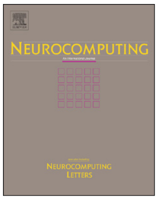
Neurocomputing 153 (2015) 180–190

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 $\text{GPower}_{l_0}$,

[*] 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 | Efficient to simultaneously attain large number of sparse PCs |
| | The sparse PCA calculation can be easily implemented under different sparsity parameter settings (i.e., $t_i$ in Eq. (3) and (4)) | 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