm included in the objective and the constraint of (3). We therefore design an specific algorithm to solve it in the next subsection.

2.2 The robust sparse PCA algorithm

We now present the algorithm for solving the robust sparse PCA model (3).

Instead of solving the entire model, the greedy strategy is designed to attain the approximate solution of (3) by sequentially solving the following single principal component problem:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w} \in \mathbb{R}^d}{\arg\max} \ \left\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}\right\|_1, \ \text{s.t. } \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} = 1, \ \|\boldsymbol{w}\|_p^p \leqslant t. \tag{4}$$

To solve this problem, two techniques, named AUGMENT and SPARSE, are employed. Specifically, the AUGMENT function generates a new unit vector in order to increase the value of the current objective

$\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}\right\|_1$. The SPARSE function produces the nearest vector, which satisfies the sparse constraint, to the vector obtained by the AUGMENT function. By executing the AUGMENT and the SPARSE functions alternately, the proposed algorithm produces a reasonable solution to the robust sparse PCA model. The algorithm is proposed as follows.

Algorithm 1 Robust sparse PCA algorithm

Input: Data matrix $\boldsymbol{X}=[\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n]\in\mathbb{R}^{d\times n}$, sparsity k, the number of sparse PCs $m>1$.

Step 1: Initialize $\boldsymbol{w}_0=\overrightarrow{0}$, where $\overrightarrow{0}$ is an all-zero vector.

Step 2: **For** $j=1,\ldots,m$:

Step 2.1 (Orthogonal projection) $\boldsymbol{X}=\{\boldsymbol{x}_i=\boldsymbol{x}_i-\boldsymbol{w}_{j-1}(\boldsymbol{w}_{j-1}^{\mathrm{T}}\boldsymbol{x}_i)\}_{i=1}^n$.

%This step projects the data set to the subspace orthogonal to $\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_{j-1}$, making the constraints $\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{w}_s=0, s<j$ approximately satisfied for $\boldsymbol{w}_j$.

Step 2.2 Initialize unit vector $\boldsymbol{w}(0)$, set $t=0$.

Step 2.3 (Optimizing the objective function) Generating unit vector by

$\boldsymbol{v}(t+1)=\mathrm{AUGMENT}(\boldsymbol{w}(t),\boldsymbol{X})$.

%This step increases the value of the objective $\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}\|_1$, i.e. $\|\boldsymbol{v}(t+1)^{\mathrm{T}}\boldsymbol{X}\|_1\geqslant\|\boldsymbol{w}(t)^{\mathrm{T}}\boldsymbol{X}\|_1$.

Step 2.4 (Sparse approximation) Computing the sparse vector

$\boldsymbol{u}(t+1)=\mathrm{SPARSE}(\boldsymbol{v}(t+1),k)$ which is the nearest one to $\boldsymbol{v}(t+1)$.

%This step makes the sparsity constraint $\|\boldsymbol{w}\|_p^p\leqslant t$ satisfied.

Step 2.5 (Normalization) $\boldsymbol{w}(t+1)=\frac{\boldsymbol{u}(t+1)}{\|\boldsymbol{u}(t+1)\|_2}$.

%This step makes the constraint $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}=1$ satisfied.

Step 2.6 Check the stopping criterion:

(1) If $\boldsymbol{w}(t)\neq\boldsymbol{w}(t+1)$, set $t=t+1$ and go to step 2.3; else, check (2);

(2) if there exists i, such that $\boldsymbol{w}^{\mathrm{T}}(t+1)\boldsymbol{x}_i=0$ and $\left|\mathrm{sign}(\boldsymbol{w}^{\mathrm{T}}(t+1))\right|\left|\mathrm{sign}(\boldsymbol{x}_i)\right|\neq 0$,

then $\boldsymbol{w}(t+1)=\frac{\boldsymbol{w}(t)+\triangle\boldsymbol{w}}{\|\boldsymbol{w}(t+1)+\triangle\boldsymbol{w}\|_2}$, where $\triangle\boldsymbol{w}$ is a random vector with small amount,

set $t=t+1$ and go to step 2.3; else, set $\boldsymbol{w}_j=\boldsymbol{w}(t)$, go to step 2.

End For

Step 3: Output k-sparse PCs $\{\boldsymbol{w}_j\}_{j=1}^m$.

We discuss the AUGMENT and the SPARSE functions, which are the key to the above algorithm, in the following.

2.3 The AUGMENT function

The AUGMENT function is of the following form:

$$\mathrm{AUGMENT}(\boldsymbol{w},\boldsymbol{X})=\frac{\sum_{i=1}^n p_i\boldsymbol{x}_i}{\left\|\sum_{i=1}^n p_i\boldsymbol{x}_i\right\|_2},\quad\text{where}\quad p_i=\begin{cases}1, & \text{if}\ \ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i\geqslant 0,\\ -1, & \text{if}\ \ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i<0.\end{cases}\tag{5}$$

The AUGMENT function, which increases the value of the l_1-norm variance $\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}\right\|_1$ after update, was first proposed by Kwak [17]. The construction of the AUGMENT function is based on the following theorem.

Theorem 1. [17] For any matrix $\boldsymbol{X}\in\mathbb{R}^{d\times n}$ and unit vector $\boldsymbol{w}\in\mathbb{R}^d$, denote $v=\mathrm{AUGMENT}(\boldsymbol{w},\boldsymbol{X})$, where the AUGMENT function is defined by (5). Then $\left\|\boldsymbol{X}^{\mathrm{T}}v\right\|_1\geqslant\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}\right\|_1$.

Proof. See [17].

Based on the above theorem, we can obtain the updated vector at which the objective function is increased by step 2.3 of the proposed algorithm.

2.4 The SPARSE function

The SPARSE function seeks the nearest vector to $\boldsymbol{v}$ under the sparsity constraint, i.e. solves the following problem:

$$\min_{\boldsymbol{u}}\ \frac{1}{2}\left\|\boldsymbol{u}-\boldsymbol{v}\right\|_2^2,\ \text{s.t.}\ \|\boldsymbol{u}\|_p^p\leqslant t,\tag{6}$$

where p can be any value in $[0, 1]$.

We first discuss the case where $0 < p < 1$. Denote $\boldsymbol{u} = (u_1, u_2, \ldots, u_d)^{\mathrm{T}} \in \mathbb{R}^d$, $\boldsymbol{v} = (v_1, v_2, \ldots, v_d)^{\mathrm{T}} \in \mathbb{R}^d$, and denote by θ_k the kth largest element of the absolute values of vector $\boldsymbol{v}$. Then we have the following theorem.

Theorem 2. Let $0 < p < 1$ and define $\lambda^* = \frac{[2(1-p)]^{1-p}}{(2-p)^{2-p}}\theta_k^{2-p}$. Then given the sparsity k, the optimal solutions of the optimization (6) are

$$u_i = \begin{cases} \mathrm{sign}(v_i)\beta_i^*, & |v_i| \geqslant \theta_k, \\ 0, & \text{else}, \end{cases} \tag{7}$$

where $\beta_i^* \in (\frac{2(1-p)}{2-p}, |v_i|]$ solves

$$\beta_i + \lambda^* p \beta_i^{p-1} = |v_i|, \quad \beta_i > 0 \tag{8}$$

and can be computed from the iteration

$$\beta_i^{k+1} = |v_i| - \lambda^* p (\beta_i^k)^{p-1} \tag{9}$$

with the initial condition $\beta_i^0 \in [\frac{2(1-p)}{2-p}, |v_i|]$.

Proof. The Lagrangian function of (6) is given by

$$L(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{v}\|_2^2 + \lambda(\|\boldsymbol{u}\|_p^p - t).$$

Minimizing this function results in a series of single variable optimization problems

$$\min_{u_i} \frac{1}{2}|u_i - v_i|_2^2 + \lambda |u_i|^p, \quad i = 1, 2, \ldots, d, \tag{10}$$

which can be solved based on Theorem 1 in [23]. Considering that the sparsity k is given as a prior, the Lagrangian multiplier λ can be analytically computed by $\lambda = \lambda^* = \frac{[2(1-p)]^{1-p}}{(2-p)^{2-p}}\theta_k^{2-p}$, and thus (7) follows.

Based on the above theorem, the SPARSE function can thus be defined and the k-sparse vector nearest to the current vector can be obtained through step 2.4 of the proposed algorithm.

When $p = 0, 1$, or, as a special case of $p \in (0, 1)$, i.e. $p = 1/2$, the SPARSE function has analytical expressions without iterations in Theorem 2. In fact, we have

1. l_0 constraint

$$u_i = \begin{cases} v_i, & |v_i| \geqslant \theta_k, \\ 0, & \text{else}; \end{cases} \tag{11}$$

2. l_1 constraint

$$u_i = \begin{cases} \mathrm{sign}(v_i)(|v_i| - \theta_k), & |v_i| \geqslant \theta_k, \\ 0, & \text{else}; \end{cases} \tag{12}$$

3. $l_{1/2}$ constraint

$$u_i = \begin{cases} \frac{2}{3}v_i\left(1 + \cos\left(\frac{2}{3}\pi - \frac{2}{3}\varphi\right)\right), & |v_i| \geqslant \theta_k, \\ 0, & \text{else}; \end{cases} \tag{13}$$

where $\varphi = \arccos\left(\frac{\sqrt{2}}{2}(\frac{\theta_k}{|v_i|})^{3/2}\right)$.

These analytical expressions are based on the following theorem.

Theorem 3. Given the sparsity k, $\boldsymbol{u} = \mathrm{SPARSE}(\boldsymbol{v}, k)$ defined by (11)–(13) are the optimal solution of optimization (6) corresponding to $p = 0, 1, 1/2$, respectively.

Proof. The Lagrangian function of (6) is given by

$$L(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{v}\|_2^2 + \lambda(\|\boldsymbol{u}\|_p^p - t) = \frac{1}{2}\Big(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u} - 2\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}\Big) + \lambda(\|\boldsymbol{u}\|_p^p - t).$$

Using the thresholding method corresponding to $p = 0, 1, 1/2$ to solve the formula

$$\frac{\partial L(\boldsymbol{u})}{\partial \boldsymbol{u}} = 0,$$

and considering the k-sparse prior, we can obtain the optimal solutions of (6), which is defined by (11)–(13), respectively. They are also known as hard thresholding [24], soft thresholding [4], and half thresholding [25], respectively.

Therefore, step 2.4 can be computed analytically when $p = 0, 1, 1/2$, making the proposed algorithm more efficient.

2.5 Initialization and stopping criterion

Now, we briefly discuss the initialization $\boldsymbol{w}(0)$ in step 2.2 and the stopping criterion in step 2.6 of our algorithm.

Since the proposed algorithm converges to a local optimum (detailed in the next subsection), the initial value should be carefully chosen. Two strategies are suggested as [16] indicated. The first is to initialize $\boldsymbol{w}(0)$ as the solution of the traditional PCA, which is expected to yield good sparse PC for its global optimality under the l_2-norm variance. The second is to run the algorithm multiple times with different initial values (e.g. the random vector, all-0 or all-1 vector), and output the solution with the maximal value of the objective. Due to the simplicity and efficiency of the latter strategy, it is adopted in our experiments.

It should be noted that two conditions are included in the stopping criterion of the proposed algorithm. Except common condition (1), there is another condition (2) in step 2.6. This criterion implies two requirements: $\boldsymbol{w}^{\mathrm{T}}(t+1)\boldsymbol{x}_i \neq 0$ guarantees that the algorithm should converge to a strictly local maximum (detailed in the next subsection); $\big|\mathrm{sign}(\boldsymbol{w}^{\mathrm{T}}(t+1))\big|\,|\mathrm{sign}(\boldsymbol{x}_i)| \neq 0$ avoids the abnormity that the non-zero elements of the optimal sparse PC $\boldsymbol{w}_j$ are located with the same positions of the zero elements of certain $\boldsymbol{x}_i$ such that $\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{x}_i = 0$ is always satisfied and the algorithm is trapped into infinite loops.

2.6 Convergence analysis

This section analyzes the convergence property of the proposed algorithm. Specifically, the following theorem shows that the proposed algorithm converges to a local maximum of (4) under the l_0 or l_1 constraint. Our proof is based on the convergence analysis of the PCA-L1 algorithm in [17], and can be considered as a generalization of its proof, by supplementally considering the sparsity constraint.

Theorem 4. When $p = 0$ or $p = 1$, the produced $\boldsymbol{w}(t)$ by step 2.2-2.6 of the robust sparse PCA algorithm satisfies the following properties:

(i) $\big\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}(t)\big\|_1$ is monotone increasing as t increases.

(ii) $\boldsymbol{w}(t)$ converges to a k-sparse vector $\boldsymbol{w}^* \in \mathbb{R}^d$.

(iii) Denote the k-dimensional subspace where the non-zero elements of $\boldsymbol{w}^*$ are located as Ω_k. Then $\boldsymbol{w}^*$ is a local maximum point of $\big\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}\big\|_1$ in Ω_k.

Proof. First, we prove that for a fixed sparsity k, $\frac{\mathrm{SPARSE}(\boldsymbol{v},k)}{\|\mathrm{SPARSE}(\boldsymbol{v},k)\|}$ produces the optimal solution of the following optimization when $p = 0, 1$:

$$\max_{\tilde{\boldsymbol{w}}} \tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{v},\ \text{s.t.}\ \|\tilde{\boldsymbol{w}}\|_2 = 1,\ \ \|\tilde{\boldsymbol{w}}\|_p^p \leqslant t. \tag{14}$$

When $p = 0$, it is obvious that the optimal solution of (14) lies on the k-dimensional subspace where the elements with the first k largest absolute value of $\boldsymbol{v}$ are located, and is parallel to $\boldsymbol{v}$. Together with the

constraint $\|\tilde{\boldsymbol{w}}\|_2 = 1$, we can conclude that $\frac{\text{SPARSE}(\boldsymbol{v},k)}{\|\text{SPARSE}(\boldsymbol{v},k)\|}$ is the optimal solution. When $p = 1$, a similar result was given in [16].

Then we prove the monotonicity of $\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}(t)\right\|_1$ and the convergence of $\boldsymbol{w}(t)$. In fact, the following inequalities hold:

$$\begin{aligned}\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}(t)\right\|_1 &= \sum_{i=1}^{n}\left|\boldsymbol{w}^{\mathrm{T}}(t)\boldsymbol{x}_i\right| = \boldsymbol{w}^{\mathrm{T}}(t)\sum_{i=1}^{n}p_i(t)\boldsymbol{x}_i \geqslant \boldsymbol{w}^{\mathrm{T}}(t)\sum_{i=1}^{n}p_i(t-1)\boldsymbol{x}_i \\ &\geqslant \boldsymbol{w}^{\mathrm{T}}(t-1)\sum_{i=1}^{n}p_i(t-1)\boldsymbol{x}_i = \sum_{i=1}^{n}\left|\boldsymbol{w}^{\mathrm{T}}(t-1)\boldsymbol{x}_i\right| = \left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}(t-1)\right\|_1.\end{aligned}$$

The first inequality holds because $p_i(t)\boldsymbol{w}^{\mathrm{T}}(t)\boldsymbol{x}_i \geqslant 0$ for all i. It then follows that $p_i(t)\boldsymbol{w}^{\mathrm{T}}(t)\boldsymbol{x}_i \geqslant p_i(t-1)\boldsymbol{w}^{\mathrm{T}}(t)\boldsymbol{x}_i$. The correctness of the second inequality is based on (14). Denote $\boldsymbol{v} = \sum_{i=1}^{n}p_i(t-1)\boldsymbol{x}_i$. Because $\boldsymbol{w}^{\mathrm{T}}(t)$ is the optimal solution of (14), this inequality naturally holds. Therefore, $\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}(t)\right\|_1$ increases with t. Considering the constraint $\|\boldsymbol{w}(t)\|_2 = 1$, we can conclude that $\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}(t)\right\|$ is bounded, yielding the result that $\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}(t)\right\|_1$ converges. It is clear that $\boldsymbol{w}(t)$ converges to a k-sparsity vector $\boldsymbol{w}^*$, for every $\boldsymbol{w}(t)$ is k-sparse during the iterations.

We now prove that $\boldsymbol{w}^*$ is a local maximum of $\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}\right\|_1$ in Ω_k. For $i = 1, 2, \ldots, n$, if $\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x}_i < 0$, then let $p_i = -1$; otherwise let $p_i = 1$. Based on the stopping criterion, for all $\boldsymbol{x}_i$s that satisfy $\left|\text{sign}(\boldsymbol{w}^{*\mathrm{T}})\right|\left|\text{sign}(\boldsymbol{x}_i)\right| \neq 0$, $\boldsymbol{w}^{*\mathrm{T}}p_i x_i > 0$ holds. For such $\boldsymbol{x}_i$, there exists a neighborhood $N(\boldsymbol{w}^*)$ of $\boldsymbol{w}^*$ in Ω_k, such that for any $\boldsymbol{w} \in N(\boldsymbol{w}^*)$, $\boldsymbol{w}^{\mathrm{T}}p_i\boldsymbol{x}_i \geqslant 0$ holds, i.e. $\boldsymbol{w}^{\mathrm{T}}p_i\boldsymbol{x}_i = \left|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i\right|$. For $\boldsymbol{x}_i$ that satisfies $\left|\text{sign}(\boldsymbol{w}^{*\mathrm{T}})\right|\left|\text{sign}(\boldsymbol{x}_i)\right| \neq 0$, it is obvious that its k elements located in Ω_k are all zeros, leading to the result that $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i = \boldsymbol{w}^{\mathrm{T}}p_i\boldsymbol{x}_i = \left|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i\right| = 0$ holds for any $\boldsymbol{w} \in \Omega_k$. Therefore, for any $\boldsymbol{w} \in N(\boldsymbol{w}^*)$, $\boldsymbol{w}^{\mathrm{T}}p_i\boldsymbol{x}_i = \left|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i\right|$. Based on the result about (14) derived before, we can conclude that $\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}^*\right\|_1 = \boldsymbol{w}^{*\mathrm{T}}\sum_{i=1}^{n}p_i\boldsymbol{x}_i \geqslant \boldsymbol{w}^{\mathrm{T}}\sum_{i=1}^{n}p_i\boldsymbol{x}_i = \left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}\right\|_1$ holds for all $\boldsymbol{w} \in N(\boldsymbol{w}^*)$. Thus, $\boldsymbol{w}^*$ is a solution in Ω_k such that $\left\|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w}\right\|_1$ attains a local maximum.

The above theorem implies the convergence of the proposed algorithm when $p = 0, 1$. Due to the complex expression of the SPARSE function for $p \in (0, 1)$, we have not achieved similar theoretical result in this case. However, monotone increasing property of the objective function can always be observed empirically. Specifically, when $p = 1/2$, a special case of $p \in (0, 1)$, the algorithm always performs as well as, or even better than $p = 0$ and $p = 1$ in the experiments as will be shown in Section 3. Therefore, we consider it as a good choice for robust sparse PCA, too.

2.7 Computation complexity

We then discuss the computational complexity of the proposed algorithm in comparison with the current sparse PCA methods.

The complexity of the proposed algorithm is mainly determined by the iteration process of the algorithm. Note that, when $p = 0, 1, 1/2$, only simple computation is involved in the iteration, and the complexity of the proposed algorithm is around $O(mnd\log d) \times n_{\text{it}}$, where n_{it} is the number of iterations, which approximately linearly increases with the size and the dimensionality of the data. The complexity of the proposed algorithm is not much higher than, or even lower than those of the current sparse PCA methods, such as $O(mnd^3) \times n_{\text{it}}$ of SPCA, $O(mnd^4\log d) \times n_{\text{it}}$ of DSPCA, $O(mnd\log d) \times n_{\text{it}}$ of EMPCA, $O(mnd^2) \times n_{\text{it}}$ of ALSPCA, $O(mnd) \times n_{\text{it}}$ of GPower, and $O(mn^3d) \times n_{\text{it}}$ of PathSPCA. Considering the model possessing the supplemental robustness property, such a computation complexity is satisfactory. When $0 < p < 1$ and $p \neq 1/2$, although more iterations are needed to compute the SPARSE function, the number can be bounded, and thus the complexity is not much higher.

3 Experiments

To verify the effectiveness of the proposed algorithm, especially on the data sets with noises and outliers, we applied it to a series of synthetic and pattern recognition data sets. We also compare the robust

Table 1 Results obtained by the traditional PCA, PCA-L1, the 12 current sparse PCA methods, and the robust sparse PCA algorithms on the synthetic data

Method	PC1	PC2	ARE
PCA	$(0.6811, 0.7322)$	$(-0.7322.6811)$	1.0155
PCA-L1	$(0.9699, 0.2435)$	$(-0.2435.0.9699)$	0.6286
SPCA, PathSPCA, EMPCA, GPower$_{l_1}$, GPower$_{l_0}$, DSPCA ALSPCA, sPCA-rSVD-l_0, sPCA-rSVD-l_1, sPCA-rSVD-SCAD	$(0, 1)$	$(1, 0)$	1.2500
GPower$_{l_1,m}$	$(0, 1)$	$(0.9945, 0.1048)$	1.2500
GPower$_{l_0,m}$	$(0.6766, 0.7363)$	$(0, 0)$	1.0188
sPCAgrid, RSPCA-l_0, RSPCA-l_1, RSPCA-$l_{1/2}$	$(1, 0)$	$(0, 1)$	0.5720

sparse PCA method with current methods, including the traditional PCA, PCA-L1 [17], and 12 sparse PCA methods, including SPCA [6], DSPCA [7], PathSPCA [14], EMPCA [9], GPower$_{l_0}$, GPower$_{l_1}$, GPower$_{l_0,m}$, GPower$_{l_1,m}$ [10], ALSPCA [12], sPCA-rSVD-l_0, sPCA-rSVD-l_1, sPCA-rSVD-SCAD [8]. We also implemented the sPCAgrid [15] algorithm involving the robustness issue of sparse PCA. For the proposed algorithm, we choose the value of p as $p = 1, 0, 1/2$, denoted by RSPCA$_{l_1}$ [16], RSPCA$_{l_0}$ and RSPCA$_{l_{1/2}}$, respectively. We choose these three values for the computational efficiency. Besides, $p = 1/2$ had empirically demonstrated its representative among the $l_p(0 < p < 1)$ constraints [26–28].

3.1 A synthetic problem

We first applied the robust sparse PCA algorithm to a synthetic 2-dimensional data set, denoted by $\{x_i, y_i\}_{i=1}^{50}$. The data set is generated as follows: first pick x_i from -2.4 to 2.5 with interval 0.1; then generate corresponding y_i from Gaussian distribution $N(0, \sigma^2)$ for each x_i, where σ^2 is set to 0.25; finally, set the value of y_i corresponding to $x_i = 1.3$ and $x_i = 1.5$ around 7 as outliers. It is obvious that the first PC of the data should be $(1, 0)$, discarding the outliers.

Table 1 summarizes the results obtained by the traditional PCA, PCA-L1, the 13 current sparse PCA methods and the robust sparse PCA algorithm. The first two PC vectors and the average residual error (ARE in brief) of the projected data on the first PC (the residual error on the first PC of x_i is defined by $e_i = \left\|(x_i, y_i)^{\mathrm{T}} - \boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}(x_i, y_i)^{\mathrm{T}}\right\|_2$, where $\boldsymbol{w}$ is the first PC obtained by the corresponding algorithm). As can be seen, the RSPCA$_{l_0}$, RSPCA$_{l_1}$, RSPCA$_{l_{1/2}}$, attain correct PCs, showing their robustness, while all of the other methods, except sPCAgrid, are affected by outliers and give wrong results. Besides, the ARE obtained by the robust sparse PCA algorithms are lower than those of the other methods, which further verifies the effectiveness of the proposed method.

3.2 Hastie data

We now consider the Hastie data set, which is one of the most often used data sets in the previous sparse PCA research [6–8]. Hastie data are generated through two steps: first 3 hidden factors were constructed as

$$V_1 \backsim N(0, 290),\ V_2 \backsim N(0, 300),\ V_3 = -0.3V_1 + 0.925V_2 + \varepsilon,$$

where $\varepsilon \backsim N(0, 1)$, and V_1, V_2, ε are independent; then 10 variables were generated as

$$\begin{aligned} x_i &= V_1 + \varepsilon_i,\ i = 1, 2, 3, 4, \\ x_i &= V_2 + \varepsilon_i,\ i = 5, 6, 7, 8, \\ x_i &= V_3 + \varepsilon_i,\ i = 9, 10, \end{aligned} \tag{15}$$

where $\varepsilon_i \backsim N(0, \sigma^2),\ i = 1, 2, \ldots, 10$ are independent, and σ^2 is the noise extent. Intrinsically, the first PC of the Hastie data is determined by (x_5, x_6, x_7, x_8) corresponding to V_2, and the second by (x_1, x_2, x_3, x_4) corresponding to V_1. Therefore, the PCs of the data are intrinsically sparse. We added noises and outliers with different extents to evaluate the performance of the proposed algorithm. The test data sets were generated as follows:

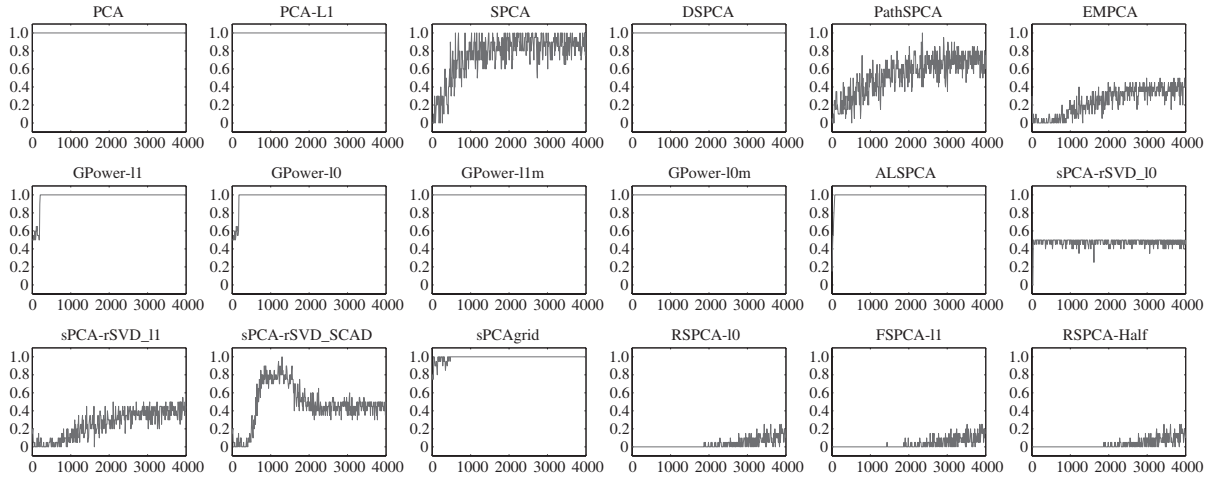

Figure 2 The tendency curves of the misidentification rate with respect to the noise extent σ^2 of the Hastie data. The σ^2 and the misidentification rate are depicted with horizontal and vertical axes, respectively.

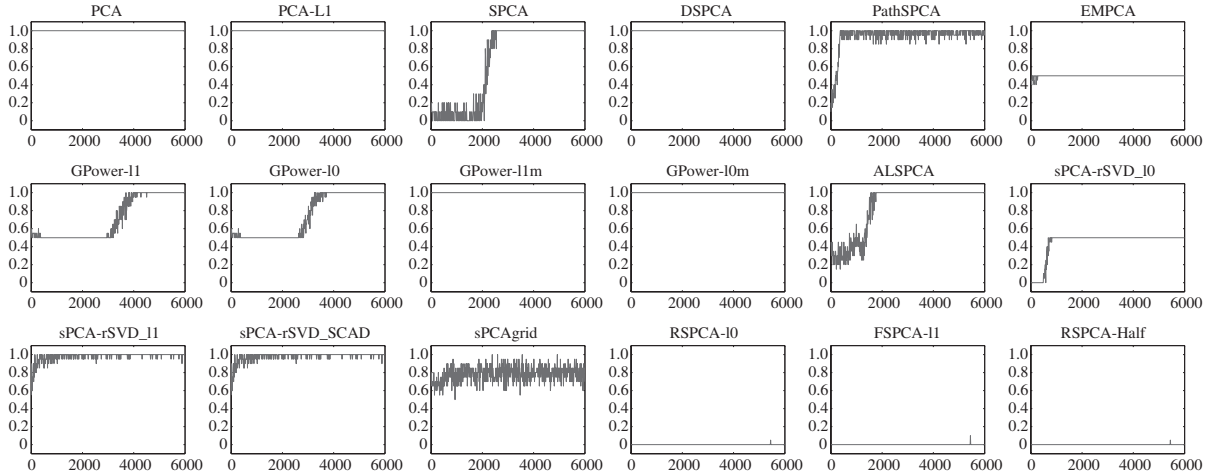

Figure 3 The tendency curves of the misidentification rate with respect to the outlier extent δ^2 of the Hastie data. The δ^2 and the misidentification rate are depicted with horizontal and vertical axes, respectively.

1. Noise data sets: Contain 4000 data sets, each with size 10000. Each data set was generated by the Hastie distribution as in (15) with noise extents $\sigma^2 = 1, \ldots, 4000$;

2. Outlier data sets: Contain 6000 data sets, each with size 10000. 9500 points in each data were generated by the Hastie distribution with noise extent $\sigma^2 = 1$, and 500 outliers. Specifically, the first 8 dimensions of the outliers are zeros, and the 9-th and 10-th dimension were generated by Gaussian distribution $N(0, \delta^2)$ with outlier extents $\delta^2 = 1, \ldots, 6000$.

We applied the traditional PCA, PCA-L1, 13 current sparse PCA methods and the robust sparse PCA method to the above data sets, and calculate the corresponding misidentification rates (MR in brief) of each method. The MR values of each method with varying σ^2 and δ^2 are depicted in Figures 2 and 3, respectively[1)]. Besides, Tables 2 and 3 summarize the first two PCs of each method at the maximum noise extent ($\sigma^2 = 4000$) and the maximum outlier extent ($\delta^2 = 6000$).

It is easy to observe that the robust sparse PCA algorithms perform more robust than other sparse PCA methods. In particular, the MR tendency curves of the RSPCA$_{l_1}$, RSPCA$_{l_0}$ and RSPCA$_{l_{1/2}}$ are always near zero. That is, the RSPCA algorithms generally achieve the correct information of the sparse PCs, while the other methods always cannot. In Tables 2 and 3, it is evident that RSPCA$_{l_1}$, RSPCA$_{l_0}$ and RSPCA$_{l_{1/2}}$ capture the correct sparse PCs while almost all of the other methods fail to attain this task.

1) The tendency in figures are based on calculating the average of every ten successive MR values.

Table 2 The first two PCs obtained by different algorithms on the Hastie data with noise extent $\sigma^2 = 4000$

PCA		PCA-L1		SPCA		DSPCA		PathSPCA	
$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$
−0.1422	−0.544	−0.1183	0.4433	0.622	0	0	−0.8983	0	−0.576
−0.0815	−0.4465	−0.0997	0.4244	0.4177	0	0	−0.0935	0	−0.449
−0.1017	−0.4615	−0.124	0.4867	0.4689	0	0	−0.3314	0	−0.4828
−0.1134	−0.4532	−0.1119	0.5516	0.4677	0	0	−0.2729	0	−0.4833
0.4067	−0.1173	0.3786	0.2112	0	0	0	0	−0.5011	0
0.4095	−0.1285	0.3778	0.074	0	0.4602	0.3992	0	−0.5057	0
0.4346	−0.1164	0.4449	0.0972	0	0.8543	0.891	0	−0.5057	0
0.389	−0.2012	0.3543	0.1325	0	0.2032	0.0984	0	−0.4872	0
0.3558	0.045	0.4151	0.0677	0	0	0	0	0	0
0.3874	0.003	0.4079	−0.0231	0	0.1308	0.1925	0	0	0

EMPCA		ALSPCA		GPower_{l_1,l_0}		$\text{GPower}_{l_1,m}$		$\text{GPower}_{l_0,m}$		sPCA-rSVD_{l_0}	
$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$
0	−0.576	0	1	0	0	−0.1475	−0.545	−0.1534	−0.553	0	−0.4788
0	−0.449	0	0	0	0	−0.0859	−0.446	−0.0892	−0.4391	0	−0.5204
0	−0.4828	0	0	0	0	−0.1062	−0.4626	−0.1099	−0.4565	0	−0.4353
0	−0.4833	0	0	0	0	−0.1179	−0.4546	−0.1214	−0.4456	0	−0.5571
0	0	0	0	1	0	0.4055	−0.1156	0.4077	−0.0844	0.5147	0
−0.5069	0	0	0	0	0	0.4083	−0.1272	0.4064	−0.1459	0	0
−0.5427	0	−1	0	0	1	0.4334	−0.1158	0.4309	−0.1445	0.4913	0
−0.4792	0	0	0	0	0	0.3871	−0.201	0.3852	−0.2089	0	0
0	0	0	0	0	0	0.356	0	0.3563	0.0336	0.4819	0
−0.4678	0	0	0	0	0	0.3874	0	0.3864	−0.0178	0.5113	0

sPCA-rSVD_{l_1}		$\text{sPCA-rSVD}_{\text{SCAD}}$		sPCAgrid		RSPCA_{l_0}		RSPCA_{l_1}		$\text{RSPCA}_{l_{1/2}}$	
$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$
0	−0.4692	0	−0.4788	0	0	0	0.5497	0	0.6426	0	0.5497
0	−0.5279	0	−0.5204	0.0016	0	0	0.4617	0	0.3668	0	0.4617
0	−0.4088	0	−0.4353	−0.0016	0	0	0.4866	0	0.4774	0	0.4866
0	−0.5779	0	−0.5571	0	0	0	0.4979	0	0.4739	0	0.4979
0.6474	0	0.5147	0	0	1	0.4987	0	0.4707	0	0.4987	0
0	0	0	0	0	0	0.493	0	0.3607	0	0.493	0
0.3635	0	0.4913	0	0	0	0.5092	0	0.4666	0	0.5092	0
0	0	0	0	0	0	0.4989	0	0.6563	0	0.4989	0
0.1244	0	0.4819	0	0	0	0	0	0	0	0	0
0.6582	0	0.5113	0	1	0	0	0	0	0	0	0

3.3 MNIST handwriting digit data

The robust sparse PCA method was also implemented to the MNIST handwriting digit data set. The MNIST handwriting digit data set contains the images (28×28 pixels) of ten digits from "0" to "9" with about 7000 data per digit. Figure 4 shows some samples from the data set. It is obvious that the digit images are sparse (the gray values of the background are all zeros), and thus suitable for evaluating the performance of sparse PCA methods. We applied the aforementioned methods to all ten digits and the proposed algorithm always attained a very robust performance. We show the results of digit "6" for illustration.

Firstly, the data sets with noises and outliers were generated as follows:

1. Noise data set: Randomly choose 2000 images and randomly add rectangular noise consisting of

Table 3 The first two PCs obtained by different algorithms on the Hastie data with outlier extent $\delta^2 = 6000$

PCA		PCA-L1		SPCA		DSPCA		PathSPCA	
$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$
−0.0944	0.4807	−0.097	0.4706	0.4999	0	0	0.4995	0	0
−0.0945	0.4807	−0.0967	0.47	0.4999	0	0	0.4993	0	0.1229
−0.0941	0.4809	−0.0964	0.4702	0.5	0	0	0.4999	0	0
−0.0943	0.481	−0.0964	0.4699	0.5002	0	0	0.5012	0	0
0.3752	0.1361	0.3764	0.1615	0	0.0046	0.0993	0	−0.4626	0
0.3749	0.1363	0.3766	0.1619	0	0.0023	0.0985	0	−0.4624	0
0.3748	0.1358	0.3767	0.1615	0	0	0.098	0	−0.4621	0
0.3747	0.1359	0.3765	0.1617	0	0	0.0977	0	0	−0.5129
0.4545	−0.0256	0.4435	−0.0398	0	0.8965	0.9516	0	−0.5989	0.1331
0.4423	−0.0247	0.4458	−0.0988	0	0.4429	0.236	0	0	−0.8392

EMPCA		ALSPCA		GPower_{l_1,l_0}		$\text{GPower}_{l_1,m}$		$\text{GPower}_{l_0,m}$		sPCA-rSVD_{l_0}	
$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$
0	−0.4999	0	0	0	0	−0.0977	−0.4797	−0.0987	0.4793	0	0.5003
0	−0.4999	0	0	0	0	−0.0978	−0.4796	−0.0988	0.4793	0	0.5005
0	−0.5	0	0	0	0	−0.0974	−0.4798	−0.0984	0.4795	0	0.4998
0	−0.5002	0	1	0	0	−0.0975	−0.48	−0.0986	0.4796	0	0.4994
−0.4626	0	−1	0	0	0	0.3743	−0.1401	0.374	0.1414	−0.4457	0
−0.4624	0	0	0	0	0	0.374	−0.1404	0.3737	0.1417	0	0
−0.4621	0	0	0	0	0	0.3739	−0.1398	0.3736	0.1411	0	0
0	0	0	0	0	0	0.3738	−0.1399	0.3735	0.1412	−0.4456	0
−0.5989	0	0	0	1	0	0.4547	0.0207	0.4547	−0.0195	−0.5498	0
0	0	0	0	0	1	0.4424	0.0199	0.4425	−0.0182	−0.5482	0

sPCA-rSVD_{l_1}		$\text{sPCA-rSVD}_{\text{SCAD}}$		sPCAgrid		RSPCA_{l_0}		RSPCA_{l_1}		$\text{RSPCA}_{l_{1/2}}$	
$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_1$	$\boldsymbol{w}_2$
0	0	0	0	0	0.4085	0	0.5002	0	0.5002	0	0.5002
0	0	0	0	0	0.4965	0	0.4999	0	0.4998	0	0.4999
0	0	0	0	0	0.5228	0	0.5	0	0.5	0	0.5
0	0	0	0	0	0.4993	0	0.5	0	0.5	0	0.5
−0.0008	−0.0002	−0.0007	−0.0002	0.3968	0	0.5003	0	0.5057	0	0.5003	0
−0.0001	0	−0.0001	0	0.4155	0	0.5	0	0.5004	0	0.5	0
0	0	0	0	0.4155	0	0.4999	0	0.4976	0	0.4999	0
−0.0007	0	−0.0007	0	0.3998	0	0.4998	0	0.4963	0	0.4998	0
−0.9999	0	−0.9999	0	0.4324	0	0	0	0	0	0	0
0	−0.9999	0	0	0.3878	0	0	0	0	0	0	0

random black and white dots of size 10×18 (samples are shown in the second line of 4) into them;

2. Outlier data set: Add 3000 dummy images of the same size as the original images consisting of random black and white dots to the data set.

We applied the traditional PCA, PCA-L1, 13 current sparse PCA methods and the proposed algorithms to the above two data sets, calculating first 100 PCs. Then we used the PCs so obtained to reconstruct the original images, and assessed the quality of the reconstruction quantitatively via the average reconstruction error defined by

$$\text{ERR}(m) = \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}_i^{\text{org}} - \sum_{j=1}^{m}\boldsymbol{w}_j\boldsymbol{w}_j^{\text{T}}\boldsymbol{x}_i\right\|_2, \tag{16}$$

where m is the number of PCs, n is the number of the original images, $\boldsymbol{w}_j$ is the jth PC obtained by

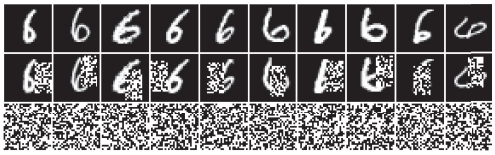

Figure 4 The first row shows typical samples from the original MNIST handwriting digit images; the second row shows typical samples of handwriting digit images with noise; the third row shows typical samples of outliers added to the data.

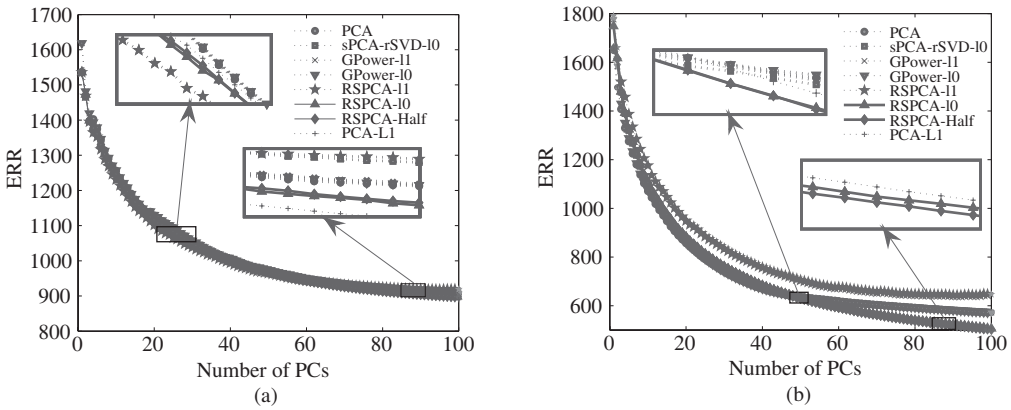

Figure 5 The average reconstruction error curves of different algorithms on the MNIST data: (a) noise case; (b) outlier case.

a certain algorithm, $\boldsymbol{x}_i^{\text{org}}$ and $\boldsymbol{x}_i$ are the ith original image and the corresponding really adopted one in the data set, respectively. Figure 5 shows the ERR tendency curves of different methods in noise and outlier data sets, respectively. Table 4 summarizes the average sparsity, average reconstruction error and computation time of different methods.

Figure 5 shows that the RSPCA algorithms, especially the proposed RSPCA$_{l_0}$ and RSPCA$_{l_{1/2}}$, have the best performance among all of the employed algorithms except PCA-L1. Specifically, the proposed algorithms always capture the lowest ERR except PCA-L1 with different PCs (especially $\geqslant$ 40 PCs). Table 4 further shows this result quantitatively. As can be seen, the ERRs obtained by RSPCA$_{l_0}$ and RSPCA$_{l_{1/2}}$ are about 2% lower than the other sparse PCA methods in noise data set, and 15% lower in outlier data set. We here give some comments on the performance of PCA-L1. It can be seen that the proposed algorithms achieve almost the same result (even better in some cases) as PCA-L1, and take less time. Although PCA-L1 is not a sparse PCA algorithm and should be expected to capture the intrinsic information of data better, it is impressing that the proposed algorithms always tend to attain comparable or even better reconstruction than PCA-L1. This is due to the intrinsic sparsity of the data used in experiments. For such data sets with essential sparsity, the proposed algorithm can always be preferred to be utilized. Besides, from Table 4 it can be observed that the computation time of the proposed algorithm is comparable with or even less than the current sparse PCA methods (that some sparse PCA methods are not involved is due to the fact that they cannot converge in a reasonable time). Considering the robustness of the proposed algorithm, its efficiency is evident.

We then compared the classification performance of the proposed method with other PCA and sparse PCA methods on this handwriting digit data set. We randomly chose 1000 and 500 data per digit to form the training and testing set, respectively. To compare the robustness of each method, we also added noise, as in reconstruction experiment, to 20% of the training data which were randomly chosen. In summary, the final training set consisted of 10000 data with 2000 being noisy and the testing set consists of 5000 data. We first extracted features from the original training data using different methods, and then applied one nearest neighbor (1-NN) classifier to the data projected to feature space which was with much lower dimension than the original data space. The number of features was varying from 10 to 100,

Table 4 The results of the MNIST handwriting data experiment (100 PCs). The 2–4 and the 5–7 columns are with the noise and outlier case, respectively

Method	Average sparstiy	ERR	CPU time (s)	Average sparstiy	ERR	CPU time (s)
PCA	784	905.0617	2.26	784	570.3221	2.59
PCA-L1	784	889.2563	561.23	784	507.5231	936.77
sPCA-rSVD$_{l_0}$	400	914.5049	894.66	400	573.1356	1222.25
GPower$_{l_1}$	609.13	906.1718	134.78	591.77	571.294	221.17
GPower$_{l_0}$	423.67	906.0196	94.14	607.62	571.0965	143.29
RSPCA$_{l_1}$	400	918.8353	264.46	400	646.3972	521.28
RSPCA$_{l_0}$	400	896.3046	307.25	400	505.9623	560.30
RSPCA$_{l_{1/2}}$	400	897.0606	284.82	400	505.6411	546.34

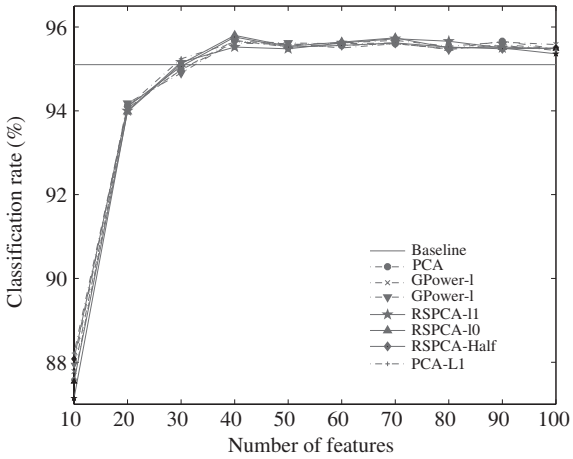

Figure 6 The classification rates of different algorithms on the MNIST data.

and the classification rates were summarized in Figure 6. The 1-NN classifier was also applied to the original data as baseline.

Figure 6 shows that all the methods implemented improve the baseline classification rate when 40 or more features are extracted and the proposed methods can always achieve the highest or comparable rate in each case. Specifically, RSPCA$_{l_0}$ achieves the highest classification rate, 95.80%, using 40 features, among all the results obtained. The effectiveness of the proposed method can thus be substantiated.

4 Conclusion

We have proposed robust sparse PCA method, aiming at enhancing the robustness of sparse PCA calculation. The proposed algorithm is very efficient, and its computational complexity approximately linearly increases with both the size and the dimensionality of the given data. This complexity is comparable as, or even better than that of current sparse PCA methods. Besides, we have proved that the proposed algorithm can converge to a reasonable local optimum of the robust sparse PCA model. A series experiments on synthetic and digit image data sets have been conducted to substantiate the effectiveness of the proposed algorithm. In particular, the proposed algorithm depicts significant robustness on the data with noises and outliers, as compared with other sparse PCA methods.

However, there are some problems to be further investigated. First, the convergence of the proposed method with general l_p $(0 < p < 1)$ norm, which has not been proved, should be analyzed further theoretically. Second, because the proposed algorithm can only be guaranteed to converge to a local optimum, a more powerful algorithm with global optimality is expected. Also, the proposed method needs to be further evaluated in more applications.

Acknowledgements

This research was supported by National Basic Research Program of China (973) (Grant No. 2013CB329404) and National Natural Science Foundation of China (Grant Nos. 61373114, 11131006).

References

1 Jollife I. Principal Component Analysis. New York: Springer-Verlag, 1986
2 Jollife I. Rotation of principal components: choice of normalization constraints. J Appl Stat, 1995, 22: 29–35
3 Cadima J, Jollife I. Loadings and correlations in the interpretation of principal components. J Appl Stat, 1995, 22: 203–214
4 Tibshirani R. Regression shrinkage and selection via the Lasso. J Roy Statist Soc Ser B Met, 1996, 58: 267–288
5 Jollife I, Uddin M. A modied principal component technique based on the Lasso. J Comput Graph Stat, 2003, 12: 531–547
6 Zou H, Hastie T, Tibshirani R. Sparse principal component analysis. J Comput Graph Stat, 2006, 15: 265–286
7 d'Aspremont A, El Ghaoui L, Jordan M I, et al. A direct formulation for sparse PCA using semidefinite programming. SIAM Rev, 2007, 49: 434–448
8 Shen H P, Huang J Z. Sparse principal component analysis via regularized low rank matrix approximation. J Multivariate Anal, 2008, 99: 1015–1034
9 Sigg C D, Buhmann J M. Expectation maximization for sparse and non-negative PCA. In: Proceedings of the 25th International Conference on Machine Learning, Helsinki, 2008. 960–967
10 Journée M, Nesterov Y, Richtárik P, et al. Generalized power method for sparse principal component analysis. J Mach Learn Res, 2010, 11: 517–553
11 Sriperumbudur B K, Torres D A, Lanckriet G R. Sparse eigen methods by D.C. programming. In: Proceedings of the 24th International Conference on Machine learning, Corvallis, 2007. 831–838
12 Lu Z S, Zhang Y. An augmented Lagrangian approach for sparse principal component analysis. Math Program, 2012, 135: 149–193.
13 Moghaddam B, Weiss Y, Avidan S. Spectral bounds for sparse PCA: exact and greedy algorithms. In: Proceedings of the 19th Conference on Neural Information Processing Systems, Vancouver, 2005. 915–922
14 d'Aspremont A, Bach F R, Ghaoui L E. Optimal solutions for sparse principal component analysis. J Mach Learn Res, 2008, 9: 1269–1294
15 Croux C, Filzmoser P, Fritz H. Robust sparse principal component analysis. Technometrics, 2013, 55: 202–214
16 Meng D Y, Zhao Q, Xu Z B. Improve robustness of sparse PCA based on L_1-norm maximization. Patt Recog, 2012, 45: 487–497
17 Kwak N. Principal component analysis based on L_1-norm maximization. IEEE Trans Patt Anal Mach Intell, 2008, 30: 1672–1680
18 De la Torre F, Black M J. A framework for robust subspace learning. Int J Comput Vis, 2003, 54: 117–142
19 Aanas H, Fisker R, Astrom K, et al. Robust factorization. IEEE Trans Patt Anal Mach Intell, 2002, 24: 1215–1225
20 Ding C, Zhou D, He X, et al. R1-PCA: rotational invariant L_1-norm principal component analysis for robust subspace factorization. In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, 2006. 281–288
21 Baccini A, Besse P, Falguerolles A D. A L_1-norm PCA and a heuristic approach. In: Diday E, Lechevalier Y, Opitz O, eds. Ordinal and Symbolic Data Analysis. New York: Springer-Verlag, 1996. 359–368
22 Ke Q, Kanade T. Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington D.C.: IEEE, 2005. 739–746
23 Marjanovic G, Solo V. On l_q optimization and matrix completion. IEEE Trans Signal Process, 2012, 60: 5714–5724
24 Blumensath T, Davies M E. Iterative hard thresholding for compressed sensing. Appl Comput Harmonic Anal, 2009, 27: 265–274
25 Xu Z B, Chang X Y, Xu F M, et al. $L_{1/2}$-regularization: a thresholding representation theory and a fast solver. IEEE Trans Neural Netw Learn Syst, 2012, 23: 1013–1027
26 Xu Z B, Zhang H, Wang Y, et al. $L_{1/2}$-regularization. Sci China Inf Sci, 2010, 53: 1159–1169
27 Zeng J S, Fang J, Xu Z B. Sparse SAR imaging based on $L_{1/2}$ regularization. Sci China Inf Sci, 2012, 55: 1755–1775
28 Xu Z B, Guo H L, Wang Y, et al. Representative of $L_{1/2}$ regularization among L_q $(0 < q \leqslant 1)$ regularizations: an experimental study based on phase diagram. Acta Autom Sin, 2012, 38: 1225–1228