Following the Entire Solution Path of Sparse Principal Component Analysis by Coordinate-Pairwise Algorithm

Deyu Meng, Hengbin Cui, Zongben Xu, Kaili Jing

Institute for Information and System Sciences and Ministry of Education Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, P.R. China

Abstract

In this paper we derive an algorithm to follow the entire solution path of the sparse principal component analysis (PCA) problem. The core idea is to iteratively identify the pairwise variables along which the objective function of the sparse PCA model can be largest increased, and then incrementally update the coefficients of the two variables so selected by a small stepsize. The new algorithm dominates on its capability of providing a computational shortcut to attain the entire spectrum of solutions of the sparse PCA problem, which is always beneficial to real applications. The proposed algorithm is simple and easy to be implemented. The effectiveness of our algorithm is empirically verified by a series of experiments implemented on synthetic and real problems, as compared with other typical sparse PCA methods.

Keywords: Sparse principal component analysis, entire solution path, forward stagewise strategy, mining methods and algorithms.

[*]Tel: 86 130 3290 4180; Fax: 86 29 8266 8559.
Email address: dymeng@mail.xjtu.edu.cn. (Deyu Meng)

1. Introduction

The principal component analysis (PCA) is one of the most classical and popular techniques for data processing and dimensionality reduction, and has a wide range of applications throughout science and engineering [1, 2, 3]. In essence, PCA aims to find the orthogonal directions along which the variance of the input data can be maximally preserved. Such directions correspond to the so called principle components (PCs). Denote the data matrix as $X \in R^{d\times n}$, where d and n are the number of variables (dimensionality) and the number of observations (size), respectively. The first principal component (PC) of the data X is the solution to the following optimization model:

$$[L_{pca}] : \quad \begin{array}{c} \mathbf{w}_{pca} = \underset{\mathbf{w}}{\operatorname{argmax}}\, V(\mathbf{w}) = \mathbf{w}^T X X^T \mathbf{w} \\ s.t.\ \mathbf{w}^T\mathbf{w} \leq 1. \end{array} \qquad (1)$$

The second PC can be successively attained by solving (1), under the constraint that it is orthogonal to the first, and so on.

Recently, the research on the sparse PCA problem has attracted much attention [4]-[13]. The motivation of sparse PCA is to facilitate the interpretation of dimensionality reduction by involving fewer non-zero elements of the variables in the derived PCs. This series of research is especially meritorious in the area where the original variables are of significant physical meanings. Currently, sparse PCA has been successfully applied to many applications such as object recognition [16], biological gene analysis [10], and financial asset trading [5].

The sparse PCA model can be directly formulated by supplementing the l_0 constraint to the traditional PCA model, L_{pca}, to enforce sparsity of the derived PCs. The corresponding optimization model is:

$$[L_0(k)] : \quad \begin{array}{c} \mathbf{w}_{l_0}(k) = \underset{\mathbf{w}}{\operatorname{argmax}}\, V(\mathbf{w}) = \mathbf{w}^T X X^T \mathbf{w} \\ s.t.\ \mathbf{w}^T\mathbf{w} \leq 1 \\ \|\mathbf{w}\|_0 \leq k. \end{array} \qquad (2)$$

The above $L_0(k)$ optimization is a hard combinatorial problem and very difficult to be exactly solved, especially for high dimensional data. Currently, by virtue of the slacking, thresholding, and greedy techniques, several methods, including DSPCA [5], PathSPCA [6], DCPCA [7], GPower$_{l_0}$, GPower$_{l_0,m}$ [8], and etc., have been developed to approximate the solution to $L_0(k)$ or its extensions.

As compared with the $L_0(k)$ model, another model for sparse PCA is more generally employed by relaxing the non-convex l_0 constraint to a weaker but convex l_1 constraint, as expressed in the following:

$$[L_1(t)]: \quad \begin{array}{c} \mathbf{w}_{l_1}(t) = \underset{\mathbf{w}}{\operatorname{argmax}}\ V(\mathbf{w}) = \mathbf{w}^T X X^T \mathbf{w} \\ s.t.\ \mathbf{w}^T\mathbf{w} \leq 1 \\ \|\mathbf{w}\|_1 \leq t. \end{array} \tag{3}$$

The typical methods constructed on this model or its related extensions include SCoTLASS [9], SPCA [4], GPower$_{l_1}$, GPower$_{l_1,m}$ [8], EMPCA [10], ALSPCA [11], PMD [13], sPCA-rSVD [14], RSPCA [15], and etc..

Despite these developments, an important problem is still often encountered in real applications of sparse PCA: how to select an appropriate parameter k/t of the l_0/l_1 constraint for the $L_0(k)/L_1(t)$ model based on the given data. In practice, users often use some default value for the parameter, or retrain the model multiple times under different parameter settings and then figure out a good choice of k or t from them [4]. This, however, is actually a very difficult task, since on one hand, multiple training for a sparse PCA method is always very time consuming, and on the other hand, there is no specific criterion, like the predictive performance for the classification or regression problem, to judge whether a sparse PC vector is "good" for the unsupervised sparse PCA problem. A very useful methodology against this challenge is to derive the entire solution path of the sparse PCA model, i.e., the set of solutions for all meaningful values of the tuning parameter. The solution path so derived not only is capable of offering great convenience on proper selection of optimal tuning parameter against specific application of sparse PCA, but also giving an insightful spectrum to reflect the intrinsic mechanism underlying the sparse PCA model. Along this line, multiple efficient path-following algorithms have been designed for a family of well known machine learning and pattern recognition problems. They include the LARS for lasso [17], the SVMPath for L_1 and L_2 constraint SVMs [18, 19], the GLM path algorithm for generalized linear models [20], the path algorithm for multiple kernel learning [21], and etc..

In this paper we consider the extension of such path-following technique to the sparse PCA problem. Inspired by the forward stagewise regression method (FS$_\varepsilon$, [22, 23]) designed for lasso, the core idea of the proposed method is to repeatedly identify the pairwise variables along which the objective $V(\mathbf{w})$ of the sparse PCA model can be increased at most, and then incrementally update the coefficients of the two variables by a small stepsize. The new method capitalizes on its capability of creating a coefficient profile to fit the entire solution path of the sparse PCA problem, which is always beneficial to real applications.

In Section 2 the core idea of our method and its implemented details are introduced. In Section 3 a series of experimental results are presented to substantiate the effectiveness of the proposed method, as compared with the existing techniques. We finish with conclusion in Sections 4.

2. The coordinate-pairwise algorithm for sparse PCA

Denote the input data matrix as $X = [\mathbf{x}_1, \cdots, \mathbf{x}_d]^T \in R^{d\times n}$, where d and n are the numbers of the variables and the observations, respectively, and $\mathbf{x}_i \in R^n$ corresponds to the coefficients of the i-th variable. Throughout the paper, we denote matrices, vectors, and scalars by upper-case letters, lower case bold-faced letters, and lower-case non-bold-faced letters, respectively.

2.1. Reformulation of the $L_1(t)$ model

The proposed path-following algorithm is constructed on an equivalent reformulation of the $L_1(t)$ model, as expressed in the following:

$$[L_{2,c}(s)]: \quad \begin{array}{c} \mathbf{w}_{l_2,c}(s) = \underset{\mathbf{w}}{\operatorname{argmax}}\ V(\mathbf{w}) = \mathbf{w}^T X X^T \mathbf{w} \\ s.t.\ \mathbf{w}^T\mathbf{w} \le s \\ \|\mathbf{w}\|_1 \le c, \end{array} \tag{4}$$

where c is a pre-specified constant.

Although the models $L_{2,c}(s)$ and $L_1(t)$ look somewhat alike, they are of significant difference in their intrinsic mechanisms of implementing sparse PC calculation. In specific, $L_1(t)$ attains the PC vector with different cardinality through fixing the l_2 constraint $\mathbf{w}^T\mathbf{w} \le 1$, while varying the l_1 constraint $\|\mathbf{w}\|_1 \le t$ with respect to t. Contrarily, $L_{2,c}(s)$ realizes this aim by fixing the l_1 constraint $\|\mathbf{w}\|_1 \le c$, while changing the l_2 constraint $\mathbf{w}^T\mathbf{w} \le s$ with respect to s. Our motivation for this reformulation can be very intuitively understood by virtue of Figure 1. For the $L_1(t)$ model, the optimal solution $\mathbf{w}_{l_1}(t)$ with respect to t tends to be shifted along the vertex of the constraint area of $L_1(t)$, i.e., along the ***nonlinear*** sub-manifold of $\mathbf{w}^T\mathbf{w} = 1$ (illustrated as the sub-circle in the left panel of Figure 1). While for the reformulated $L_{2,c}(s)$ model, the corresponding optimal $\mathbf{w}_{l_2}(c, s)$ with respect to s inclines

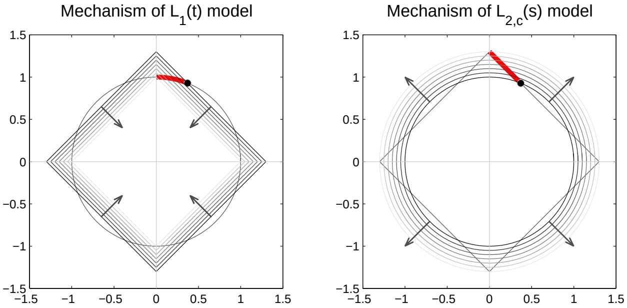

Figure 1: Graphical presentation for the implementation mechanisms of the $L_1(t)$ (left panel) and $L_{2,c}(s)$ (right panel) models. $L_1(t)$ attains different sparse PCs by setting the l_2 constraint $\mathbf{w}^T\mathbf{w} \leq 1$ (the circular area) fixed while the l_1 constraint $\|\mathbf{w}\|_1 \leq t$ (the diamond area) altered; Yet $L_{2,c}(s)$ realizes this aim by fixing the l_1 constraint $\|\mathbf{w}\|_1 \leq c$ while varying the l_2 constraint $\mathbf{w}^T\mathbf{w} \leq s$. The solution paths of the two models tend to be moved along the red curves as depicted in the two panels, respectively (started from the same initial point, depicted as the circles in the figure).

to move along the ***linear*** surface of $\|\mathbf{w}\|_1 = c$ (depicted as the line segment in the right panel of Figure 1). It is intuitively clear that the solution path of the new model with respect to s tends to be more easily followed than that of the $L_1(t)$ model with respect to t. Based on this direct comprehension, we expected to develop an effective and simple strategy to generate the entire solution path of sparse PCA by virtue of the $L_{2,c}(s)$ model.

A natural question is what the relationship between $L_1(t)$ and its reformulation $L_{2,c}(s)$ is. The following theorem clarifies the intrinsic equivalence between the two models.

Theorem 1. *For the optimal solutions* $\mathbf{w}_{l_1}(t)$ *and* $\mathbf{w}_{l_{2,c}}(s)$ *of* $L_1(t)$ *and* $L_{2,c}(s)$ *models, respectively, it holds that* $\mathbf{w}_{l_1}(t) = \frac{t}{c}\mathbf{w}_{l_{2,c}}(\frac{c^2}{t^2})$ *and* $\mathbf{w}_{l_{2,c}}(s) = \sqrt{s}\mathbf{w}_{l_1}(\frac{c}{\sqrt{s}})$.

The proof of Theorem 1 is given in the Appendix. This theorem implies that a comprehensive solution path of $L_1(t)$ with respect to t can be equivalently achieved by searching the entire solution path of $L_{2,c}(s)$ model with respect to s. This constitutes the fundament of the to-be-constructed path-following algorithm for sparse PCA.

2.2. The core idea of our method

Inspired by the forward stagewise regression strategy (FS$_\varepsilon$, [23]) proposed for lasso, we aim to build up the entire solution path for sparse PCA by iteratively generating the solution of $L_{2,c}(s+\varepsilon)$ from that of $L_{2,c}(s)$ in successive small step ε. In specific, there are two steps involved in each iteration of the proposed method: (1) selecting the pairwise coordinates/variables along which the objective function of the $L_{2,c}(s)$ model tends to be maximally increased; and (2) updating the pairwise coordinates so selected in a small step with the other variables fixed. Correspondingly, two key problems are required to be resolved: (i) how to find the proper pairwise coordinates to be updated in each iteration; (ii) how to build up easy computation to increment the pairwise coordinates so selected.

We first consider the aforementioned problem (ii). Denote the solution of $L_{2,c}(s)$ as $\mathbf{w}^o = (w_1^o, w_2^o, \cdots, w_d^o)^T$, and suppose that the i-th and j-th coordinates (w_i^o, w_j^o) of $\mathbf{w}^o$ are selected to be updated. Our aim is to formulate simple computation to incrementally update them in the feasible region of $L_{2,c}(s+\varepsilon)$ such that the objective function $\mathbf{V}(\mathbf{w})$ can be increased at most. To this aim, we should first find the direction v along which $\mathbf{V}(\mathbf{w}_0)$ tends to be largest increased in the feasible region of $L_{2,c}(s+\varepsilon)$, and then optimally move the i-th and j-th coefficients of $\mathbf{w}^o$ along this direction to approach the solution $\mathbf{w}^*$ of $L_{2,c}(s+\varepsilon)$.

When we fix the coefficients of the variables of $\mathbf{w}^o$ except the i-th and j-th ones, the cost function $V(\mathbf{w})$ can be reexpressed as:

$$\mathbf{V}(\mathbf{w}){=}V(w_i, w_j) + c_0,$$

where $V(w_i, w_j)$ corresponds to the portion of $\mathbf{V}(\mathbf{w})$ with respect to w_i and w_j, and c_0 is a constant independent of w_i and w_j. Then the model $L_{2,c}(s+\varepsilon)$ with respect to the pairwise variables w_i and w_j is transformed into the following form:

$$[L_{2,c}^{(i,j)}(s+\varepsilon)]: \quad \begin{array}{c} \max\limits_{w_i, w_j} V(w_i, w_j) \\ s.t.\ w_i^2 + w_j^2 \le s - \sum\limits_{k \ne i,j} (w_k^o)^2 + \varepsilon \\ |w_i| + |w_j| \le c - \sum\limits_{k \ne i,j} |w_k^o| \,. \end{array}$$

Here we further assume that $\mathbf{w}^{oT}\mathbf{w}^o = s$ and $\|\mathbf{w}^o\|_1 = c$ (in the following we will prove that this assumption always holds along the generated solution path). Under this assumption, it is easy to deduce that $s - \sum\limits_{k \ne i,j} (w_k^o)^2 = (w_i^o)^2 + (w_j^o)^2$ and $c - \sum\limits_{k \ne i,j} |w_k^o| = |w_i^o| + \left|w_j^o\right|$, and thus $L_{2,c}^{(i,j)}(s)$ can be equivalently written as:

$$[L_{2,c}^{(i,j)}(s+\varepsilon)]: \quad \begin{array}{c} \max\limits_{w_i,w_j} V(w_i,w_j) \\ s.t.\ w_i^2+w_j^2 \le (w_i^o)^2+(w_j^o)^2+\varepsilon \\ |w_i|+|w_j| \le |w_i^o|+\left|w_j^o\right|. \end{array} \tag{5}$$

We then introduce an easy strategy to heuristically attain the direction v, along which the objective function $V(w_i,w_j)$ tends to be largest increased in the feasible region of $L_{2,c}^{(i,j)}(s+\varepsilon)$, in the following.

Since $\mathbf{w}^o$ corresponds to the solution to $L_{2,c}(s)$, we can reasonably assume the following KKT conditions (we will discuss the reasonability of this assumption in Section 2.4):

$$\begin{array}{rcl} 2XX^T\mathbf{w}^o-\gamma\mathbf{w}^o-\lambda sign(\mathbf{w}^o) &=& 0 \\ \gamma\left(\mathbf{w}^{oT}\mathbf{w}^o-s-\epsilon\right) &=& 0, \\ \lambda\left(\|\mathbf{w}^o\|_1-c\right) &=& 0, \\ \gamma \quad \ge \quad 0, \lambda &\ge& 0. \end{array} \tag{6}$$

Since $\Delta\mathbf{V}(\mathbf{w})=2XX^T\mathbf{w}$, the gradient direction $\Delta V=(v_i^o,v_j^o)^T$ of $V(w_i,w_j)$ at (w_i^o,w_j^o) can be attained by:

$$v_i^o=sign(w_i^o)(\gamma|w_i^o|+\lambda),\ v_j^o=sign(w_j^o)(\gamma\left|w_j^o\right|+\lambda). \tag{7}$$

Assume $|w_i^o|\ge|w_j^o|$ without loss of generality, we can then get that: for the orthogonal directions $v=(sign(w_i^o),-sign(w_j^o))^T$ and $v'=(sign(w_i^o),sign(w_j^o))^T$, it holds that

$$v^T\Delta V=\gamma(|w_i^o|-\left|w_j^o\right|)\ge 0,\ v'^T\Delta V=\gamma(|w_i^o|+\left|w_j^o\right|)+2\lambda>0.$$

It can then be deduced that the largest increase of the cost function $V(w_i,w_j)$ at $(w_i^o,w_j^o)^T$ in the feasible region of $L_{2,c}^{(i,j)}(s+\varepsilon)$ is to be attained along the direction $v=\left(sign(w_i^o),-sign(w_j^o)\right)^T$, i.e., along the edge of the l_1 constraint area $\Omega_c=\{(w_i,w_j)^T|\,|w_i|+|w_j|\le|w_i^o|+\left|w_j^o\right|\}$, as depicted in Figure 2. For small stepsize ε, the optimum $\mathbf{w}^*=(w_1^*,w_2^*,\cdots,w_d^*)^T$ of $L_{2,c}(s+\varepsilon)$ is thus expected to be obtained by

$$w_k^*=\left\{\begin{array}{c} w_k^o,\ \text{for } k\ne i,j, \\ w_i^o+sign(w_i^o)\eta,\ \text{for } k=i, \\ w_j^o-sign(w_j^o)\eta,\ \text{for } k=j, \end{array}\right. \tag{8}$$

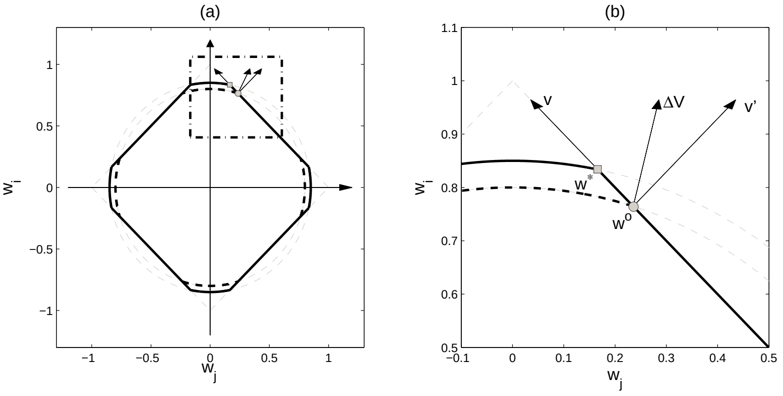

Figure 2: Graphical illustration for the mechanism of the pairwise-cooperate stagewise updating process of the proposed method. The right panel depicts the demarcated area of the left panel in a larger scale. The areas surrounded by the dashed and solid curves represents the the constraint areas of $L_{2,c}^{(i,j)}(s)$ and $L_{2,c}^{(i,j)}(s+\varepsilon)$, respectively. The circle and square represent $\mathbf{w}^o$, $\mathbf{w}^*$ involved in (8), respectively; $\Delta V = (v_i, v_j)^T$ denotes the gradient direction of $V(w_i, w_j)$; and $v = (sign(w_i^o), -sign(w_j^o))^T$ and $v' = (sign(w_i^o), sign(w_j^o))^T$ are two orthogonal directions. It is easy to see that v lies on the edge of the l_1 constraint area of $L_{2,c}^{(i,j)}(s)$, $\Omega_c = \{(w_i, w_j)^T | \, |w_i| + |w_j| \leq |w_i^o| + |w_j^o|\}$.

where the stepsize η from $\mathbf{w}^o$ to $\mathbf{w}^*$ can easily be computed by

$$\begin{array}{lc} & (w_i^o + sign(w_i^o)\eta)^2 + \left(w_j^o - sign(w_j^o)\eta\right)^2 - (w_i^o)^2 - \left(w_j^o\right)^2 = \epsilon \\ \Rightarrow & 2\eta^2 + 2(|w_i^o| - \left|w_j^o\right|)\eta = \epsilon \\ \Rightarrow & \eta = \frac{\epsilon}{\sqrt{(\left|w_i^o\right| - \left|w_j^o\right|)^2 + 2\epsilon} + \left|w_i^o\right| - \left|w_j^o\right|}. \end{array} \quad (9)$$

By updating $\mathbf{w}^o$ to $\mathbf{w}^*$ as aforementioned, it is easy to see that the assumptions $\mathbf{w}^{*T}\mathbf{w}^* = s + \varepsilon$ and $\|\mathbf{w}^*\|_1 = c$ still hold. Such pairwise-coordinate updating can thus be successively implemented until the convergence condition is met. All of the aforementioned can be easily understood by observing the graphical illustration of Figure 2.

It should be noted that after the pairwise-coordinate updating (8), the increased value of the cost function $V(\mathbf{w})$ can be easily calculated as follows:

$$\begin{aligned} \mathcal{J}(i, j, \epsilon) &= V(\mathbf{w}^*) - V(\mathbf{w}^o) \\ &= \left(sign(w_i^o)v_i^o - sign(w_j^o)v_j^o\right)\eta \\ &\qquad + \left\|sign(w_i^o)x_i - sign(w_j^o)x_j\right\|_2^2 \eta^2. \end{aligned} \quad (10)$$

The above $\mathcal{J}(i, j, \epsilon)$ can thus be taken as a reasonable criterion against the

Algorithm 1 Coordinate-Pairwise Algorithm for Sparse PCA (COP-PCA)

```
Given: $X = [x_1, \cdots, x_d]^T \in R^{d \times n}$, the stepsize $\varepsilon$
Execute:
    1. Compute the optimal solution $\mathbf{w}_{pca}$ of $L_{pca}$; $\mathbf{w}(0) \leftarrow \mathbf{w}_{pca}$; $t \leftarrow 0$
    2. Repeat
        2.1. $\mathcal{I} \leftarrow$ nonzero element index of $\mathbf{w}(t)$
        2.2. $(i^*, j^*) \leftarrow \operatorname*{argmax}_{i,j \in \mathcal{I}} \mathcal{J}(i, j, \varepsilon)$
        2.3. $\eta \leftarrow \frac{\varepsilon}{\sqrt{(|w_{i^*}^t|-|w_{j^*}^t|)^2+2\varepsilon}+|w_{i^*}^t|-|w_{j^*}^t|}$
        2.4. If $|w_{j^*}^t| < \eta$, amend $\eta = |w_{j^*}^t|$, and go to step 2.5
        2.5. $\mathbf{w}(t+1) \leftarrow (w_1^{t+1}, w_2^{t+1}, \cdots, w_p^{t+1})^T$, where $w_{i^*}^{t+1} = w_{i^*}^t + sign(w_{i^*}^t)\eta$,
             $w_{j^*}^{t+1} = w_{j^*}^t - sign(w_{j^*}^t)\eta$, and $w_k^{t+1} = w_k^t$ for $k \neq i^*, j^*$
        2.6. $t \leftarrow t + 1$
      Until the termination condition is satisfied
Return: the solution path $\{\mathbf{w}(0), \mathbf{w}(1), \mathbf{w}(2), \cdots\}$ of $L_2(c, s)$
```

aforementioned problem (i), i.e., the proper selection of the pairwise coordinates to be updated. It should be noted that the zero element of $\mathbf{w}^o$ should not be selected as the candidate since its absolute value cannot be decreased any more and the updating step (8) cannot be implemented for such element.

The above analysis implies that the solution path of $L_2(c, s)$ with varying s can be sequentially approximated by iteratively updating the pairwise coordinates (see Eq. (8)) along which the maximum of $\mathcal{J}(i, j, \varepsilon)$ (see Eq. (10)) can be attained. The initial point $\mathbf{w}^o$ can be appropriately set as the optimal solution $\mathbf{w}_{pca}$ to the L_{pca} model. It is easy to deduce that such $\mathbf{w}^o$ corresponds to the solution to $L_{2,c}(s)$ where $c = \|\mathbf{w}_{pca}\|_1$ and $s = 1$[1].

2.3. The coordinate-pairwise algorithm for sparse PCA

We imbed the coordinate-pairwise updating technique as aforementioned into Algorithm 1 (called the COP-PCA algorithm briefly). It is easy to observe that the algorithm only involves simple computations and thus is easy to be implemented. Here we only discuss the method for one PC vector. More PCs of the data can be approximately constructed by applying the proposed algorithm greedily to the remainder of the projected data into the orthogonal spaces to the obtained PC vectors.

[1] After the initialization, the proposed method is to incrementally track the solution path of $L_{2,c}(s)$ under fixed $c = \|\mathbf{w}_{pca}\|_1$ and gradually increased s.

It should be noted that when $|w_{j^*}^t| < \eta$, step 2.4 is activated to amend the stepsize η for updating $w_{i^*}^t$ and $w_{j^*}^t$ in the algorithm. This is because in this case, the stepsize η calculated in step 2.3 of Algorithm 1 will conduct the abnormity that the updated $\mathbf{w}(t+1)$ goes out of the feasible region of $L_{2,c}(s+\varepsilon)$. In specific, let $\eta_1 = \eta - |w_{j^*}^t|$, and then we have

$$\begin{aligned} & |w_{i^*}^t + sign(w_{i^*}^t)\eta| + |w_{j^*}^t - sign(w_{j^*}^t)\eta| \\ = \; & |w_{i^*}^t + sign(w_{i^*}^t)\eta_1 + sign(w_{i^*}^t)|w_{j^*}^t|| \\ & \quad + |w_{j^*}^t - sign(w_{j^*}^t)\eta_1 - sign(w_{j^*}^t)|w_{j^*}^t|| \\ = \; & |w_{i^*}^t| + \eta_1 + |w_{j^*}^t| + \eta_1 \\ > \; & |w_{i^*}^t| + |w_{j^*}^t|. \end{aligned}$$

Step 2.4 of the proposed algorithm easily resolves this problem by shortening the stepsize η as a smaller $\eta_1 = |w_{j^*}^t|$. It is easy to see that as long as this step is activated, $w_{j^*}^{t+1}$ attains 0 and the corresponding label j^* is to be moved out from the non-zero element index $\mathcal{I}$, and simultaneously the number of nonzero elements (i.e., the sparsity) of $\mathbf{w}(t)$ reduces one along the solution path so generated. It can thus be deduced that this step is to be activated no more than $d-1$ times.

The remainder problem is the proper specification of the stepsize ε and appropriate setting of the termination condition for coordinate-pairwise updating iterations. When we initiate $\mathbf{w}_{pca}$ (the solution of L_{pca}) as the starting point of our algorithm, a natural idea is to implement the iterations of the algorithm until the sparsity of $\mathbf{w}(t)$ is reduced to one. In this process, the coefficients of $\mathbf{w}(t)$ shrink to 0s one by one based on their capability of catching up the variance information $V(\mathbf{w})$ from data. Along the solution path so generated, the l_2 constraint parameter s monotonically increases from 1 (corresponding to $\mathbf{w}(0) = \mathbf{w}_{pca}$) to $\|\mathbf{w}_{pca}\|_1^2$ (corresponding to the last element in the solution path, where only one nonzero element left in $\mathbf{w}(t)$), and in each iteration, $\mathbf{w}(t+1)^T\mathbf{w}(t+1)$ brings ε increase to $\mathbf{w}(t)^T\mathbf{w}(t)$. Thus, the stepsize ε and the number of iteration steps $IterNum$ is of the following relationship:

$$IterNum = \frac{\|\mathbf{w}_{pca}\|_1^2 - 1}{\varepsilon}. \tag{11}$$

This means that instead of directly specifying ε, we can more easily preset an appropriate iteration number $IterNum$ for the algorithm, and the stepsize ε

is then implied to be $\frac{\|\mathbf{w}_{pca}\|_1^2-1}{IterNum}$ based on Eq. (11). Under such specification, the algorithm is to be terminated after $IterNum$ iterations, and the entire solution path of the $L_2(c,s)$ model with respect to s is simultaneously to be achieved. Besides, if the proper PC sparsity is known beforehand by prior knowledge or experience, then we can simply initiate a small ε or equivalently a large iteration number $IterNum$, and repeat the coordinate-pairwise updating of $\mathbf{w}(t)$ until its sparsity attains the pre-specified value.

2.4. Computational complexity

In this subsection we discuss the computational complexity of the proposed algorithm. While only very simple computations are involved in the coordinate-pairwise updating process (i.e., steps 2.3-2.6), the computation of the proposed COP-PCA algorithm is mainly costed on sorting the $d(d-1)/2$ elements of $\mathcal{J}(i,j,\varepsilon)$ (i.e., step 2.2), which needs around $O(d^2 \text{log} d)$ time by utilizing the well known heapsort algorithm. This cost, however, can be further alleviated since the $\{i,j\}$ candidates which are possibly chosen as the maximum of $\mathcal{J}(i,j,\varepsilon)$ can be picked up from $\mathcal{I}$ by some useful prior information, as described in the following.

Based on Eq. (10), by omitting the $o(\varepsilon)$ element of $\mathcal{J}(i,j,\varepsilon)$, it can be approximated as

$$\begin{aligned}\mathcal{J}(i,j,\varepsilon) &\approx \frac{|v_i^o| - |v_j^o|}{\sqrt{(|w_i^o| - |w_j^o|)^2 + 2\varepsilon} + |w_i^o| - |w_j^o|} \\ &= \frac{\gamma}{\sqrt{1 + 2\frac{\varepsilon}{|w_i^o| - |w_j^o|}} + 1}.\end{aligned}$$

This naturally implies the following fact: Instead of sorting the $d(d-1)/2$ elements of $\mathcal{J}(i,j,\varepsilon)$, we only need to sort $O(d)$ elements of $\{|w_k^t|, k \in \mathcal{I}\}$, and collect m ($m \ll d$) largest and smallest ones from them as candidates for further comparison of $\mathcal{J}(i,j,\varepsilon)$. The computation of the proposed algorithm is then decreased to $O(nd\text{log}d) \times IterNum$ correspondingly. This means that the computational complexity of the proposed algorithm is approximately linear in both the size and the dimensionality of the input data.

As compared with the computational complexities of the current sparse PCA methods, such as $O(nd^3) \times IterNum$ of SPCA, $O(nd^4\text{log}d) \times IterNum$ of DSPCA, $O(nd\text{log}d) \times IterNum$ of EMPCA, $O(nd^2) \times IterNum$ of ALSPCA, $O(nd) \times IterNum$ of GPowerl_0, and $O(nd^3) \times IterNum$ of PathSPCA (where $IterNum$ is the iteration number of the corresponding method), it is evident that the proposed algorithm is of a comparable computational complexity for sparse PCA calculation. Then the advantage of the new algorithm is obvious: it is capable of yielding the entire solution path of the sparse PCA model only under one such computation. Also, because along the path, each solution is ameliorated gradually by fully making use of the previous solution information, the proposed algorithm is expected to perform consistently well to get the entire solutions to the problem.

2.5. Discussion on the reasonability of the KKT assumption

The reasonability of the KKT assumption (6), as well as the COP-PCA algorithm, lies in the following two aspects. First, the solution path generated by the COP-PCA method is capable of effectively tracking the exact path of $L_{2,c}(s)$ in all of our implemented experiments. This empirically validates that the KKT conditions (6) of $L_{2,c}(s)$ are expected to be satisfied along the generated path. Second, the path $\mathbf{w}(t)$ generated from Algorithm 1 is always beneficial to explore the intrinsic information of sparse PCs of the data at the following four-fold aspects: (i) $\|\mathbf{w}(t)\|_1$ keeps to be a constant (i.e., $\|\mathbf{w}_{pca}\|_1$) along the path; (ii) $\mathbf{w}(t)^T\mathbf{w}(t)$ linearly increases (with the slope ε) along the path; (iii) $V(\mathbf{w}(t))$ (the objective function) monotonically increases along the path; (iv) the sparsity of $\mathbf{w}(t)$ monotonically decreases from d to 1 along the path (Figure 5 graphically illustrates these aspects). The path so generated thus provides a very useful spectrum underlying the the intrinsic implementation mechanism of the sparse PCA model $L_{2,c}(s)$, as substantiated in the following experiments.

3. Experiments

To evaluate the performance of the proposed COP-PCA algorithm, it was applied to several synthetic and real problems. For comparison, 9 of the current sparse PCA methods, including SPCA [4], DSPCA [5], PathSPCA [6], EMPCA [10], GPower$_{l_1}$, GPower$_{l_0}$, GPower$_{l_1,m}$, GPower$_{l_0,m}$[8], and ALSPCA [11], have also been utilized. The results are summarized and interpreted in the following discussion. It should be noted that for each problem, the sparse PCs corresponding to different sparsity constraint parameters were attained by executing the proposed algorithm only once, while by running the other competing methods multiple times.

3.1. Hastie data

The Hastie data set was firstly proposed by [4], and has become one of the most frequently utilized data for the performance evaluation of sparse PCA. The data set contains a collection of 10-D data points $(x_1, \cdots, x_{10})^T$ generated via the following two processes: firstly three hidden factors were created:

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

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

$$\begin{aligned} x_i &= V_1 + \varepsilon_i,\ \varepsilon_i \backsim N(0, 1),\ i = 1, 2, 3, 4, \\ x_i &= V_2 + \varepsilon_i,\ \varepsilon_i \backsim N(0, 1),\ i = 5, 6, 7, 8, \\ x_i &= V_3 + \varepsilon_i,\ \varepsilon_i \backsim N(0, 1),\ i = 9, 10, \end{aligned} \tag{12}$$

where ε_is $(i = 1, \cdots, 10)$ are independent. It has been clarified that the data generated as above are of intrinsic sparse PC vectors [4]. The first PC vector should recover the factor V_2 only using (x_5, x_6, x_7, x_8), and the second should recover the factor V_1 only using (x_1, x_2, x_3, x_4). The 9 current sparse PCA methods and the proposed COP-PCA method were respectively employed to calculate the first two PC vectors of the Hastie data. Through properly tuning parameters, all of the employed methods, except EMPCA, Gpower$_{l_{0,m}}$, and Gpower$_{l_{1,m}}$, can faithfully deliver the ideal sparse representations of the first two PCs underlying the data. The speciality of COP-PCA is that it further generates the smooth solution paths for the corresponding sparse PC vectors, as depicted in Figure 3. The path intuitively depicts the intrinsic evolution process of the corresponding PC vector when it varies from dense to sparse.

3.2. Pitprop data

The pitprop data firstly introduced in [24] contain 180 observations and 13 measured variables. It is the classic example showing the difficulty of interpreting principal components [4, 9]. For the first PC of this data set, the proposed COP-PCA method, together with DSPCA, EMPCA, GPower$_{l_1}$, and GPower$_{l_0}$, consistently deliver the ideal PC vector with different pre-specification of its sparsity[2]. This can be easily observed from Figure 4(a).

[2] We have tried but failed to properly tune the parameters of Gpower$_{l_{0,m}}$, Gpower$_{l_{1,m}}$ in the pitprop data, and hence both of their results are not involved in this section.

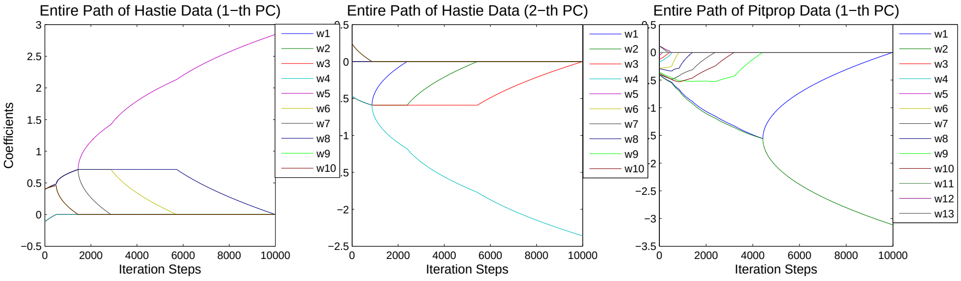

Figure 3: The entire solution paths corresponding to the first (the left panel) and second (the middle panel) PCs of the Hastie data, and the first PC of the pitprop data (the right panel). All paths were generated from the proposed COP-PCA method by presetting the iteration number as 10000.

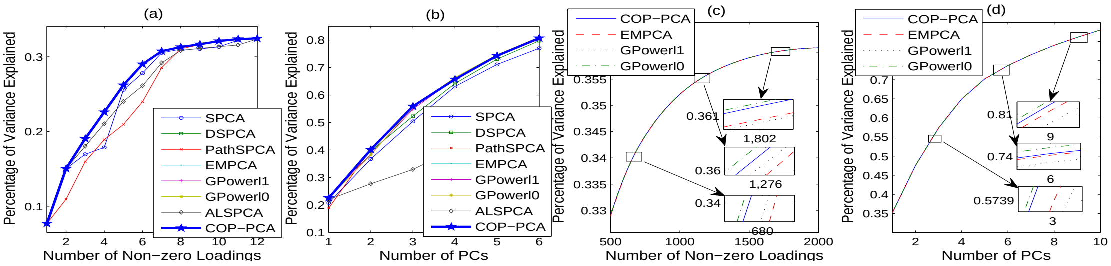

Figure 4: (a)(c) Percentage of explained variances with respect to different pre-specified cardinalities of the first PC vector attained by applying the employed methods to pitprop data and colon cancer data, respectively. (b)(d) Percentage of cumulative variances explained by the first 6 and 10 PCs attained by applying the sparse PCA methods to pitprop data and colon cancer data, respectively. The embedded sub-panels in (c)(d) depict the amplifications of the positions the corresponding arrows point from.

It is clear that more variance is explained by these methods than other employed sparse PCA methods with the same number of non-zero PC loadings. Besides, since the first 6 PCs of the data capture 87% of the total variance, we compared the explanatory powers of 6 sparse PCs of all these employed methods. The COP-PCA captures 80.68% of the total variance with cardinality pattern of $(4, 4, 4, 4, 4, 4)$ (totally 24 non-zero loadings), as compared with 76.99% of SPCA, 79.71% of DSPCA, 80.28% of PathSPCA, 80.68% of EMPCA, 80.95% of GPower$_{l_1}$, 81.04% of Gpower$_{l_0}$, 53.59% of ALSPCA under the same PC cardinality settings, as depicted in Figure 4(b) respectively. It is evident that as compared to the current sparse PCA methods, the proposed method is of the comparable, or even better performance on variance-capturing capability on the first 6 PCs. The prominence of the COP-PCA method lies on the fact that it can further generate the entire solution path of the problem, as depicted in the right panel of Figure 3. It is seen that the 13 variables of the data sequentially shrink to zeroes, intrinsically reflecting their different significance on capturing data variance.

3.3. Colon cancer data

The colon cancer data [7] consist of 62 tissue samples (22 normal and 40 cancerous) with the gene expression profiles of 2000 genes extracted from DNA micro-array data. The biological background of the data makes it a suitable candidate for studying the performance of sparse PCA methods where feature selection is needed to get interpretable results. We have performed the 9 current sparse PCA methods on the colon cancer data, while the experiments on SPCA, DSPCA, PathSPCA, GPower$_{l_1,m}$, GPower$_{l_0,m}$, and ALSPCA could not be completed in reasonable time. Thus the results do not include the results of these methods.

The first PC vector of the data with different specified number of non-zero loadings (from 500 to 1999) were calculated by EMPCA, Gpower$_{l_1}$, Gpower$_{l_0}$, and COP-PCA methods, respectively. The mean variance explained by COP-PCA among these cardinality specifications is 0.003382%, 0.003191%, and −0.003074% more than those of EMPCA, Gpower$_{l_1}$, and Gpower$_{l_0}$, respectively. It can be clearly observed from Figure 4(c) that such deviations are very unsubstantial. The similar phenomenon is observed when applying these methods to compute the first 10 sparse PCs of the data. The corresponding cardinalities of these PC vectors were all set as 1000 for easy comparison. As compared to 84.1821% of the total variance captured by the first 10 PCs of classical PCA, COP-PCA, EMPCA, Gpower$_{l_1}$, and Gpower$_{l_0}$

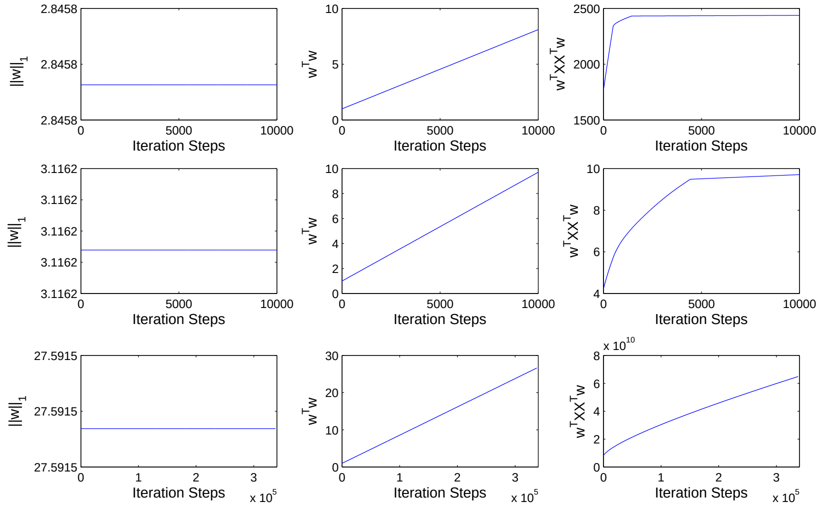

Figure 5: The tendency curves of the l_1 constraint $\|\mathbf{w}(t)\|_1$, the l_2 constraint $\mathbf{w}(t)^T\mathbf{w}(t)$, and the objective $V(\mathbf{w}(t))$, corresponding to the first PC solution path yielded from the COP-PCA method. The first, second, third rows of panels depict the results obtained on the Hastie data, the pitprop data and the colon cancer data, respectively.

explain 83.0324%, 83.0309%, 83.0269%, and 83.0341% of the total variance by their corresponding first 10 sparse PC vectors, as demonstrated in Figure 4(d). From the figure, it is evident that the four cumulative variance curves cannot be materially distinguished, either. These results show that for such data, all of the four employed methods are of similar capabilities on sparse PCA computation, while the proposed method dominates on its meaningful exploration on the entire solution path of the sparse PC vectors and easy specification of initial parameters.

Besides the above results, we also depict in Figure 5 the tendency curves of the l_1 constraint $\|\mathbf{w}(t)\|_1$, the l_2 constraint $\mathbf{w}(t)^T\mathbf{w}(t)$, and the objective $V(\mathbf{w}(t))$ of the $L_{2,c}(s)$ model, corresponding to the solution paths of the first PC of the Hastie data, the pitprop data and the colon cancer data, as calculated by the COP-PCA method, respectively. It is easy to observe that the l_1 constraint $\|\mathbf{w}(t)\|_1$ keeps to be a constant, the l_2 constraint $\mathbf{w}(t)^T\mathbf{w}(t)$ linearly increases, and the objective $V(\mathbf{w}(t))$ monotonically increases along the generated solution path $\mathbf{w}(t)$. All of these results are consistent with our theoretical arguments presented in the end of Section 2, and thus further substantiate the intrinsic effectiveness mechanism of the proposed method.

3.4. Computation complexity evaluation

As analyzed in Section 2.4, the computational cost of the proposed method is comparable or even less than the current sparse PCA methods. In this section we want to further verify this point through experiments. For this task, we have designed 18 extended Hasite data sets with dimensions ranging from $d = 10$ to $d = 180$ with interval 10. Each data set contains a collection of data points $(x_1, x_2, \cdots, x_d)^T$ generated in the following way: firstly create three hidden factors:

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

where $\varepsilon \backsim N(0, 1)$, and V_1, V_2 and ε are independent; and then generate d observed variables as:

$$\begin{aligned} x_i &= V_1 + \varepsilon_i,\ \varepsilon_i \backsim N(0,1),\ i = 1, 2, \cdots, 0.4d, \\ x_i &= V_2 + \varepsilon_i,\ \varepsilon_i \backsim N(0,1),\ i = 0.4d + 1, 0.4d + 2, \cdots, 0.8d, \\ x_i &= V_3 + \varepsilon_i,\ \varepsilon_i \backsim N(0,1),\ i = 0.8d + 1, 0.8d + 2, \cdots, d, \end{aligned} \tag{13}$$

where ε_is $(i = 1, \cdots, d)$ are all independent. Just as the Hastie data, the data sets so generated are also of intrinsic sparse PC vectors. The first PC vector tends to recover the factor V_2 only using $(0.4d + 1, 0.4d + 2, \cdots, 0.8d)$ variables, and the second should recover the factor V_1 only using $(1, 2, \cdots, 0.4d)$ ones. The 9 competing sparse PCA methods, including SPCA, PathSPCA, EMPCA, GPower$_{l_1}$, GPower$_{l_0}$, GPower$_{l_1,m}$, GPower$_{l_0,m}$, ALSPCA and the proposed COP-PCA method were utilized to calculate the first two sparse PC vectors of each data set. We recorded the computation times of these methods and compared their efficiency in Figure 6. To make a fair comparison, we set the maximal iteration number of all competing methods for calculating each PC vector as 100. The actual average iteration number of the competing methods among these 18 experiments are: SPCA, 200; PathSPCA, 76; EMPCA, 200; GPower$_{l_1}$, 8.3; GPower$_{l_0}$, 6.7; GPower$_{l_1,m}$, 6.3; GPower$_{l_0,m}$, 5.7; ALSPCA, 5.6; and COP-PCA, 200, respectively. The average time of the utilized methods among these experiments are: SPCA, 0.2157s; PathSPCA, 0.1466s; EMPCA, 1.8792s; GPower$_{l_1}$, 0.0614s; GPower$_{l_0}$, 0.0496s; GPower$_{l_1,m}$, 0.0795s; GPower$_{l_0,m}$, 0.0646s; ALSPCA, 0.1574s; and COP-PCA, 0.1128s, respectively.

From the above statistics and Figure 6, it can be seen that the computation cost of the proposed COP-PCA method is comparable to the other

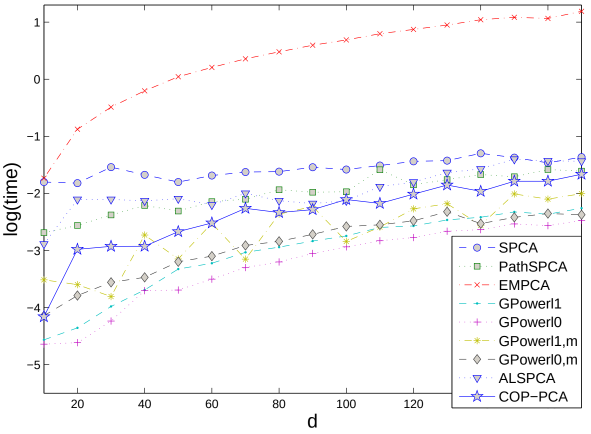

Figure 6: The computational times of SPCA, PathSPCA, EMPCA, GPower$_{l_1}$, GPower$_{l_0}$, GPower$_{l_1,m}$, GPower$_{l_0,m}$, ALSPCA and COP-PCA on 18 extended Hastie data sets with the dimensions varying from $d = 10$ to $d = 180$.

competing sparse PCA methods (a little higher than GPower$_{l_1}$, GPower$_{l_0}$, GPower$_{l_1,m}$ and GPower$_{l_0,m}$, while lower than SPCA, PathSPCA, EMPCA and ALSPCA). This complies with our theoretical analysis aforementioned in Section 2.4.

4. Conclusion

Inspired by the early path methods constructed on the other settings, we have proposed a new path-following algorithm for the sparse PCA problem. The proposed algorithm is simple and easy to be implemented, and is expected to effectively explore the entire solution path of the sparse PCA model. Along the path so generated, the data variables sequentially shrink to zeroes, intrinsically reflecting their different significance on capturing data variance. The path so generated can not only provide great convenience on proper selection of optimal tuning parameter for real sparse PCA applications, but also give further insight into the intrinsic effect of the sparse PCA model.

[1] I. Jollife, *Principal component analysis*, Springer Verlag, New York, 1986.

[2] Y. Kanda, R. Fontugne, K. Fukuda, T. Sugawara. 2013, ADMIRE: Anomaly detection method using entropy-based PCA with three-step sketches, *Computer Communications*, 36(2013) 575-588.

[3] M. Scholz, Validation of Nonlinear PCA, *Neural Processing Letters*, 36(2012) 21-30.

[4] H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, *Journal of Computational and Graphical Statistics*, 15(2006) 265-286.

[5] A. d'Aspremont, L. El Ghaoui, M.I. Jordan, G.R. Lanckriet, A direct formulation for sparse PCA using semidefinite programming, In *NIPS*, 2005.

[6] A. d'Aspremont, F.R. Bach, L.E. Ghaoui, Optimal solutions for sparse principal component analysis, *Journal of Machine Learning Research*, 9(2008) 1269-1294.

[7] B.K. Sriperumbudur, D.A. Torres, G.R.G. Lanckriet, Sparse eigen methods by D.C. programming, In *ICML*, 2007.

[8] M. Journée, Y. Nesterov, P. Richtárik, R. Sepulchre, Generalized power method for sparse principal component analysis, *Journal of Machine Learning Research*, 11(2010) 517-553.

[9] I.T. Jollife, M. Uddin, A modied principal component technique based on the lasso, *Journal of Computational and Graphical Statistics*, 12(2003) 531-547.

[10] C.D. Sigg, J.M. Buhmann, Expectation maximization for sparse and non-negative PCA, In *ICML*, 2008.

[11] Z.S. Lu, Y. Zhang, *An augmented lagrangian approach for sparse principal component analysis*, Technical report, Department of Mathematics, Simon Fraser University, 2009.

[12] D. Shen, H.P. Shen, J.S. Marron, Consistency of sparse PCA in High Dimension, Low Sample Size contexts. *Journal of Multivariate Analysis*, 115(2013), 317-333.

[13] D.M. Witten, R. Tibshirani, T. Hastie, A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, *Biostatistics*, 10(2009) 515-534.

[14] H.P. Shen, J.Z. Huang, Sparse principal component analysis via regularized low rank matrix approximation, *Journal of Multivariate Analysis*, 99(2008) 1015-1034.

[15] D.Y. Meng, Q. Zhao, Z.B. Xu, Improve robustness of sparse PCA by L1-norm maximization, *Pattern Recognition*, 45(2012) 487-497.

[16] N. Naikal, A.Y. Yang, S.S. Sastry, Informative feature selection for object recognition via sparse PCA, In *ICCV*, 2011.

[17] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angel regression, *Annals of Statistics*, 32(2004) 407-499.

[18] J. Zhu, T. Hastie, S. Rosset, R. Tibshirani, 1-norm support vector machines, In *NIPS*, 2004.

[19] T. Hastie, S. Rosset, R. Tibshirani, J. Zhu, The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5(2004) 1391-1415.

[20] M.Y. Park, T. Hastie, L1-regularization path algorithm for generalized linear models, *Journal of the Royal Statistical Society: Series B*, 69(2007) 659-677.

[21] F.R. Bach, R. Thibaux, M.I. Jordan, Computing regularization paths for learning multiple kernels, In *NIPS*, 2005.

[22] T. Hastie, J. Taylor, R. Tibshirani, G. Walther, Forward stagewise regression and the monotone lasso, *Electronic Journal of Statistics*, 1(2007) 1-29.

[23] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning-Data Mining, Inference, and Prediction, second edition*, Springer Verlag, New York, 2009.

[24] J. Jeffers, Two case studies in the application of principal component, *Applied Statistics*, 16(1967) 225-236.

Appendix: Proof of Theorem 1

Theorem 1. *For the optimal solutions* $\mathbf{w}_{l_1}(t)$ *and* $\mathbf{w}_{l_{2,c}}(s)$ *of* $L_1(t)$ *and* $L_{2,c}(s)$ *models, respectively, it holds that* $\mathbf{w}_{l_1}(t) = \frac{t}{c}\mathbf{w}_{l_{2,c}}(\frac{c^2}{t^2})$ *and* $\mathbf{w}_{l_{2,c}}(s) = \sqrt{s}\mathbf{w}_{l_1}(\frac{c}{\sqrt{s}})$.

Proof:

(1) $\mathbf{w}_{l_{2,c}}(\frac{c^2}{t^2})$ can be attained through the optimization model $L_{2,c}(\frac{c^2}{t^2})$ as follows:

$$\begin{array}{c} \mathbf{w}_{l_{2,c}}(\frac{c^2}{t^2}) = \underset{\mathbf{w}}{\operatorname{argmax}}\ V(\mathbf{w}) = \mathbf{w}^T XX^T \mathbf{w} \\ s.t.\ \mathbf{w}^T\mathbf{w} \le \frac{c^2}{t^2} \\ \|\mathbf{w}\|_1 \le c. \end{array} \tag{14}$$

As a comparison, $\frac{c}{t}\mathbf{w}_{l_1}(t)$ can be obtained through solving the following optimization:

$$\begin{array}{c} \frac{c}{t}\mathbf{w}_{l_1}(t) = \underset{\mathbf{w}}{\operatorname{argmax}}\ V(\frac{t}{c}\mathbf{w}) = \frac{t^2}{c^2}\mathbf{w}^T XX^T \mathbf{w} \\ s.t.\ \frac{t^2}{c^2}\mathbf{w}^T\mathbf{w} \le 1 \\ \frac{t}{c}\|\mathbf{w}\|_1 \le t, \end{array} \tag{15}$$

which is equivalent to the following model:

$$\begin{array}{c} \underset{\mathbf{w}}{\operatorname{argmax}}\ \frac{t^2}{c^2}\mathbf{w}^T XX^T \mathbf{w} \\ s.t.\ \mathbf{w}^T\mathbf{w} \le \frac{c^2}{t^2} \\ \|\mathbf{w}\|_1 \le c. \end{array} \tag{16}$$

Evidently, the two optimization models (14) and (16) are intrinsically equivalent, and thus $\mathbf{w}_{l_{2,c}}(\frac{c^2}{t^2}) = \frac{c}{t}\mathbf{w}_{l_1}(t)$, i.e.,

$$\mathbf{w}_{l_1}(t) = \frac{t}{c}\mathbf{w}_{l_{2,c}}(\frac{c^2}{t^2}). \tag{17}$$

(2) By substituting $s = \frac{c^2}{t^2}$ into (17), we have

$$\mathbf{w}_{l_1}(\frac{c}{\sqrt{s}}) = \frac{1}{\sqrt{s}}\mathbf{w}_{l_{2,c}}(s),$$

i.e.,

$$\mathbf{w}_{l_{2,c}}(s) = \sqrt{s}\mathbf{w}_{l_1}(\frac{c}{\sqrt{s}}).$$

The proof is then completed. ■