Neurocomputing 153 (2015) 180–190

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A block coordinate descent approach for sparse principal component analysis

Qian Zhao [a], Deyu Meng [a,*], Zongben Xu [a], Chenqiang Gao [b]

[a] Institute for Information and System Sciences, School of Mathematics and Statistics, and Ministry of Education Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, PR China
[b] Chongqing Key Laboratory of Signal and Information Processing, Chongqing University of Posts and Telecommunications, Chongqing 400065, PR China

ARTICLE INFO

Article history:
Received 23 December 2013
Received in revised form
14 October 2014
Accepted 12 November 2014
Communicated by Zhi Yong Liu
Available online 28 November 2014

Keywords:
Block coordinate descent
Nonnegativity
Principal component analysis
Sparsity

ABSTRACT

There are mainly two methodologies utilized in current sparse PCA calculation, the greedy approach and the block approach. While the greedy approach tends to be incrementally invalidated in sequentially generating sparse PCs due to the cumulation of computational errors, the block approach is difficult to elaborately rectify individual sparse PCs under certain practical sparsity or nonnegative constraints. In this paper, a simple while effective block coordinate descent (BCD) method is proposed for solving the sparse PCA problem. The main idea is to separate the original sparse PCA problem into a series of simple sub-problems, each having a closed-form solution. By cyclically solving these sub-problems in an analytical way, the BCD algorithm can be easily constructed. Despite its simplicity, the proposed method performs surprisingly well in extensive experiments implemented on a series of synthetic and real data. In specific, as compared to the greedy approach, the proposed method can iteratively ameliorate the deviation errors of all computed sparse PCs and avoid the problem of accumulating errors; as compared to the block approach, the proposed method can easily handle the constraints imposed on each individual sparse PC, such as certain sparsity and/or nonnegativity constraints. Besides, the proposed method converges to a stationary point of the problem, and its computational complexity is approximately linear in both data size and dimensionality, which makes it well suited to handle large-scale problems of sparse PCA.

1. Introduction

Principal component analysis (PCA) is one of the most classical and popular tools for data analysis and dimensionality reduction, and has a wide range of successful applications throughout science and engineering [1]. By seeking the so-called principal components (PCs), along which the data variance is maximally preserved, PCA can always capture the intrinsic latent structure underlying data. Such information greatly facilitates many further data processing tasks, such as feature extraction and pattern recognition.

Despite its many advantages, the conventional PCA suffers from the fact that each component is generally a linear combination of all data variables, and all weights in the linear combination, also called loadings, are typically non-zeros. In many applications, however, the original variables have meaningful physical interpretations. In biology, for example, each variable of gene expression data corresponds to a certain gene. In such cases, the derived PC loadings are always expected to be sparse (i.e. contain fewer non-zeros) so as to facilitate their interpretability. Moreover, in certain applications, such as financial asset trading, the sparsity of the PC loadings is especially expected since fewer nonzero loadings imply fewer transaction costs.

Accordingly, sparse PCA has attracted much attention in the recent decade, and a variety of methods for this topic have been developed [2–23]. The first attempt for this topic is to make certain post-processing transformation, e.g. rotation by Jolliffe [2] and simple thresholding by Cadima and Jolliffe [3], on the PC loadings obtained by the conventional PCA to enforce sparsity. Jolliffe et al. [4] further advanced a SCoTLASS algorithm by simultaneously calculating sparse PCs on the PCA model with additional l_1-norm penalty on loading vectors. Better results have been achieved by the SPCA algorithm of Zou et al. [5], which was developed based on iterative elastic net regression. D'Aspremont et al. [6] proposed a method, called DSPCA, for finding sparse PCs by solving a sequence of semidefinite programming (SDP) relaxations of sparse PCA. Shen and Huang [7] developed a series of methods called sPCA-rSVD (including $\text{sPCA}-\text{rSVD}_{l_0}$, $\text{sPCA}-\text{rSVD}_{l_1}$, and $\text{sPCA}-\text{rSVD}_{SCAD}$), computing sparse PCs by low-rank matrix factorization under multiple sparsity-including penalties. Journée et al. [8] designed four algorithms, denoted as GPower_{l_0},

[*] Corresponding author.
E-mail addresses: zhao.qian@stu.xjtu.edu.cn (Q. Zhao), dymeng@mail.xjtu.edu.cn (D. Meng), zbxu@mail.xjtu.edu.cn (Z. Xu), gaochenqiang@gmail.com (C. Gao).

http://dx.doi.org/10.1016/j.neucom.2014.11.038
0925-2312/© 2014 Elsevier B.V. All rights reserved.

Table 1
The general pros and cons of the greedy approach and the block approach for the sparse PCA problem.

	Greedy approach	Block approach
Pros	The first several sparse PCs can generally be properly extracted in a sequential way The sparse PCA calculation can be easily implemented under different sparsity parameter settings (i.e., t_i in Eq. (3) and (4))	Efficient to simultaneously attain large number of sparse PCs Convergence to a reasonable solution of the sparse PCA problem with respect to all sparse PCs sometimes can be proved (e.g., the ALSPCA method [15])
Cons	The computation for more sparse PCs tends to be incrementally invalidated due to the cumulation of computational errors, e.g., the SPCA method tends to be less effective in our colon data experiments when the number of sparse PCs are increasing (Section 3.2.2)	Difficult to elaborately rectify each individual sparse PC under certain requirements of sparse PCs (e.g. the sparsity or nonnegative constraints on sparse PCs), e.g., in our pitprops data experiments, the GPower$_{l_0,m}$ and GPower$_{l_1,m}$ methods cannot derive sparse PCs with preset cardinality settings (Section 3.2.1)

GPower$_{l_1}$, GPower$_{l_0,m}$, and GPower$_{l_1,m}$, respectively, for sparse PCA by formulating the issue as non-concave maximization problems with l_0- or l_1-norm sparsity-inducing penalties and extracting single unit sparse PC sequentially or block units ones simultaneously. Based on probabilistic generative model of PCA, some methods have also been attained [9–12], e.g. the EMPCA method derived by Sigg and Buhmann [9] for sparse and/or nonnegative sparse PCA. Sriperumbudur et al. [13,14] provided an iterative algorithm called DCPCA, where each iteration consists of solving a quadratic programming (QP) problem. Recently, Lu and Zhang [15] developed an augmented Lagrangian method (ALSPCA briefly) for sparse PCA by solving a class of non-smooth constrained optimization problems. Additionally, d'Aspremont et al. [16] derived a PathSPCA algorithm that computes a full set of solutions for all target numbers of nonzero coefficients. Very recently, Meng et al. [24] presented another path algorithm by utilizing the coordinate-pairwise updating strategy. The method can attain the entire spectrum of solutions of the problem, providing more insight for sparse PCA solution.

There are mainly two methodologies utilized in the aforementioned sparse PCA methods. The first is the greedy approach, including DSPCA [6], sPCA-rSVD [7], EMPCA [9], and PathSPCA [16]. These methods mainly focus on the solving of one-sparse-PC model, and more sparse PCs are sequentially calculated one-by-one on the deflated data matrix or data covariance [25]. The second is the block approach. Typical methods include SCoTLASS [4], GPower$_{l_0,m}$, GPower$_{l_1,m}$ [8], ALSPCA [15], etc. These methods aim to calculate multiple sparse PCs at once by utilizing certain block optimization techniques. The general pros and cons of both approaches are listed in Table 1 for easy comparison. All these properties have been extensively exhibited in our experiments, as introduced in Section 3.

In this paper, we design a surprisingly simple while effective block coordinate descent method for solving the sparse PCA problem. The main idea is to decompose the original large and complex problem of sparse PCA into a series of small subproblems, and then cyclically solve them. Each of these subproblems has a closed-form solution, which makes the new method very easy to implement. Despite its simplicity, the proposed method performs very well in sparse PCA calculation. On one hand, as compared to the greedy approach, attributed to its recursive updating over all sparse PC variables, the proposed method can iteratively ameliorate the deviation errors of all computed sparse PCs and avoid the problem of accumulating errors. On the other hand, as compared to the block approach, the new method can easily handle the constraints imposed on each individual sparse PC, such as certain sparsity and/or nonnegative constraints. Furthermore, the proposed method converges to a stationary solution of the original sparse PCA problem, and its computational complexity is approximately linear in both data size and dimensionality, which makes it well suited to handle large-scale problems of sparse PCA. The aforementioned properties have been extensively substantiated in experiments implemented on synthetic and real data.

In what follows, the main idea and the implementation details of the proposed method are first introduced in Section 2. Its convergence and computational complexity are also analyzed in this section. The effectiveness of the proposed method is comprehensively substantiated based on a series of empirical studies in Section 3. Then the paper is concluded with a summary and outlook for future research. Throughout the paper, we denote matrices, vectors and scalars by the upper-case bold-faced letters, lower-case bold-faced letters, and lower-case letters, respectively.

2. The block coordinate descent method for sparse PCA

In the following, we first introduce the fundamental models for the sparse PCA problem.

2.1. Basic models of sparse PCA

Denote the input data matrix as $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n]^T\in\mathbb{R}^{n\times d}$, where n and d are the size and the dimensionality of the given data, respectively. After a location transformation, we can assume all $\{\mathbf{x}_i\}_{i=1}^n$ to have zero mean. Let $\boldsymbol{\Sigma}=(1/n)\mathbf{X}^T\mathbf{X}\in\mathbb{R}^{d\times d}$ be the data covariance matrix.

The classical PCA can be solved through two types of optimization models [1]. The first is constructed by finding the $r(\le d)-$dimensional linear subspace where the variance of the input data $\mathbf{X}$ is maximized [26]. On this data-variance-maximization viewpoint, the PCA is formulated as the following optimization model:

$$\max_{\mathbf{V}} \operatorname{Tr}(\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}) \quad \text{s.t. } \mathbf{V}^T\mathbf{V}=\mathbf{I}, \tag{1}$$

where $\operatorname{Tr}(\mathbf{A})$ denotes the trace of the matrix $\mathbf{A}$ and $\mathbf{V}=(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_r)\in\mathbb{R}^{d\times r}$ denotes the array of PC loading vectors. The second is formulated by seeking the r-dimensional linear subspace on which the projected data and the original ones are as close as possible [27]. On this reconstruction-error-minimization viewpoint, the PCA corresponds to the following model:

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 \quad \text{s.t. } \mathbf{V}^T\mathbf{V}=\mathbf{I}, \tag{2}$$

where $\|\mathbf{A}\|_F$ is the Frobenius norm of $\mathbf{A}$, $\mathbf{V}\in\mathbb{R}^{d\times r}$ is the matrix of PC loading array and $\mathbf{U}=(\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_r)\in\mathbb{R}^{n\times r}$ is the matrix of projected data. The two models are intrinsically equivalent and can attain the same PC loading vectors [1].

Corresponding to the PCA models (1) and (2), the sparse PCA problem has the following two mathematical formulations[1]:

$$\max_{\mathbf{V}} \operatorname{Tr}(\mathbf{V}^T\boldsymbol{\Sigma}\mathbf{V}) \quad \text{s.t. } \mathbf{v}_i^T\mathbf{v}_i=1,\ \ \|\mathbf{v}_i\|_p\le t_i\ (i=1,2,\ldots,r), \tag{3}$$

[1] It should be noted that the orthogonality constraints of PC loadings in (1) and (2) are not imposed in (3) and (4). This is because simultaneously enforcing sparsity and orthogonality is generally a very difficult (and perhaps unnecessary) task. Like

and

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 \quad \text{s.t. } \mathbf{v}_i^T\mathbf{v}_i = 1, \;\; \|\mathbf{v}_i\|_p \le t_i \;(i=1,2,\ldots,r), \tag{4}$$

where $p=0$ or 1 and the corresponding $\|\mathbf{v}\|_p$ denotes the l_0- or the l_1-norm of $\mathbf{v}$, respectively. Note that the involved l_0 or l_1 penalty in the above models (3) and (4) tends to enforce sparsity of the output PCs. Methods constructed on Eq. (3) include SCoTLASS [4], DSPCA [6], DCPCA [13,14], ALSPCA [15], etc., and those related to Eq. (4) include SPCA [5], sPCA-rSVD [7], SPC [19], GPower [8], etc. In this paper, we will construct our method on the reconstruction-error-minimization model (4), while our experiments will verify that the proposed method also performs well based on the data-variance-maximization criterion.

2.2. Decompose original problem into small sub-problems

The objective function of the sparse PCA model (4) can be equivalently formulated as follows:

$$\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 = \|\mathbf{X}-\textstyle\sum_{j=1}^r \mathbf{u}_j\mathbf{v}_j^T\|_F^2 = \|\mathbf{E}_i-\mathbf{u}_i\mathbf{v}_i^T\|_F^2,$$

where $\mathbf{E}_i = \mathbf{X}-\sum_{j\neq i}\mathbf{u}_j\mathbf{v}_j^T$. It is then easy to separate the original large minimization problem, which is with respect to $\mathbf{U}$ and $\mathbf{V}$, into a series of small minimization problems, which are each with respect to a column vector $\mathbf{u}_i$ of $\mathbf{U}$ and $\mathbf{v}_i$ of $\mathbf{V}$ for $i=1,2,\ldots,r$, respectively, while keeping other variables fixed, as follows:

$$\min_{\mathbf{v}_i} \|\mathbf{E}_i-\mathbf{u}_i\mathbf{v}_i^T\|_F^2 \quad \text{s.t. } \mathbf{v}_i^T\mathbf{v}_i = 1, \;\; \|\mathbf{v}_i\|_p \le t_i, \tag{5}$$

and

$$\min_{\mathbf{u}_i} \|\mathbf{E}_i-\mathbf{u}_i\mathbf{v}_i^T\|_F^2. \tag{6}$$

Through cyclically optimizing these small sub-problems, the new method for solving the sparse PCA model (4) can then be naturally constructed. Note that each of the subproblem is not equivalent to the original problem, but the whole procedure deduces a block coordinate descent (BCD) approach for solving optimization (4). The details are analyzed in Section 2.5.

It is very fortunate that both the minimization problems in (5) and (6) have closed-form solutions. This implies that the to-be-constructed method can be fast and efficient, as presented in the following sub-sections.

2.3. The closed-form solutions of (5) and (6)

For the convenience of notation, we first rewrite (5) and (6) as the following forms:

$$\min_{\mathbf{v}} \|\mathbf{E}-\mathbf{u}\mathbf{v}^T\|_F^2 \quad \text{s.t. } \mathbf{v}^T\mathbf{v} = 1, \;\; \|\mathbf{v}\|_p \le t, \tag{7}$$

and

$$\min_{\mathbf{u}} \|\mathbf{E}-\mathbf{u}\mathbf{v}^T\|_F^2, \tag{8}$$

where $\mathbf{u}$ is n-dimensional and $\mathbf{v}$ is d-dimensional. Since the objective function $\|\mathbf{E}-\mathbf{u}\mathbf{v}^T\|_F^2$ can be equivalently transformed as

$$\|\mathbf{E}-\mathbf{u}\mathbf{v}^T\|_F^2 = \|\mathbf{E}\|_F^2 - 2\mathbf{u}^T\mathbf{E}\mathbf{v} + \mathbf{u}^T\mathbf{u}\mathbf{v}^T\mathbf{v},$$

(7) and (8)are equivalent to the following optimization problems, respectively:

$$\max_{\mathbf{v}} (\mathbf{E}^T\mathbf{u})^T\mathbf{v} \quad \text{s.t. } \mathbf{v}^T\mathbf{v} = 1, \;\; \|\mathbf{v}\|_p \le t, \tag{9}$$

and

$$\min_{\mathbf{u}} \mathbf{u}^T\mathbf{u} - 2(\mathbf{E}\mathbf{v})^T\mathbf{u}. \tag{10}$$

The closed-form solutions of (9) and (10), i.e. (7) and (8), can then be presented as follows.

We present the closed-form solution to Eq. (8) in the following theorem.

Theorem 1. *The optimal solution of Eq. (8) is* $\mathbf{u}^*(\mathbf{v}) = \mathbf{E}\mathbf{v}$.

The theorem is very easy to prove by calculating where the gradient of $\mathbf{u}^T\mathbf{u}-2(\mathbf{E}\mathbf{v})^T\mathbf{u}$ is equal to zero. We thus omit the proof.

In the $p=0$ case, the closed-form solution to (9) is presented in the following theorem. Here, we denote $\mathbf{w}=\mathbf{E}^T\mathbf{u}$, and $hard_\lambda(\mathbf{w})$ the hard thresholding function, whose i-th element corresponds to $I(|w_i| \ge \lambda)w_i$, where w_i is the i-th element of $\mathbf{w}$ and $I(x)$ (equals 1 if x is true, and 0 otherwise) is the indicator function. The proof of the theorem is provided in Appendix A.

Theorem 2. *The optimal solution of*

$$\max_{\mathbf{v}} \mathbf{w}^T\mathbf{v} \quad \text{s.t. } \mathbf{v}^T\mathbf{v} = 1, \;\; \|\mathbf{v}\|_0 \le t \tag{11}$$

is given by

$$\mathbf{v}_0^*(\mathbf{w},t) = \begin{cases} \phi, & t<1, \\ \dfrac{hard_{\theta_k}(\mathbf{w})}{\|hard_{\theta_k}(\mathbf{w})\|_2}, & k \le t < k+1 \;(k=1,2,\ldots,d-1), \\ \dfrac{\mathbf{w}}{\|\mathbf{w}\|_2}, & t \ge d, \end{cases}$$

where θ_k denotes the k-th largest element of $|\mathbf{w}|$.

In the above theorem, ϕ denotes the empty set, implying that when $t<1$, the optimum of Eq. (11) does not exist.

In the $p=1$ case, Eq. (7) has the following closed-form solution. In the theorem, we denote $f_{\mathbf{w}}(\lambda) = soft_\lambda(\mathbf{w})/\|soft_\lambda(\mathbf{w})\|_2$, where $soft_\lambda(\mathbf{w})$ represents the soft thresholding function $\text{sign}(\mathbf{w})(|\mathbf{w}|-\lambda)_+$, where $(\mathbf{x})_+$ represents the vector attained by projecting $\mathbf{x}$ to its nonnegative orthant, and $(I_1, I_2, \ldots, I_d)$ denotes the permutation of $(1,2,\ldots,d)$ based on the ascending order of $|\mathbf{w}| = (|w_1|, |w_2|, \ldots, |w_d|)^T$. The proof of the theorem is provided in Appendix B.

Theorem 3. *The optimal solution of*

$$\max_{\mathbf{v}} \mathbf{w}^T\mathbf{v} \quad \text{s.t. } \mathbf{v}^T\mathbf{v} = 1, \;\; \|\mathbf{v}\|_1 \le t \tag{12}$$

is given by

$$\mathbf{v}_1^*(\mathbf{w},t) = \begin{cases} \phi, & t<1, \\ f_{\mathbf{w}}(\lambda_k), & \|f_{\mathbf{w}}(|w_{I_k}|)\|_1 \le t < \|f_{\mathbf{w}}(|w_{I_{k-1}}|)\|_1 \;(k=2,3,\ldots,d-1), \\ f_{\mathbf{w}}(\lambda_1), & \|f_{\mathbf{w}}(|w_{I_1}|)\|_1 \le t < \sqrt{d}, \\ f_{\mathbf{w}}(0), & t \ge \sqrt{d}, \end{cases}$$

where for $k=1,2,\ldots,d-1$,

$$\lambda_k = \frac{(m-t^2)(\sum_{i=1}^m a_i) - \sqrt{t^2(m-t^2)(m\sum_{i=1}^m a_i^2 - (\sum_{i=1}^m a_i)^2)}}{m(m-t^2)},$$

where $(a_1, a_2, \ldots, a_m) = (|w_{I_k}|, |w_{I_{k+1}}|, \ldots, |w_{I_d}|)$, $m=d-k+1$.

It should be noted that we have proved that $\|f_{\mathbf{w}}(|w_{I_{d-1}}|)\|_1 = 1$ and $\|f_{\mathbf{w}}(\lambda)\|_1$ is a monotonically decreasing function with respect to λ in Lemma 1 of Appendix B. This means that we can conduct the optimum $\mathbf{v}^*(\mathbf{w})$ of the optimization problem (7) for any $\mathbf{w}$ based on the above theorem.

The new algorithm, which we called BCD-SPCA, can then be easily constructed based on Theorems 1–3.

(*footnote continued*)
most of the existing sparse PCA methods [5–8], we do not enforce orthogonal PCs in the models.

2.4. The BCD-SPCA algorithm for sparse PCA

The main idea of the proposed BCD-SPCA method is to recursively optimize each column, $\mathbf{u}_i$ of $\mathbf{U}$ or $\mathbf{v}_i$ of $\mathbf{V}$ for $i=1,2,\ldots,r$, with other $\mathbf{u}_j$s and $\mathbf{v}_j$s $(j\neq i)$ fixed. The process is summarized as follows:

- Update each column $\mathbf{v}_i$ of $\mathbf{V}$ for $i=1,2,\ldots,r$ by the closed-form solution of Eq. (5) calculated by Theorem 2 (for $p=0$) or Theorem 3 (for $p=1$).
- Update each column $\mathbf{u}_i$ of $\mathbf{U}$ for $i=1,2,\ldots,r$ by the closed-form solution of Eq. (6) calculated by Theorem 1.

Through implementing the above procedures iteratively, $\mathbf{U}$ and $\mathbf{V}$ can be recursively updated until the stopping criterion is satisfied. We summarize the aforementioned procedure as Algorithm 1.

Algorithm 1. BCD algorithm for sparse PCA.

```
Input: Data matrix $\mathbf{X}\in R^{n\times d}$, number of sparse PCs $r$, sparsity
    parameters $\mathbf{t}=(t_1,\ldots,t_r)$.
1: Initialize $\mathbf{U}=(\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_r)\in R^{n\times r}$, $\mathbf{V}=(\mathbf{v}_1,\mathbf{u}_2,\ldots,\mathbf{v}_r)\in R^{d\times r}$.
2: repeat
3:   for $i=1,\ldots r$ do
4:     Compute $\mathbf{E}_i=\mathbf{X}-\sum_{j\neq i}\mathbf{u}_j\mathbf{v}_j^T$.
5:     Update $\mathbf{v}_i$ via solving Eq. (5) based on Theorem 2 (for
    $p=0$) or Theorem 3 (for $p=1$).
6:     Update $\mathbf{u}_i$ via solving Eq. (6) based on Theorem 1.
7:   end for
8: until stopping criterion satisfied.
Output: The sparse PC loading vectors $\mathbf{V}=(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_r)$.
```

We then briefly discuss how to specify the stopping criterion of the algorithm. The objective function of the sparse PCA model (4) is monotonically decreasing in the iterative process of Algorithm 1 since each of the step 5 and step 6 in the iterations makes an exact optimization for a column vector $\mathbf{u}_i$ of $\mathbf{U}$ or $\mathbf{v}_i$ of $\mathbf{V}$, with all of the others fixed. We can thus terminate the iterations of the algorithm when the updating rate of $\mathbf{U}$ or $\mathbf{V}$ is smaller than some preset threshold, or the maximum number of iterations is reached.

Now we briefly analyze the computational complexity of the proposed BCD-SPCA algorithm. It is evident that the computational complexity of Algorithm 1 is essentially determined by the iterations between step 5 and step 6, i.e. the calculation of the closed-form solutions of $\mathbf{v}_i$ and $\mathbf{u}_i$ of $\mathbf{V}$ and $\mathbf{U}$, respectively. To compute $\mathbf{u}_i$, only simple operations are involved and the computation needs $O(nd)$ cost. To compute $\mathbf{v}_i$, a sorting for the elements of the d-dimensional vector $|\mathbf{w}|=|\mathbf{E}^T\mathbf{u}|$ is required, and the total computational cost is around $O(nd\log d)$ by applying the well-known heap sorting algorithm [28]. The whole process of the algorithm thus requires around $O(Trnd\log d)$ computational cost in each iteration, where T is the preset maximal iteration number for the algorithm. That is, the computational complexity of the proposed algorithm is approximately linear in both the size and the dimensionality of input data.

2.5. Convergence analysis

In this section we evaluate the convergence of the proposed algorithm.

The convergence of our algorithm can actually be implied by the monotonic decrease of the cost function of (4) during the iterations of the algorithm. In specific, in each iteration of the algorithm, step 5 and step 6 optimize the column vector $\mathbf{u}_i$ of $\mathbf{U}$ or $\mathbf{v}_i$ of $\mathbf{V}$, with all of the others fixed, respectively. Since the objective function of Eq. (4) is evidently lower bounded (≥ 0), the algorithm is guaranteed to be convergent.

We want to go a further step to evaluate where the algorithm converges. Based on the formulation of the optimization problem (4), we can construct a specific function as follows:

$$f(\mathbf{u}_1,\ldots,\mathbf{u}_r,\mathbf{v}_1,\ldots,\mathbf{v}_r)=f_0(\mathbf{u}_1,\ldots,\mathbf{u}_r,\mathbf{v}_1,\ldots,\mathbf{v}_r)+\sum_{i=1}^{r}f_i(\mathbf{v}_i), \tag{13}$$

where

$$f_0(\mathbf{u}_1,\ldots,\mathbf{u}_r,\mathbf{v}_1,\ldots,\mathbf{v}_r)=\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2=\|\mathbf{X}-\textstyle\sum_{i=1}^r\mathbf{u}_i\mathbf{v}_i^T\|_F^2,$$

and for each of $i=1,\ldots,r$, $f_i(\mathbf{v}_i)$ is an indicator function defined as

$$f_i(\mathbf{v}_i)=\begin{cases}0 & \text{if } \|\mathbf{v}_i\|_p\leq t_i \text{ and } \mathbf{v}_i^T\mathbf{v}_i=1,\\ \infty & \text{otherwise}.\end{cases}$$

It is then easy to show that the constrained optimization problem (4) is equivalent to the unconstrained problem:

$$\min_{\{\mathbf{u}_i,\mathbf{v}_i\}_{i=1}^r} f(\mathbf{u}_1,\ldots,\mathbf{u}_r,\mathbf{v}_1,\ldots,\mathbf{v}_r). \tag{14}$$

The proposed algorithm can then be viewed as a BCD method for solving Eq. (14) [29], by alteratively optimizing $\mathbf{u}_i,\mathbf{v}_i$, $i=1,2,\ldots,r$, respectively. Then the following theorem implies that our algorithm can converge to a stationary point of the problem.

Theorem 4 (*Tseng [29]*). *Assume that the level set $X^0=\{x: f(x)\leq f(x^0)\}$ is compact and that f is continuous on X^0. If $f(\mathbf{u}_1,\ldots,\mathbf{u}_r,\mathbf{v}_1,\ldots,\mathbf{v}_r)$ is regular and has at most one minimum in each $\mathbf{u}_i$ and $\mathbf{v}_i$ with others fixed for $i=1,2,\ldots,r$, then the sequence $(\mathbf{u}_1,\ldots,\mathbf{u}_r,\mathbf{v}_1,\ldots,\mathbf{v}_r)$ generated by the BCD method converges to a stationary point of f.*

In the above theorem, the assumption that the function f, as defined in Eq. (14), is regular holds under the condition that $dom(f_0)$ is open and f_0 is Gateaux-differentiable on $dom(f_0)$ (Lemma 3.1 under Condition A1 in [29]). Based on Theorems 1–3, we can also easily see that $f(\mathbf{u}_1,\ldots,\mathbf{u}_r,\mathbf{v}_1,\ldots,\mathbf{v}_r)$ has a unique minimum in each $\mathbf{u}_i$ and $\mathbf{v}_i$ with others fixed. The above theorem then naturally follows from Theorem 4.1(c) in [29].

Another advantage of the proposed BCD methodology is that it can be easily extended to other sparse PCA applications when certain constraints are needed for output sparse PCs. In the following section we give one of the extensions of our methodology—nonnegative sparse PCA problem.

2.6. The BCD method for nonnegative sparse PCA

The nonnegative sparse PCA [30] problem differs from the conventional sparse PCA in its nonnegativity constraint imposed on the output sparse PCs. The nonnegativity property of this problem is especially important in some applications such as microeconomics, environmental science, and biology [31]. The corresponding optimization model is written as follows:

$$\min_{\mathbf{U},\mathbf{V}}\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2 \quad \text{s.t. } \mathbf{v}_i^T\mathbf{v}_i=1,\ \ \|\mathbf{v}_i\|_p\leq t_i,\ \ \mathbf{v}_i\succcurlyeq 0\ (i=1,2,\ldots,r), \tag{15}$$

where $\mathbf{v}_i\succcurlyeq 0$ means that each element of $\mathbf{v}_i$ is greater than or equal to 0.

By utilizing the similar BCD strategy, this problem can be separated into a series of small minimization problems, each with respect to a column vector $\mathbf{u}_i$ of $\mathbf{U}$ and $\mathbf{v}_i$ of $\mathbf{V}$ for $i=1,2,\ldots,r$, respectively, as follows:

$$\min_{\mathbf{v}_i}\|\mathbf{E}_i-\mathbf{u}_i\mathbf{v}_i^T\|_F^2 \quad \text{s.t. } \mathbf{v}_i^T\mathbf{v}_i=1,\ \ \|\mathbf{v}_i\|_p\leq t_i,\ \ \mathbf{v}_i\succcurlyeq 0 \tag{16}$$

and

$$\min_{\mathbf{u}_i} \|\mathbf{E}_i - \mathbf{u}_i \mathbf{v}_i^T\|_F^2, \tag{17}$$

where $p=0$ or 1. Since Eq. (17) is of the same formulation as Eq. (6), we only need to discuss how to solve Eq. (16). For the convenience of notation, we first rewrite Eq. (16) as

$$\min_{\mathbf{v}} \|\mathbf{E} - \mathbf{u}\mathbf{v}^T\|_F^2 \quad \text{s.t. } \mathbf{v}^T\mathbf{v} = 1, \ \ \|\mathbf{v}\|_p \le t, \ \ \mathbf{v} \geqslant 0. \tag{18}$$

The closed-form solution of (18) is given in the following theorem. The proof of this theorem is given in Appendix C.

Theorem 5. *The closed-form solution of Eq. (18) is* $\mathbf{v}_p^*((\mathbf{w})_+, t)$ *(*p*=0,1), where* $\mathbf{w} = \mathbf{E}^T\mathbf{u}$*, and* $\mathbf{v}_0^*(\cdot,\cdot)$ *and* $\mathbf{v}_1^*(\cdot,\cdot)$ *are defined in Theorems 2 and 3, respectively.*

By virtue of the closed-form solution of Eq. (18) given by Theorem 5, we can now construct the algorithm for solving nonnegative sparse PCA model (15), called BCD-NSPCA. Since the algorithm differs from Algorithm 1 only in step 5 (i.e. updating of $\mathbf{v}_i$), we only list this step in Algorithm 2.

Algorithm 2. BCD algorithm for nonnegative sparse PCA.

```
5:   Update $\mathbf{v}_i$ via solving Eq. (16) based on Theorem 5.
```

3. Experiments

To evaluate the performance of the proposed BCD-SPCA and BCD-NSPCA algorithms on the sparse PCA problem, we conduct experiments on a series of synthetic and real data sets. All the experiments were implemented on Matlab 7.11 (R2010b) platform in a PC with AMD Athlon (TM) 64 X2 Dual 5000+@2.60 GHz (CPU) and 2 GB (memory). In all experiments, the SVD method was utilized for initialization. The proposed algorithms under both $p=0$ and $p=1$ are denoted as $\text{BCD}-\text{SPCA}_{l_0}$ and $\text{BCD}-\text{SPCA}_{l_1}$ for sparse PCA, and $\text{BCD}-\text{NSPCA}_{l_0}$ and $\text{BCD}-\text{NSPCA}_{l_1}$ for nonnegative sparse PCA, respectively.

3.1. Synthetic simulations

Two synthetic data sets were utilized to evaluate the performance of the proposed algorithm on recovering the ground truth sparse principal components underlying data. The results are listed as follows:

3.1.1. Hastie data

Hastie data set was first proposed by Zou et al. [5] to illustrate the advantage of sparse PCA over conventional PCA on sparse PC extraction. So far this data set has become one of the most frequently utilized benchmark data for testing the effectiveness of sparse PCA methods. The data set was generated in the following way: first, three hidden factors V_1, V_2 and V_3 were created as

$$V_1 \sim \mathcal{N}(0, 290), \quad V_2 \sim \mathcal{N}(0, 300), \quad V_3 = 0.3V_1 + 0.925V_2 + \varepsilon,$$

where $\varepsilon \sim \mathcal{N}(0,1)$, and V_1, V_2 and ε are independent; afterwards, 10 observable variables were generated as

$$X_i = V_1 + \varepsilon_i^1, \quad i = 1,2,3,4,$$
$$X_i = V_2 + \varepsilon_i^2, \quad i = 5,6,7,8,$$
$$X_i = V_3 + \varepsilon_i^3, \quad i = 9,10,$$

where $\varepsilon_i^j \sim \mathcal{N}(0,1)$ and all ε_i^js are independent. The data so generated are of intrinsic sparse PCs [5]: the first recovers the factor V_2 only using (X_5, X_6, X_7, X_8), and the second recovers V_1 only utilizing (X_1, X_2, X_3, X_4).

We generated 100 sets of data, each contains 1000 data generated in the aforementioned way, and applied Algorithm 1 to them to extract the first two sparse PCs. The results show that our algorithm can perform well in all experiments. In specific, the proposed BCD-SPCA algorithm faithfully delivers the ground truth sparse PCs in all experiments. The effectiveness of the proposed algorithm is thus easily substantiated in this series of benchmark data.

3.1.2. Synthetic toy data

As [7,8], we adopted another interesting toy data to evaluate the performance of the proposed method. The data were generated from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with mean $\mathbf{0}$ and covariance $\mathbf{\Sigma} \in \mathbb{R}^{10\times 10}$, which was calculated by

$$\mathbf{\Sigma} = \sum_{j=1}^{10} c_j \mathbf{v}_j \mathbf{v}_j^T.$$

Here, $(c_1, c_2, \ldots, c_{10})$, the eigenvalues of the covariance matrix $\mathbf{\Sigma}$, were pre-specified as $(250, 240, 50, 50, 6, 5, 4, 3, 2, 1)$, respectively, and $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{10})$ are 10-dimensional orthogonal vectors, formulated by

$$\mathbf{v}_1 = (0.422, 0.422, 0.422, 0.422, 0, 0, 0, 0, 0.380, 0.380)^T,$$
$$\mathbf{v}_2 = (0, 0, 0, 0, 0.489, 0.489, 0.489, 0.489, -0.147, 0.147)^T,$$

and the rest being generated by applying the Gram–Schmidt orthonormalization to 8 randomly valued 10-dimensional vectors. It is easy to see that the data generated under this distribution are of first two sparse PC vectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

Four series of experiments, each involving 1000 sets of data generated from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, were utilized, with sample sizes 500, 1000, 2000, 5000, respectively. For each experiment, the first two PCs, $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$, were calculated by a sparse PCA method and then if both $|\hat{\mathbf{v}}_1^T \mathbf{v}_1| \ge 0.99$ and $|\hat{\mathbf{v}}_2^T \mathbf{v}_2| \ge 0.99$ were satisfied, the method was considered as a success. The proposed BCD-SPCA method, together with the conventional PCA and 12 current sparse PCA methods, including SPCA [5], DSPCA [6], PathSPCA [16], $\text{sPCA}-\text{rSVD}_{l_0}$, $\text{sPCA}-\text{rSVD}_{l_1}$, $\text{sPCA}-\text{rSVD}_{SCAD}$ [7], EMPCA [9], GPower_{l_0}, GPower_{l_1}, $\text{GPower}_{l_{0,m}}$, $\text{GPower}_{l_{1,m}}$ [8] and ALSPCA [15], have been implemented, and the success times for four series of experiments have been recorded and summarized, respectively. The results are listed in Table 2.

The advantage of the proposed $\text{BCD}-\text{SPCA}_{l_1}$ algorithm can be easily observed from Table 2. In specific, our method always attains the highest or second highest success times (in the size 1000 case, 1 less than ALSPCA) as compared with the other utilized methods in all of the four series of experiments. Considering that

Table 2
Comparison of success times of PCA and different sparse PCA methods in synthetic toy experiments with varying sample sizes. The best results are highlighted in bold.

Method	$n=500$	$n=1000$	$n=2000$	$n=5000$
PCA	0	0	0	0
SPCA	566	673	756	839
DSPCA	211	203	138	62
PathSPCA	189	187	186	171
$\text{sPCA}-\text{rSVD}_{l_0}$	646	702	797	906
$\text{sPCA}-\text{rSVD}_{l_1}$	649	715	806	909
$\text{sPCA}-\text{rSVD}_{\text{SCAD}}$	649	715	806	909
EMPCA	649	715	806	909
GPower_{l_0}	155	154	155	139
GPower_{l_1}	122	127	126	126
$\text{GPower}_{l_{0,m}}$	91	76	71	16
$\text{GPower}_{l_{1,m}}$	90	92	88	82
ALSPCA	669	**749**	826	927
$\text{BCD}-\text{SPCA}_{l_0}$	646	708	800	907
$\text{BCD}-\text{SPCA}_{l_1}$	**676**	748	**827**	**928**

the ALSPCA method, which is the only comparable method in these experiments, utilizes strict constraints on the orthogonality of output PCs while the BCD-SPCA method does not utilize any prior ground truth information of data, the capability of the proposed method on sparse PCA calculation can be more prominently verified.

3.2. *Experiments on real data*

In this section, we further evaluate the performance of the proposed BCD-SPCA method on three real data sets, including the pitprops, colon and Yale B face data. Two quantitative criteria were employed for performance assessment. They are designed in the viewpoints of reconstruction-error-minimization and data-variance-maximization, respectively, just corresponding to the original formulations (4) and (3) for sparse PCA problem.

- *Reconstruction-error-minimization criterion: RRE*. Once sparse PC loading matrix $\mathbf{V}$ is obtained by a method, the input data can then be reconstructed by $\hat{\mathbf{X}}=\hat{\mathbf{U}}\mathbf{V}^T$, where $\hat{\mathbf{U}}=\mathbf{X}\mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}$, obtained by the least square method. Then the relative reconstruction error (RRE) can be calculated by

$$\text{RRE}=\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F}{\|\mathbf{X}\|_F},$$

to assess the performance of the utilized method in data reconstruction point of view.
- *Data-variance-maximization criterion: PEV.* After obtaining the sparse PC loading matrix $\mathbf{V}$, the input data can then be reconstructed by $\hat{\mathbf{X}}=\mathbf{X}\mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$, as aforementioned. And thus the variance of the reconstructed data can be computed by $\text{Tr}((1/n)\hat{\mathbf{X}}^T\hat{\mathbf{X}})$. The percentage of explained variance (PEV [7]) of the reconstructed data from the original one can then be calculated by

$$\text{PEV}=\frac{\text{Tr}\left(\frac{1}{n}\hat{\mathbf{X}}^T\hat{\mathbf{X}}\right)}{\text{Tr}\left(\frac{1}{n}\mathbf{X}^T\mathbf{X}\right)}\times 100\%=\frac{\text{Tr}(\hat{\mathbf{X}}^T\hat{\mathbf{X}})}{\text{Tr}(\mathbf{X}^T\mathbf{X})}\times 100\%$$

to evaluate the performance of the utilized method in data variance point of view.

3.2.1. *Pitprops data*

The pitprops data set, consisting of 180 observations and 13 measured variables, was first introduced by Jeffers [32] to show the difficulty of interpreting PCs. This data set is one of the most commonly utilized examples for sparse PCA evaluation, and thus was also employed to testify the effectiveness of the proposed BCD-SPCA method. The comparison methods include SPCA [5], DSPCA [6], PathSPCA [16], $\text{sPCA}-\text{rSVD}_{l_0}$, $\text{sPCA}-\text{rSVD}_{l_1}$, $\text{sPCA}-\text{rSVD}_{SCAD}$ [7], EMPCA [9], GPower_{l_0}, GPower_{l_1}, $\text{GPower}_{l_0,m}$, $\text{GPower}_{l_1,m}$ [8] and ALSPCA [15]. For each utilized method, 6 sparse PCs were extracted from the pitprops data, with different cardinality settings: 8-5-6-2-3-2 (altogether 26 nonzero elements), 7-4-4-1-1-1 (altogether 18 nonzero elements, as set in [5]) and 7-2-3-1-1-1 (altogether 15 nonzero elements, as set in [6]), respectively. In each experiment, both the RRE and PEV values, as defined above, were calculated, and the results are summarized in Table 3. Fig. 1 further shows the RRE and PEV curves attained by different sparse PCA methods in all experiments for more illumination. It should be noted that the $\text{GPower}_{l_0,m}$, $\text{GPower}_{l_1,m}$ and ALSPCA methods employ the block methodology, as introduced in the introduction, and calculate all sparse PCs at once while cannot sequentially derive different numbers of sparse PCs with preset cardinality settings. Thus the results of these methods reported in Table 3 were calculated with the total sparse PC cardinalities being 26, 18 and 15, respectively, and are not included in Fig. 1.

Table 3
Performance comparison of different sparse PCA methods on pitprops data with different cardinality settings. The best result in each experiment is highlighted in bold.

Method	8-5-6-2-3-2		7-4-4-1-1-1		7-2-3-1-1-1	
	RRE	PEV (%)	RRE	PEV (%)	RRE	PEV (%)
SPCA	0.4162	82.68	0.4448	80.22	0.4459	80.11
DSPCA	0.4303	81.48	0.4563	79.18	0.4771	77.23
PathSPCA	0.4080	83.35	0.4660	80.11	0.4457	80.13
$\text{sPCA}-\text{rSVD}_{l_0}$	0.4139	82.87	0.4376	80.85	0.4701	77.90
$\text{sPCA}-\text{rSVD}_{l_1}$	0.4314	81.39	0.4427	80.40	0.4664	78.25
$\text{sPCA}-\text{rSVD}_{\text{SCAD}}$	0.4306	81.45	0.4453	80.17	0.4762	77.32
EMPCA	0.4070	83.44	0.4376	80.85	0.4451	80.18
GPower_{l_0}	0.4092	83.26	0.4400	80.64	0.4457	80.13
GPower_{l_1}	0.4080	83.35	0.4460	80.11	0.4457	80.13
$\text{GPower}_{l_{0,m}}$	0.4224	82.16	0.5089	74.10	0.4644	78.44
$\text{GPower}_{l_{1,m}}$	0.4187	82.46	0.4711	77.81	0.4589	78.94
ALSPCA	0.4168	82.63	0.4396	80.67	0.4537	79.42
$\text{BCD}-\text{SPCA}_{l_0}$	0.4115	83.07	0.4419	80.47	**0.4419**	**80.47**
$\text{BCD}-\text{SPCA}_{l_1}$	**0.4005**	**83.50**	**0.4343**	**81.14**	0.4420	80.46

It can be seen from Table 3 that under all cardinality settings of the first 6 PCs, the proposed BCD-SPCA method always achieves the lowest RRE and highest PEV values among all the competing methods. This means that the BCD-SPCA method is advantageous in both reconstruction-error-minimization and data-variance-maximization viewpoints. Furthermore, from Fig. 1, it is easy to see the superiority of the BCD-SPCA method. In specific, for different number of extracted sparse PC components, the proposed BCD-SPCA method can always get the smallest RRE values and the largest PEV values, as compared with the other utilized sparse PCA methods, in the experiments. This further substantiates the effectiveness of the proposed BCD-SPCA method in both reconstruction-error-minimization and data-variance-maximization viewpoints.

3.2.2. *Colon data*

The colon data set [33] consists of 62 tissue samples with the gene expression profiles of 2000 genes extracted from DNA microarray data. This is a typical data set with high-dimension and low-sample-size property, and is always employed by sparse methods for extracting interpretable information from high-dimensional genes. We thus adopted this data set for evaluation. In specific, 20 sparse PCs, each with 50 nonzero loadings, were calculated by different sparse PCA methods, including SPCA [5], PathSPCA [16], $\text{sPCA}-\text{rSVD}_{l_0}$, $\text{sPCA}-\text{rSVD}_{l_1}$, $\text{sPCA}-\text{rSVD}_{SCAD}$ [7], EMPCA [9], GPower_{l_0}, GPower_{l_1}, $\text{GPower}_{l_{0,m}}$, $\text{GPower}_{l_{1,m}}$ [8] and ALSPCA [15], respectively. Their performance is compared in Table 4 and Fig. 2 in terms of RRE and PEV, respectively. It should be noted that the DSPCA method has also been tried, while cannot be terminated in a reasonable time in this experiment, and thus we omit its result in the table. Besides, we have carefully tuned the parameters of the GPower methods (including GPower_{l_0}, GPower_{l_1}, $\text{GPower}_{l_{0,m}}$ and $\text{GPower}_{l_{1,m}}$), and can get 20 sparse PCs with total cardinality around 1000, similar as the total nonzero elements number of the other utilized sparse PCA methods, while cannot get sparse PC loading sequences each with cardinality 50 as expected. The results are thus not demonstrated in Fig. 2.

From Table 4, it is easy to see that $\text{BCD}-\text{SPCA}_{l_0}$ achieves the lowest RRE and highest PEV values, as compared with the other 11 employed sparse PCA methods. Fig. 2 further demonstrates that as the number of extracted sparse PCs increases, the advantage of the

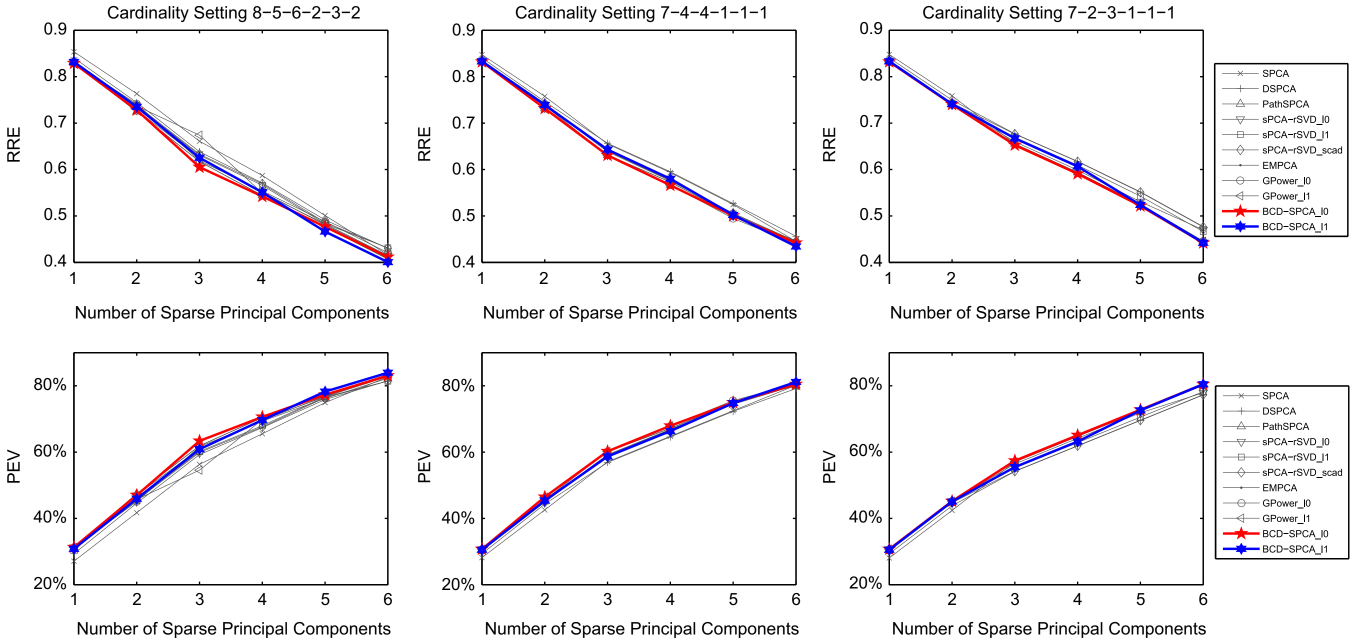


Fig. 1. The tendency curves of RRE and PEV with respect to the number of extracted sparse PCs obtained by different sparse PCA methods on pitprops data. Three cardinality settings for the extracted sparse PCs are utilized, including 8-5-6-2-3-2, 7-4-4-1-1-1 and 7-2-3-1-1-1.

Table 4
Performance comparison of different sparse PCA methods on colon data. The best results are highlighted in bold.

Method	SPCA	PathSPCA	sPCA − rSVD$_{l_0}$	sPCA − rSVD$_{l_1}$	sPCA − rSVD$_{\mathrm{SCAD}}$
RRE	0.7892	0.5287	0.5236	0.5628	0.5723
PEV	37.72%	72.05%	72.58%	68.32%	67.25%
Method	EMPCA	GPower$_{l_0}$	GPower$_{l_1}$	GPower$_{l_{0,m}}$	GPower$_{l_{1,m}}$
RRE	0.5211	0.5042	0.5076	0.4870	0.4904
PEV	72.84%	74.56%	74.23%	76.29%	75.95%
Method		ALSPCA		BCD − SPCA$_{l_0}$	BCD − SPCA$_{l_1}$
RRE		0.5917		**0.4737**	0.5536
PEV		64.99%		**77.56**%	69.35%

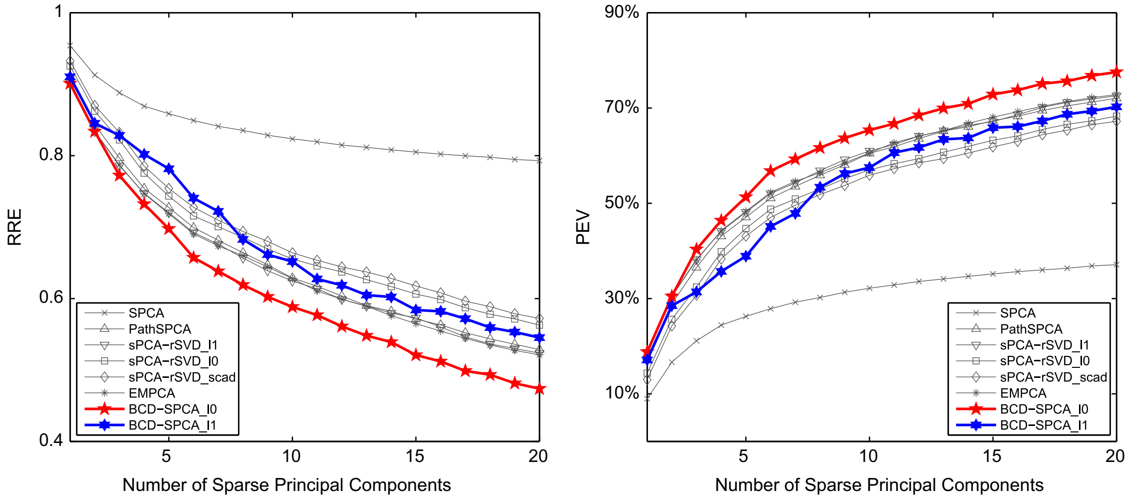


Fig. 2. The tendency curves of RRE and PEV with respect to the number of extracted sparse PCs, each with cardinality 50, attained by different sparse PCA methods on colon data.

Table 5
Performance comparison of different sparse PCA methods on Yale Face Database B. The best results are highlighted in bold.

Method	PathSPCA	$\mathrm{sPCA-rSVD}_{l_0}$	$\mathrm{sPCA-rSVD}_{l_1}$	$\mathrm{sPCA-rSVD}_{\mathrm{SCAD}}$
RRE.	0.2943	0.2895	0.3074	0.3159
PEV.	91.34%	91.62%	90.55%	90.02%
Method	EMPCA	$\mathrm{BCD-SPCA}_{l_0}$	$\mathrm{BCD-SPCA}_{l_1}$	
RRE.	0.2855	**0.2657**	0.2976	
PEV.	91.85%	**92.94**%	91.15%	

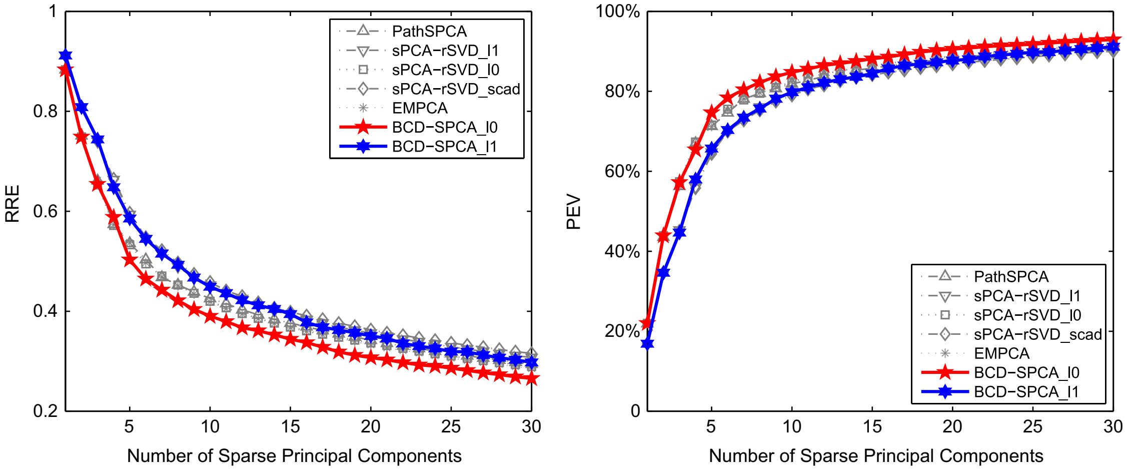


Fig. 3. The tendency curves of RRE and PEV with respect to the number of extracted sparse PCs, each with cardinality 2000, attained by different sparse PCA methods on Yale Face Database B.

proposed method tends to be more dominant than other methods, with respect to both the RRE and PEV criteria. This further substantiates the effectiveness of the proposed BCD strategy and implies its potential usefulness in applications with various interpretable components.

3.2.3. *Yale Face Database B*

We then test the performance of sparse PCA methods on Yale Face Database B [34]. A total of 20 face images for each of the first 10 subjects in this database were randomly chosen, resulting total 200 images. Then the images were cropped to 192 × 168 pixels as in [35], and further sub-sampled to 96 × 84 pixels, resulting in the data matrix of size 200 × 8064. A total of 30 sparse PCs, each with 2000 nonzero loadings, were calculated by different sparse PCA methods, including PathSPCA [16], $\mathrm{sPCA-rSVD}_{l_0}$, $\mathrm{sPCA-rSVD}_{l_1}$, $\mathrm{sPCA-rSVD}_{SCAD}$ [7] and EMPCA [9] (other methods either failed to produce the sparse PCs with preset cardinality or cannot be executed in reasonable time), respectively, together with the proposed method. Their performance is compared in Table 5 and Fig 3 in terms of both RRE and PEV, respectively. It can easily observed that $\mathrm{BCD-SPCA}_{l_0}$ always achieves the lowest RRE and highest PEV values, as compared with the other employed sparse PCA methods.

3.3. *Nonnegative sparse PCA experiments*

We further testify the performance of the proposed BCD-NSPCA method (Algorithm 2) in nonnegative sparse PC extraction. For comparison, two existing methods for nonnegative sparse PCA, NSPCA [30] and Nonnegative EMPCA (N-EMPCA, briefly) [9] were also employed.

Table 6
Performance comparison of success times attained by PCA, NSPCA, N-EMPCA, $\mathrm{BCD-NSPCA}_{l_0}$ and $\mathrm{BCD-NSPCA}_{l_1}$ on synthetic toy experiments with different sample sizes. The best results are highlighted in bold.

Method	$n=500$	$n=1000$	$n=2000$	$n=5000$
PCA	0	0	0	0
NSPCA	739	948	933	993
N-EMPCA	620	655	631	639
$\mathrm{BCD-NSPCA}_{l_0}$	834	948	939	996
$\mathrm{BCD-NSPCA}_{l_1}$	**835**	**949**	**978**	**1000**

3.3.1. *Synthetic toy data*

As the toy data utilized in Section 3.1.2, we also formulated a Gaussian distribution $\mathcal{N}(\mathbf{0},\mathbf{\Sigma})$ with mean $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma}=\sum_{j=1}^{10} c_j \mathbf{v}_j \mathbf{v}_j^T \in \mathbb{R}^{10\times 10}$. The leading two eigenvectors of $\mathbf{\Sigma}$ were specified as nonnegative and sparse vectors as

$$\mathbf{v}_1=(0.474,0,0.158,0,0.316,0,0.791,0,0.158,0)^T,$$

$$\mathbf{v}_2=(0,0.140,0,0.840,0,0.280,0,0.140,0,0.420)^T,$$

and the rest were then generated by applying the Gram–Schmidt orthonormalization to 8 randomly valued 10-dimensional vectors. The 10 corresponding eigenvalues $(c_1,c_2,\ldots,c_{10})$ were preset as $(210,190,50,50,6,5,4,3,2,1)$, respectively. Four series of experiments were designed, each with 1000 data sets generated from $\mathcal{N}(\mathbf{0},\mathbf{\Sigma})$, with sample sizes 500, 1000, 2000 and 5000, respectively. For each experiment, the first two PCs were calculated by the conventional PCA, NSPCA, N-EMPCA and BCD-NSPCA methods, respectively. The success times, calculated in the similar way as introduced in Section 3.1.2, of each utilized method on each series of experiments were recorded, as listed in Table 6.

From Table 6, it is seen that the proposed BCD-NSPCA methods achieve the highest success rates in all experiments, and its advantage on nonnegative sparse PCA calculation, as compared with the other utilized methods, can thus been verified in these experiments.

3.3.2. Colon data

The colon data set was utilized again for nonnegative sparse PCA calculation. The NSPCA and N-EMPCA methods were adopted as the competing methods. Since the NSPCA method cannot directly pre-specify the cardinalities of the extracted sparse PCs, we thus first applied NSPCA on the colon data (with parameters $\alpha = 1 \times 10^6$ and $\beta = 1 \times 10^7$) and then used the cardinalities of the nonnegative sparse PCs obtained by this method to preset the N-EMPCA and BCD-NSPCA methods for fair comparison. A total of 20 sparse PCs were computed by the three methods, and the performance was compared in Table 7 and Fig. 4, in terms of RRE and PEV, respectively.

Just as expected, it is evident that the proposed BCD-NSPCA methods dominate in both RRE and PEV viewpoints. From Table 7, we can observe that our method achieves the lowest RRE and highest PEV on 20 extracted nonnegative sparse PCs than the other two utilized methods. Furthermore, Fig. 4 shows that our method is advantageous, as compared with the other methods, for any preset number of extracted sparse PCs, and this advantage tends to be more significant as more sparse PCs are to be calculated. The effectiveness of the proposed method on nonnegative sparse PCA calculation can thus be verified.

Table 7
Performance comparison of different nonnegative sparse PCA methods on colon data. The best results are highlighted in bold.

Method	NSPCA	N-EMPCA	$\text{BCD}-\text{NSPCA}_{l_0}$	$\text{BCD}-\text{NSPCA}_{l_1}$
RRE	0.3674	0.3399	**0.2706**	0.2864
PEV	86.50%	88.45%	**92.68**%	91.80%

3.3.3. Application to face recognition

In this section, we introduce the performance of our method in the face recognition problem [30]. The proposed BCD-NSPCA method, together with the conventional PCA, NSPCA and N-EMPCA methods, has been applied to this problem and its performance is compared in this application. Since the l_0 version of BCD-NSPCA always outperforms the l_1 version, we report the results of $\text{BCD}-\text{NSPCA}_{l_0}$. The employed data set is the MIT CBCL Face Dataset #1, downloaded from "http://cbcl.mit.edu/software-datasets/FaceData2.html". This data set consists of 2429 aligned face images and 4548 non-face images, each with resolution 19 × 19. For each of the four utilized methods, 10 PC loading vectors were computed on face images, as shown in Fig. 5, respectively. For easy comparison, we also list the RRE and PEV values of three nonnegative sparse PCA methods in Table 8.

As depicted in Fig. 5, the nonnegative sparse PCs obtained by our BCD-NSPCA method more clearly exhibit the interpretable

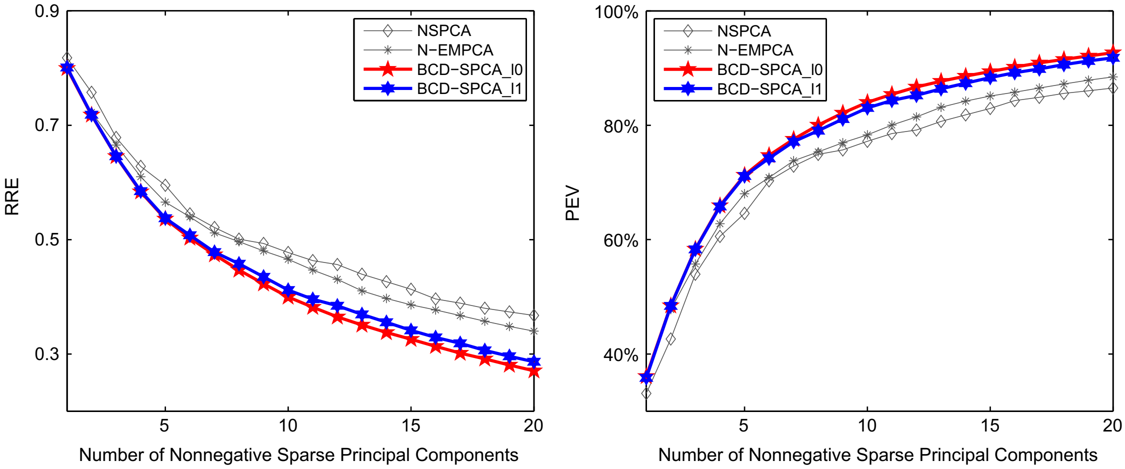


Fig. 4. The tendency curves of RRE and PEV, with respect to the number of extracted nonnegative sparse PCs, obtained by NSPCA, N-EMPCA, $\text{BCD}-\text{NSPCA}_{l_0}$ and $\text{BCD}-\text{NSPCA}_{l_1}$ on colon data.

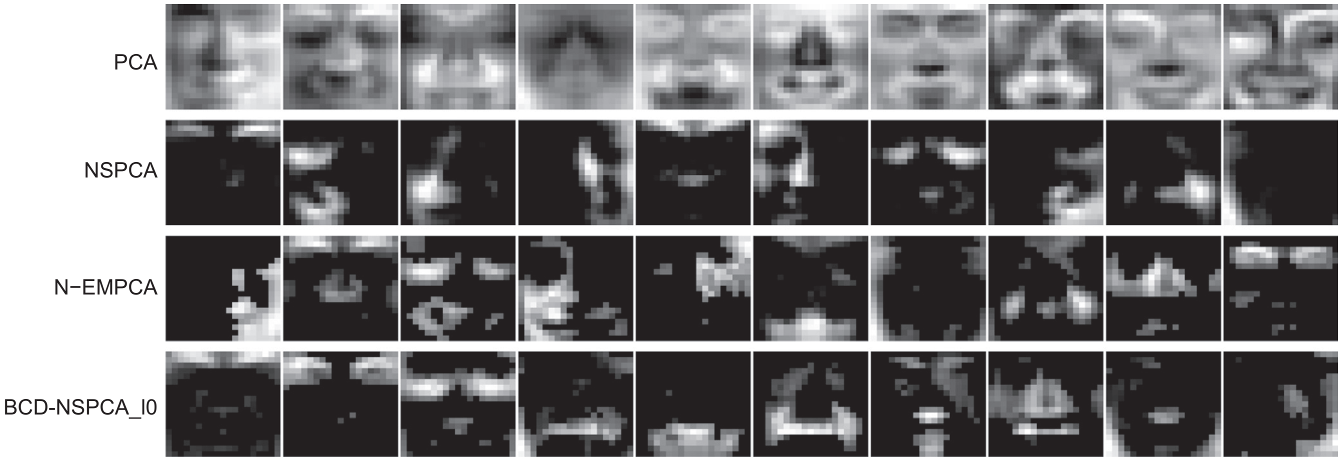


Fig. 5. From top row to bottom row: 10 PCs extracted by PCA, NSPCA, N-EMPCA and $\text{BCD}-\text{NSPCA}_{l_0}$, respectively, on MIT CBCL Face Dataset.

Table 8
Performance comparison of different nonnegative sparse PCA methods on MIT CBCL Face Dataset #1. The best results are highlighted in bold.

Method	NSPCA	N-EMPCA	$\mathrm{BCD-NSPCA}_{l_0}$
RRE	0.6993	0.6912	**0.6606**
PEV	51.10%	52.22%	**56.36**%

Table 9
Performance comparison of the classification accuracy obtained by different nonnegative sparse PCA methods. The best results are highlighted in bold.

Method	Face (%)	Non-face (%)	Total (%)
LR	96.71	93.57	94.47
PCA+LR	96.64	94.17	94.88
NSPCA+LR	94.89	93.49	93.89
N-EMPCA+LR	96.71	94.39	95.06
$\mathrm{BCD-NSPCA}_{l_0}$ +LR	**96.78**	**94.46**	**95.84**

features underlying faces, as compared with the other utilized methods, e.g. the first five PCs calculated from our method clearly demonstrate the eyebrows, eyes, cheeks, mouth and chin of faces, respectively. The advantage of the proposed method can further be verified quantitatively by its smallest RRE and largest PEV values, among all employed methods, in the experiment, as shown in Table 8. The effectiveness of our method can thus be substantiated.

To further show the usefulness of the proposed method, we applied it to face classification under this data set as follows. First we randomly chose 1000 face images and 1000 non-face images from MIT CBCL Face Dataset #1, and took them as the training data and the rest images as testing data. We then extracted 10 PCs by utilizing the PCA, NSPCA, N-EMPCA and $\mathrm{BCD-NSPCA}_{l_0}$ methods to the training set, respectively. By projecting the training data onto the corresponding 10 PCs obtained by each of these four methods and then fitting the linear logistic regression (LR) [36] model on these dimension-reduced data (10-dimensional), we can get a classifier for testing. The classification accuracy of the classifier so obtained on the testing data was then computed, and the results are reported in Table 9. In the table, the classification accuracy obtained by directly fitting the LR model on the original training data and testing on the original testing data is also listed for easy comparison.

From Table 9, it is clear that the proposed BCD strategy gets the best performance among all implemented methods, most accurately recognizing both the face images and the non-face images from the testing data. This further implies the potential usefulness of the proposed method in real applications.

4. Conclusion

In this paper we have proposed an effective block coordinate descent (BCD) method for sparse PCA problem. The basic idea is to decompose the original large sparse PCA problem into a series of small sub-problems and then recursively solve them. Although the BCD methodology is very simple, it performs surprisingly well in our experiments as compared to the current sparse PCA methods in terms of both the reconstruction-error-minimization and data-variance-maximization criteria. We have also shown that the new method converges to a stationary point of the problem, and can be easily extended to other sparse PCA problems with certain constraints, such as nonnegative sparse PCA problem.

There are many interesting investigations still worthy to be further explored. For example, when we reformat the square L_2-norm error of the sparse PCA model as the L_1-norm one, the robustness of the model can always be improved for heavy noise or outlier cases, while the model is correspondingly more difficult to solve. By adopting the similar BCD methodology, however, the problem can be decomposed into a series of much simpler sub-problems, which are expected to be much more easily solved than the original model. Besides, although we have proved the convergence of the proposed method, we do not know how far the result is from the global optimum of the problem. Stochastic global optimization techniques, such as simulated annealing and evolution computation methods, may be combined with the proposed method to further improve its performance. Also, the intrinsic relationships between other newly boosting dimensionality reduction techniques and sparse PCA research will also be considered in our future investigation.

Acknowledgments

This work was partially supported by the National Basic Research Program of China (973 Program) with No. 2013CB329404, the NSFC Projects with Nos. 61373114, 11131006, 91330204, and 61102131, and the Natural Science Foundation of Chongqing Science and Technology Commission Project with No. cstc2014jcyjA40048. Qian Zhao was partially supported by China Scholarship Council.

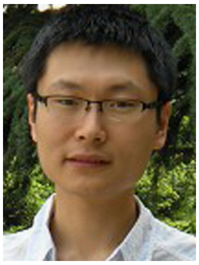
Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.neucom.2014.11.038.

References

[1] I.T. Jolliffe, Principal Component Analysis, 2nd ed., Springer, New York, 2002.
[2] I.T. Jolliffe, Rotation of principal components—choice of normalization constraints, J. Appl. Stat. 22 (1) (1995) 29–35.
[3] J. Cadima, I.T. Jolliffe, Loadings and correlations in the interpretation of principal components, J. Appl. Stat. 22 (2) (1995) 203–214.
[4] I.T. Jolliffe, N.T. Trendafilov, M. Uddin, A modified principal component technique based on the lasso, Journal of Computational and Graphical Statistics 12 (3) (2003) 531–547.
[5] H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, J. Comput. Gr. Stat. 15 (2) (2006) 265–286.
[6] A. d'Aspremont, L. El Ghaoui, M.I. Jordan, G. Lanckriet, A direct formulation for sparse PCA using semidefinite programming, SIAM Rev. 49 (3) (2007) 434–448.
[7] H.P. Shen, J. Huang, Sparse principal component analysis via regularized low rank matrix approximation, J. Multivar. Anal. 99 (6) (2008) 1015–1034.
[8] M. Journée, Y. Nesterov, P. Richtarik, R. Sepulchre, Generalized power method for sparse principal component analysis, J. Mach. Learn. Res. 11 (2010) 517–553.
[9] C. Sigg, J. Buhmann, Expectation-maximization for sparse and non-negative PCA, in: Proceedings of the 25th International Conference on Machine Learning, ACM, New York, NY, USA, 2008, pp. 960–967.
[10] Y. Guan, J. Dy, Sparse probabilistic principal component analysis, in: Proceedings of 12th International Conference on Artificial Intelligence and Statistics, 2009, pp. 185–192.
[11] K. Sharp, M. Rattray, Dense message passing for sparse principal component analysis, in: Proceedings of 13th International Conference on Artificial Intelligence and Statistics, 2010, pp. 725–732.
[12] C. Archambeau, F. Bach, Sparse probabilistic projections, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems, vol. 21, MIT Press, Cambridge, MA, 2009, pp. 73–80.
[13] B. Sriperumbudur, D. Torres, G. Lanckriet, Sparse eigen methods by dc programming, in: Proceedings of the 24th International Conference on Machine Learning, ACM, New York, NY, USA, 2007, pp. 831–838.
[14] B.K. Sriperumbudur, D.A. Torres, G. Lanckriet, A majorization–minimization approach to the sparse generalized eigenvalue problem, Mach. Learn. 85 (1–2) (2011) 3–39.
[15] Z. Lu, Y. Zhang, An augmented lagrangian approach for sparse principal component analysis, Math. Program. 135 (1–2) (2012) 149–193.
[16] A. d'Aspremont, F. Bach, L. Ghaoui, Full regularization path for sparse principal component analysis, in: Proceedings of the 24th International Conference on Machine Learning, ACM, New York, NY, USA, 2007, pp. 177–184.
[17] B. Moghaddam, Y. Weiss, S. Avidan, Spectral bounds for sparse PCA: exact and greedy algorithms, in: Y. Weiss, B. Schölkopf, J. Platt (Eds.), Advances in Neural

Information Processing Systems, vol. 18, MIT Press, Cambridge, MA, 2006, pp. 915–922.
[18] A. d'Aspremont, F. Bach, L. El Ghaoui, Optimal solutions for sparse principal component analysis, J. Mach. Learn. Res. 9 (2008) 1269–1294.
[19] D.M. Witten, R. Tibshirani, T. Hastie, A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, Biostatistics 10 (3) (2009) 515–534.
[20] A. Farcomeni, An exact approach to sparse principal component analysis, Comput. Stat. 24 (4) (2009) 583–604.
[21] Y. Zhang, L.E. Ghaoui, Large-scale sparse principal component analysis with application to text data, in: J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, vol. 24, MIT Press, Cambridge, MA, 2011, pp. 532–539.
[22] D.Y. Meng, Q. Zhao, Z.B. Xu, Improve robustness of sparse PCA by l_1-norm maximization, Pattern Recognit. 45 (1) (2012) 487–497.
[23] Y. Wang, Q. Wu, Sparse PCA by iterative elimination algorithm, Adv. Comput. Math. 36 (1) (2012) 137–151.
[24] D. Meng, H. Cui, Z. Xu, K. Jing, Following the entire solution path of sparse principal component analysis by coordinate-pairwise algorithm, Data Knowl. Eng. 88 (2013) 25–36.
[25] L. Mackey, Deflation methods for sparse PCA, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems, vol. 21, MIT Press, Cambridge, MA, 2009, pp. 1017–1024.
[26] H. Hotelling, Analysis of a complex of statistical variables into principal components, J. Educ. Psychol. 24 (1933) 417–441.
[27] K. Pearson, On lines and planes of closest fit to systems of points in space, Philos. Mag. 2 (7–12) (1901) 559–572.
[28] D. Knuth, The Art of Computer Programming, Addison-Wesley, Reading, MA, 1973.
[29] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, J. Optim. Theory Appl. 109 (3) (2001) 475–494.
[30] R. Zass, A. Shashua, Nonnegative sparse PCA, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems MIT Press, Cambridge, MA, 2007, pp. 1561–1568.
[31] A. Cichocki, R. Zdunek, A. Phan, S. Amari, Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation, John Wiley & Sons, West Sussex, United Kingdom, 2009.
[32] J. Jeffers, Two case studies in the application of principal component analysis, Appl. Stat. 16 (1967) 225–236.
[33] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, A. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Cell Biol. 96 (12) (1999) 6745–6750.
[34] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.
[35] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 684–698.
[36] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, Springer, New York, 2001.



Qian Zhao received the B.Sc. in 2009 from Xi'an Jiaotong University, Xi'an, China, where he is currently working toward the Ph.D. degree. His current research interests include low-rank matrix factorization, dimensionality reduction, and Bayesian method for machine learning.

Deyu Meng received the B.Sc., M.Sc., and Ph.D. degrees in 2001, 2004, and 2008, respectively, from Xi'an Jiaotong University, Xi'an, China. He is currently an Associate Professor with the Institute for Information and System Sciences, School of Mathematics and Statistics, Xi'an Jiaotong University. His current research interests include principal component analysis, nonlinear dimensionality reduction, feature extraction and selection, compressed sensing, and sparse machine learning methods.

Zongben Xu received the Ph.D. degree in mathematics from Xi'an Jiaotong University, Xi'an, China, in 1987. He currently serves as a Vice President with Xi'an Jiaotong University, the Academician of the Chinese Academy of Sciences, the Chief Scientist of the National Basic Research Program of China (973 Project), and the Director of the Institute for Information and System Sciences of the University. His current research interests include nonlinear functional analysis and intelligent information processing. Dr. Xu was a recipient of the National Natural Science Award of China in 2007 and was a winner of the CSIAM Su Buchin Applied Mathematics Prize in 2008. He delivered a talk at the International Congress of Mathematicians 2010.

Chenqiang Gao received the B.Sc. degree in computer science from the China University of Geosciences, Wuhan, China, in 2004 and the Ph.D. degree in pattern recognition and intelligence systems from the Huazhong University of Science and Technology, Wuhan, China, in 2009. He is currently an Associate Professor with Chongqing University of Posts and Telecommunications, Chongqing, China. His current research interests include image processing, infrared target detection, and event detection.