Applied Mathematics & Information Sciences **4**(1) (2010), 79–91
– An International Journal

Principal Component Analysis with Weighted Sparsity Constraint

Thanh D. X. Duong[1] and Vu N. Duong[2,3]

[1]Ton Duc Thang University, Vietnam
Email Address: thanhddx@tut.edu.vn

[2]University of Science, VNU-HCM, Vietnam
[3]EUROCONTROL Experimental Center, France
Email Address: vuduong@cse.hcmut.edu.vn

Received September 19, 2008; Revised December 8, 2008

Given a covariance matrix, principal component analysis (PCA) with sparsity constraint considers the problem of maximizing the variance explained by a particular linear combination of the input variables while constraining the number of nonzero coefficients in this combination. However, when loading an input variable is associated with an individual cost, we need to incorporate weights, which represent the loading cost of input variables, into sparsity constraint. And in this paper, we present a version of PCA with weighted sparsity constraint. This problem is reduced to solving some semidefinite programming ones via convex relaxation technique. Two applications of the PCA with weighted sparsity constraint to refine the sparsity constraint of sparse PCA illustrate its efficiency and reliability in practice.

Keywords: Principal component analysis, semidefinite relaxation, semidefinite programming, truncation, iterative re-weighting.

1 Introduction

Sparse decompositions of data are required in many applications. In economics, sparsity increases the efficiency and reduces risk of a portfolio [7], and implies lower transaction cost in financial asset trading strategies as well. In computer vision, sparse decomposition is related to the extraction of some concerned pixels which are relevant parts from images [12]. In machine learning, sparsity is closely related to feature selection and to improved generalization of learning algorithms. And in biology, the sparsity is necessary for finding focalized local patterns hidden in gene expression data analysis [1].

Being first introduced by Pearson in [18], and developed independently by Hotelling in [6], principal component analysis (PCA) has now become a popular technique used to reduce multidimensional data sets to lower dimensions for analysis with applications throughout science and engineering, see [11]. This reduction is achieved by transforming to a new set of variables, the principal components, which are uncorrelated and ordered so that the first few retain most of the variation present in all of the original variables. It aslo can be performed via a singular value decomposition of the data matrix or an eigenvalue decomposition of the data covariance matrix.

A drawback of PCA is the lack of sparseness of the principal vectors since the principal components are usually linear combinations of all variables and the loadings are typically nonzero. This makes it often difficult to be applied in many applications where the principal components would be convenient if these components contained very few nonzero loadings. This leads to appearance of methods to finding sparse principal components explaining most of the variance present in the data. To achieve this, it is necessary to sacrifice some of the explained variance and the orthogonality of the principal components. Rotation techniques in [9] can be consider the first approach. In [22], the author studied simple principal components by restricting the loadings to take values from a small set of allowable integers such as 0, 1 and -1. Simple thresholding techniques [3] was an ad hoc way to deal with the problem, where the loadings with small absolute value are thresholded to zero. SCoTLASS [10] and SLRA [23, 24] were introduced to get modified principal components with possible zero loadings. ESPCA [15] used discrete spectral formulation based on variational eigenvalue bounds and an effective greedy strategy to give provably optimal solutions via branch-and-bound search. For very large problems, SPCA [25] was proposed via a regression type optimization problem and DSPCA [4] via relaxing a hard cardinality constraint with a convex approximation.

In practice, there are many applications in which loading an input variable is associated with an individual cost. In these cases, it is essential to incorporate weights, which represent the loading cost of input variables, into PCA. Weighted PCA has thus been introduced and used in many applications such as learning from incomplete data in [19], giving an efficient search algorithm for motion data in [5], and face recognition in [17], etc. However, incorporating both weight and sparsity into PCA have not been studied yet so far and this paper tries to fill this gap.

By directly incorporating a weighted sparsity criterion in the PCA problem formulation (shortly called WSPCA), we consider a nonconvex optimization containing a weighted sparsity constraint. In the literature, there have not been available result to deal with weighted sparsity constraint. Hence, we introduce a convex-based relaxation approach to reduce WSPCA problem to solving a semidefinite program (SDP), which can be solved efficiently in polynomial time via interior-point methods [20, 21].

This paper is organized as follows. The next section contains the main results, where we introduce WSPCA problem and present its relaxed SDP problem. In Section 3, two application of WSPCA to refine sparsity constraint of sparse PCA are given. These applications are necessary and useful since the outputs of all sparse PCA methods do not satisfy sparsity constraint in general, i.e., if we hope to find a principal component with less than m nonzero entries, the outputs often contain more than m nonzero entries. Section 4 shows convincable numerical results of the two applications on both artificial data and real-life data.

Notation 1.1. In this paper, we denote the set of symmetric matrices of size n by $\mathbf{S^n}$, the vector of ones by $\mathbf{1}$, the cardinality (number of nonzero elements) of a vector x by $\mathbf{Card}(x)$, and the number of nonzero coefficients in a matrix X by $\mathbf{Card}(X)$. For $X \in \mathbf{S^n}$, the notation $X \succeq 0$ means that X is positive semidefinite, $\|X\|_F$ is the Frobenius norm of X, i.e., $\|X\|_F = \sqrt{\mathbf{Tr}(X^2)}$, and $\|x\|_2$ is the 2-Euclide norm for $x \in \mathbb{C}^n$.

2 Main Results

In this section, we derive an SDP relaxation for the problem of maximizing the variance by a vector while constraining its weighted cardinality. Then, we apply the problem to decompose a data covariance matrix into sparse factors.

2.1 Semidefinite relaxation

Let $A \in \mathbf{S^n}$ be a covariance matrix, i.e., $A \succeq 0$, and $w \in \mathbb{R}^n$ be a weight vector with $w_i > 0$ for all $i = 1, \ldots, n$. We consider a WSPCA problem of maximizing the variance of vector $x \in \mathbb{R}^n$ while constraining its weighted cardinality:

$$\begin{array}{ll} \text{maximize} & x^T A x, \\ \text{subject to} & \|x\|_2 = 1, \\ & \sum_{i=1}^{n} w_i \delta(x_i) \leq k, \\ & x \geq 0, \end{array} \tag{2.1}$$

where $\delta : \mathbb{R} \longrightarrow \{0, 1\}$ is defined by

$$\delta(x) = \begin{cases} 0, & \text{if } x = 0, \\ 1, & \text{if } x \neq 0, \end{cases}$$

and the given positive number k restricts the number of nonzero entries of the solution, thus the following inequality should be hold:

$$\min_{i=1,\ldots,n} w_i \leq k \leq \sum_{i=1}^{n} w_i.$$

Hence, by choosing $w = \mathbf{1}$, the weighted sparsity constraint $\sum_{i=1}^{n} w_i \delta(x_i) \leq k$ collapse to the classical sparsity constraint $\mathbf{Card}(x) \leq k$.

Lemma 2.1. *Let* $X = xx^T$*, then* $X \succeq 0$*,* $\mathbf{rank}(X) = 1$*,* $\mathbf{Tr}(AX) = x^T Ax$*,* $\mathbf{Tr}(X) = \|x\|_2$*, and*

$$\sum_{i,j=1}^{n} w_i \delta(X_{ij}) w_j = \left[\sum_{i=1}^{n} w_i \delta(x_i) \right]^2 .$$

Proof. The first four conclusions are well-known results of the lifting procedure for semidefinite relaxation, see [1, 4, 13, 14]. Since $\delta(X_{ij}) = \delta(x_i)\delta(x_j)$, we get the last conclusion. □

By the above lemma, problem (2.1) can be rewritten as follows:

$$\begin{aligned} \text{maximize} \quad & \mathbf{Tr}(AX), \\ \text{subject to} \quad & \mathbf{Tr}(X) = 1, \\ & \sum_{i,j=1}^{n} w_i w_j \delta(X_{ij}) \le k^2, \\ & X \succeq 0, \\ & \mathbf{rank}(X) = 1. \end{aligned} \tag{2.2}$$

It is noticed that problem (2.2) reduces to the convex maximization objective $x^T Ax$ and the nonconvex constraint $\|x\|_2 = 1$ to a linear objective and a linear constraint respectively. However, problem (2.1) is still nonconvex. Hence, we need to relax the nonconvex weighted sparsity constraint and the rank constraint.

Lemma 2.2. *Let* $x, w \in \mathbb{R}^n$ *such that* $w_i > 0$ *for all* $i = 1, \ldots, n$*. Then,*

$$\left[\sum_{i=1}^{n} \sqrt{w_i}\, |x_i| \right]^2 \le \left[\sum_{i=1}^{n} w_i \delta(x_i) \right]^2 \|x\|_2^2 .$$

Proof. By Schwartz inequality,

$$\left[\sum_{i=1}^{n} \sqrt{w_i}\, |x_i| \right]^2 \le \left[\sum_{i=1}^{n} w_i^2 \delta(x_i) \right] \|x\|_2^2 .$$

Moreover, $\sum_{i=1}^{n} w_i^2 \delta(x_i) \le \left[\sum_{i=1}^{n} w_i \delta(x_i)\right]^2$, we complete the proof. □

The condition $X \succeq 0$, $\mathbf{rank}(X) = 1$ and $\mathbf{Tr}(X) = 1$ imply that $\|X\|_F = 1$. Thanks to Lemma 2.2, this means that

$$\sum_{i,j=1}^{n} \sqrt{w_i w_j}\, |X_{ij}| \le \left[\sum_{i,j=1}^{n} w_i w_j \delta(X_{ij}) \right]^{1/2} .$$

Using the above inequality and dropping constraint $\mathbf{rank}(X) = 1$, we get a relaxation of (2.2) as follows:

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(AX), \\ \text{subject to} & \mathbf{Tr}(X) = 1, \\ & \sum_{i,j=1}^{n} \sqrt{w_i w_j}\, |X_{ij}| \leq k, \\ & X \succeq 0. \end{array} \tag{2.3}$$

It is remarkable that problem (2.3) is an SDP in the variable $X \in \mathbf{S^n}$, and dropping constraint $\mathbf{rank}(X) = 1$ is the truncation technique as in [1, 13]. This means, we will solve SDP problem (2.3) to get solution X, and an approximation solution of (2.1) is the dominant eigenvector of X.

2.2 Sparse decomposition

Let $A \in \mathbf{S^n}$ be a covariance matrix, we obtain a WSPCA decomposition as the following algorithm:

repeat

1. Solve the SDP (2.3) to get solution X.

2. Let x be is the dominant eigenvector of X. Add x to the solution set of weighted sparse PCA decomposition.

3. Update $A := A - (x^T Ax)xx^T$.

until $\max\{|A_{ij}| : i, j = 1, 2, ..., n\} <$ threshold or the number of principle components attains a specified maximum number.

It is remarkable that the specified maximum number used to terminate the above algorithms should be $\mathbf{rank}(A)$ since, in PCA, the number of principle components is at most $\mathbf{rank}(A)$, see for example [11].

3 An Application to Refine Sparsity Constraint

When $w = \mathbf{1}$, the WSPCA collapses to DSPCA and the the relaxed problem get the form as follows:

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(AX), \\ \text{subject to} & \mathbf{Tr}(X) = 1, \\ & \mathbf{1}^T X \mathbf{1} \leq m, \\ & X \succeq 0, \end{array} \tag{3.1}$$

where the integer number m represents an expectation that the dominant eigenvector of solution X gets atmost m nonzero entries. However, it is well known that the solution of sparse principal component analysis, see for example [3,4,9,10,22–25], does not satisfy the sparsity constraint. Hence, it is desirable to add a post-processing to (3.1) which finding an approximate solution satisfying the sparsity constraint. Next, we present two applications of WSPCA to refine the sparsity constraint of sparse principal component analysis (3.1).

3.1 Re-weight technique

Since large entries of weight vector w in WSPCA problem (2.3) force the solution x to concentrate on the indices where w_i is small, the weights w_i should be chosen to be inversely proportional to the magnitude of solution entries x_i. Then, the new sparsity constraint number k should be chosen to focus on the solution entries x_i with large magnitude (or small weight). The following is a simple iterative algorithm that alternates between estimating solution x and redefining the weights.

Sparse PCA with re-weight technique (RSPCA)

Input: $A \in \mathbf{S^n}$ such that $A \succeq 0$, $m \in [1, \dots, n]$, and $\varepsilon > 0$.

Output: x - a sparse principal component of A.

1. Set the iteration count l to zero $k^{(0)} = m$ and $w_{ij}^{(0)} = 1,\ i, j = 1, \dots, n.$

2. Solve the relaxed WSPCA problem (2.3) to get solution $X^{(l)}$ and set $x^{(l)}$ be the dominant eigenvector of $X^{(l)}$.

3. Update the weights and the sparsity constraint number:

$$w_i^{(l+1)} = \frac{1}{x_i^{(l)} + \varepsilon}, \quad \text{for each } i = 1, \dots, n, \tag{3.2}$$

$$k^{(l+1)} = \text{sum of } m \text{ smallest weights in } w^{(l+1)}. \tag{3.3}$$

4. Terminate when $\mathbf{Card}(x^{(l)}) \le m$ or l attains a specified maximum number of iterations l_{max}, then **return** $x := x^{(l)}$. Otherwise, increment l and go to step 2.

The updating sparsity constraint number in (3.3) is the result of expectation that the $(n - m)$ smallest entries in $x^{(l)}$ will get zero value in the next iteration. And we should set a *threshold* for expected nonzero-valued component $x_i^{(l)}$. Moreover, the parameter $\varepsilon > 0$ in (3.2) should be chosen as $\varepsilon <$ threshold to provide stability and to ensure that a zero-valued component in X_{ij} does not strictly prohibit a nonzero estimate at the next step.

3.2 Truncation technique

Let x be the dominant eigenvector of solution X of un-weight relaxed problem (3.1). Here, in the post-processing, the weights is designed so that the solution of WSPCA has atmost m nonzero entries which is corresponding to the m largest entries of x. Thus, the weights corresponding to the $(n - m)$ smallest entries of x are set to be infinity (in fact, to be $1/\varepsilon$ where ε is very small positive number), and the other weights are set as (3.2). The following is a simple iterative algorithm that finds an appropriate small positive value ε.

Sparse PCA with truncation technique (TSPCA)

Input: $A \in \mathbf{S^n}$ such that $A \succeq 0$, $m \in [1, \ldots, n]$, and $1 > \varepsilon > 0$.

Output: x - a principal component of A with m nonzero entries.

Initial $w = \mathbf{1}$.

repeat

* Solve the relaxed WSPCA problem (2.3) to get solution X,
set x be the dominant eigenvector of X,
and I be the set of index of the m largest entries of x.

* Update the weights, the sparsity constraint number and ε

$$\begin{aligned} w_i &= \frac{1}{x_i + \varepsilon}, \quad \text{for each } i \in I, \\ w_i &= \frac{1}{\varepsilon}, \quad \text{for each } i \notin I, \\ k &= \sum_{i \in I} w_i, \\ \varepsilon &= \varepsilon^2. \end{aligned}$$

Until $\mathbf{Card}(x) \leq m$.

Let $A \in \mathbf{S^n}$ be a covariance matrix, we can also obtain a sparse PCA decomposition as the algorithm in section 2.2.

4 Numerical Experiments

In this section, we will compare the effectiveness of the proposed applications of WSPCA (RSPCA and TSPCA) with the other methods mentioned in the introduction. We perform the test on an artificial data proposed by [25] and a well-known real-life data set - Pit Props data.

4.1 Artificial data

To show the effectiveness of RSPCA and TSPCA as the sparsity constraint refinement post-processing of DSPCA, we consider the simulation example proposed by [25]. In this example, three hidden factors are first created

$$V_1 \sim N(0, 290), \qquad V_2 \sim N(0, 300),$$
$$V_3 = 0.3V_1 + 0.925V_2 + \varepsilon, \quad \varepsilon \sim N(0, 1),$$

$V_1,\ V_2$ and ε are independent.

Then 10 observed variables are generated as the follows

$$
\begin{aligned}
&X_i = V_1 + \varepsilon_i^1, && \varepsilon_i^1 \sim N(0,1), && i = 1,2,3,4,\\
&X_i = V_2 + \varepsilon_i^2, && \varepsilon_i^2 \sim N(0,1), && i = 5,6,7,8,\\
&X_i = V_3 + \varepsilon_i^3, && \varepsilon_i^3 \sim N(0,1), && i = 9,10,\\
&\varepsilon_i^j \text{ are independent}, && j = 1,2,3, && i = 1,\ldots,10.
\end{aligned}
$$

To avoid the simulation randomness, the exact covariance matrix which is an infinity amount of data generated from the above model is used to compute principal components using the different approaches. The variance of the three underlying factors is nearly the same (290, 300 and 283.8, respectively). Since the first two are associated with four variables while the last one is associated with only two variables, V_1 and V_2 are almost equally important, and they are both significantly more important than V_3. In [25], the first two principal components explain 99.6% of the total variance. In [4], by choosing the sparsity constraint $m = 4$, DSPCA gives the same results as SPCA and SCoTLASS which are better than simple thresholding method. Moreover, the output of DSPCA also satisfy the the sparsity constraint $m = 4$ (having 4 nonzero entries). Thus, the sparsity constraint refinement post-processing of DSPCA is not required.

Now, we consider the results of DSPCA when choosing the sparsity constraint $m = 5$ in Table 4.1. The output of DSPCA do not satisfy the sparsity constraint when both the first and second principal components have 6 nonzero entries. Hence, the sparsity constraint refinement post-processing is needed. With the same explained variance, the first principal component of RSPCA (with threshold $= 10^{-2},\ \varepsilon = 10^{-4}$) satisfies the the sparsity constraint after 41 iteration, but the second even after 100 iterations. Here, the results show that TSPCA is the best choice. And it is noticeable that the output of RSPCA may not satisfy the sparsity constraint in some case, but the ones of TSPCA always satisfy.

Table 4.1: The first two principal components with m=5.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	Explained variance
DSPCA, PC_1	0	0	0	0	.49	.49	.49	.49	.14	.14	50.2%
DSPCA, PC_2	-.49	-.49	-.49	-.49	0	0	0	0	.14	.14	41.9%
RSPCA, PC_1	0	0	0	0	.46	.46	.46	.46	0	0.39	49.7%
RSPCA, PC_2	.48	.48	.48	.48	0	0	0	0	-.20	-.17	42.1%
TSPCA, PC_1	0	0	0	0	.45	.45	.45	.45	.42	0	49.8%
TSPCA, PC_2	.49	.49	.49	.49	0	0	0	0	0	-.18	40.7%

4.2 Pit Props Data

The pit props data (consisting of 180 observations and 13 measured variables) was introduced in [8] and is another benchmark example used to test SPCA. All simple thresholding [3], SCoTLASS [10], SPCA [25], and DSPCA [4] have been tested on this data set. As reported in [25], SPCA performs better than SCoTLASS in the sense that it identifies principal components with 7, 4, 4, 1, 1, and 1 nonzero loadings respectively - while explaining nearly the same variance as SCoTLASS, the result SPCA of is much sparser; and better than simple thresholding in the sense that it explains more variance. As reported in [4], DSPCA performs better than SPCA in the sense that it identifies principal components with 6, 2, 3, 1, 1, and 1 nonzero loadings (with respect to sparsity constraint 5, 2, 2, 1, 1, and 1).

Table 4.2: The first three principal components of DSPCA, RSPCA and TSPCA with sparsity constraint 5, 2, 2, 1, 1, and 1.

Method	DSPCA			RSPCA			TSPCA		
Variable	PC_1	PC_2	PC_3	PC_1	PC_2	PC_3	PC_1	PC_2	PC_3
topdiam	-0.56	0	0	0.50	0	0	-0.48	0	0
length	-0.58	0	0	0.51	0	0	-0.49	0	0
moist	0	0.71	0	0.71	0	0	0	0.71	0
testsg	0	0.71	0	0.71	0	0	0	0.71	0
ovensg	0	0	0	0	0	0	0	0	0
ringtop	0	0	-0.79	0	0	0.81	0	0	-0.81
ringbut	-0.26	0	-0.61	0.38	0	0.58	-0.40	0	-0.58
bowmax	-0.1	0	0	0	0	0	0	0	0
bowdist	-0.37	0	0.41	0	0	0	-0.42	0	0
whorls	-0.36	0	0	0.42	0	0	-0.43	0	0
clear	0	0	0	0	0	0	0	0	0
knots	0	0	0	0	0	0	0	0	0
diaknot	0	0	0.01	0	0	0	0	0	0
Variance %	26.6	14.48	13.15	26.17	14.48	12.33	26.20	14.48	12.17

Here, we want to compare the results of RSPCA and TSPCA - using the same sparsity constraint (5, 2, 2, 1, 1, and 1) - with those of DSPCA. The results are given in Table 4.2 with threshold being 10^{-2} and $\varepsilon = 10^{-4}$. While explaining 76.05% variance - nearly the same as DSPCA (77.3%) - the first six principal components of RSPCA satisfies the sparsity constraint after 4, 1, 3, 1, 1, and 1 iterations respectively. TSPCA method explains 75.93% variance without any iteration. It is also remarkable that these results are better than ESPCA (75.9%) in [15]. However, we can see that there is an overlap between the first principal component and the third principal component on entry "ringbut". Hence, it is reasonable to think about a better sparsity constraint as 4, 2, 2, 1, 1, and 1. The outputs for this case are displayed in Figure 4.1, where they also explain a large amount of the variance: 74.09% for RSPCA method and 74.10% for TSPCA method.

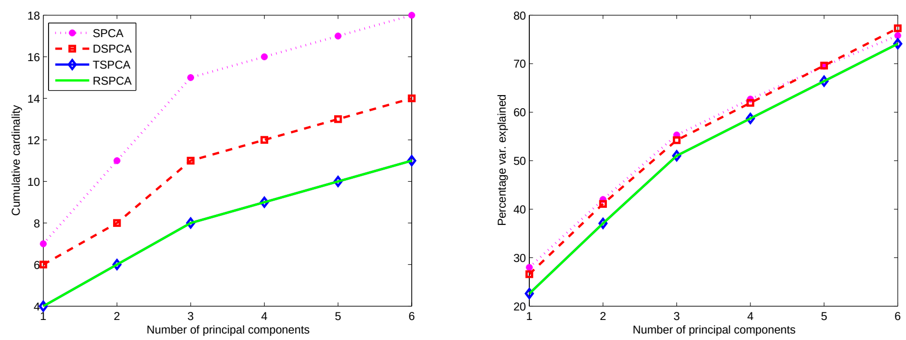

Figure 4.1: Cumulative cardinality and percentage of total variance explained versus number of principal components, for SPCA, DSPCA, RSPCA and TSPCA with sparsity constraint (4, 2, 2, 1, 1, and 1) on the pit props data.

Table 4.3: The first three principal components of RSPCA and TSPCA with sparsity constraint 5, 2, 2, 1, 1, and 1.

Method	RSPCA			TSPCA		
Variable	PC_1	PC_2	PC_3	PC_1	PC_2	PC_3
topdiam	0.50	0	0	0.48	0	0
length	0.51	0	0	0.49	0	0
moist	0	-0.71	0	0	0.71	0
testsg	0	-0.71	0	0	0.71	0
ovensg	0	0	0	0	0	0
ringtop	0	0	-0.69	0	0	-0.70
ringbut	0.38	0	-0.54	-0.40	0	-0.53
bowmax	0	0	0	1.00	0	0
bowdist	0.41	0	0	-0.42	0	0
whorls	0.41	0	0	-0.43	0	0
clear	0	0	0	0	0	0
knots	0	0	0	0	0	0
diaknot	0	0	0.48	0	0	0.50
Variance %	26.17	14.48	14.63	26.2	14.48	14.50

Finally, with the less sparsity results (5, 2, 3, 1, 1, and 1) than DSPCA , the results of RSPCA and TSPCA in Table 4.3 explain more variance than DSPCA (78.35% and 78.26% compared with 77.3%). The first six principal components of RSPCA satisfies the sparsity constraint after 4, 1, 7, 1, 1, and 1 iterations respectively. Figure 4.2 shows the cumulative number of nonzero loadings and the cumulative explained variance. In this figure, we can observe that RSPCA is the best choice for this data.

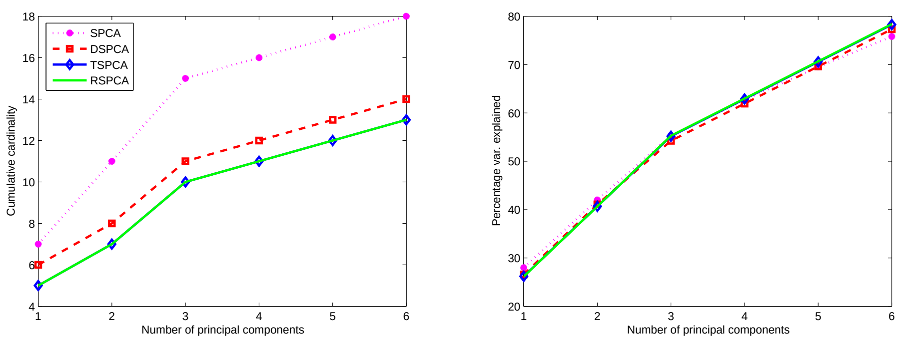

Figure 4.2: Cumulative cardinality and percentage of total variance explained versus number of principal components, for SPCA, DSPCA, RSPCA and TSPCA with sparsity constraint (5, 2, 3, 1, 1, and 1) on the pit props data.

5 Conclusions and perspectives

The application of specific solution will be discussed elsewhere since we want to keep our method general for other weighted sparsity constrained optimizations. In this paper, we attempted to present the principal component analysis with weighted sparsity constraint method (WSPCA) to find the principal components not only explaining most of the variance present in the data but also satisfying weighted sparsity constraints through the solving of semidefinite problems. Two applications of WSPCA to refine the sparsity constraint of sparse PCA has been presented via re-weight technique (RSPCA) and truncation technique (TSPCA). The numerical results show the effectiveness of RSPCA and TSPCA methods. This implies that re-weight technique and truncation technique can be usefully applied to others sparse PCA methods as well as other semidefinite problems containing sparsity constraint.

The drawback of WSPCA is that the SDP problem involved in (2.3) contains more than $O(n^2)$ constraints, which make the memory requirements of Newton's method prohibitive for very large-scale problems. This should be the subject of a future investigation by using smoothing technique, which has recently shown to be reducing memory requirements in solving large-scale SDP problems, see [4, 16]. Finally, finding an efficient re-weight function in (3.2) and updating sparsity number function in (3.3) are also of our priorities.

Acknowledgments

The authors acknowledge discussions with the colleagues in Information Technology and Applied Mathematic Department of Ton Duc Thang University related to the work presented in this paper. They are particularly grateful to the referees for their careful reading and helpful remarks.

References

[1] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, *SIAM J. Optim.* **5** (1995), 13–51.

[2] L. Badea and D. Tilivea, *Sparse factorizations of gene expression guided by binding data*, in: Pacific Symposium on Biocomputing, 2005.

[3] J. Cadima and I. T. Jolliffe, Loadings and correlations in the interpretation of principal components, *J. Appl. Statist.* **22** (1995), 203–214.

[4] A. D'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, A direct formulation for sparse PCA using semidefinite programming, *SIAM Rev.* **49** (2007), 434–448.

[5] K. Forbes and E. Fiume, *An efficient search algorithm for motion data using weighted PCA*, in: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM Press, New York, 2005, 67-76.

[6] H. Hotelling, Analysis of a complex of statistical variables into principal components, *J. Educ. Psychol.* **24** (1933), 417–441.

[7] R. Jagannathan and T. Ma, Risk reduction in large portfolios: Why imposing the wrong constraints helps, *Journal of Finance* **58** (2003), 1651–1684.

[8] J. Jeffers, Two case studies in the application of principal components, *Appl. Statist.* **16** (1967), 225–236.

[9] I. T. Jolliffe, Rotation of principal components: Choice of normalization constraints, *J. Appl. Statist.* **22** (1995), 29–35.

[10] I. T. Jolliffe, N. T Trendafilov, and M. Uddin, A modified principal component technique based on the LASSO, *J. Comput. Graphical Statist.* **12** (2003), 531–547.

[11] I. T. Jolliffe, *Principal Component Analysis*, Springer Verlag, New York, 2002.

[12] D. D. Lee and H. S. Seung, Learning the parts of objects by nonnegative matrix factorization, *Nature* **401** (1999), 788–791.

[13] C. Lemarechal, and F. Oustry, Semidefinite relaxations and lagrangian duality with application to combinatorial optimization, *Rapport de recherche 3710*, INRIA, France (1999).

[14] L. Lovasz, and A. Schrijver, Cones of matrices and set-functions and 0-1 optimization, *SIAM J. Optim.* **1** (1991), 166–190.

[15] B. Moghaddam, Y. Weiss, and S. Avidan, Spectral bounds for sparse PCA: exact & greedy algorithms, *Advances in Neural Information Processing Systems*, MIT Press **18** (2006), 915–922.

[16] Y. Nesterov, Smoothing technique and its application in semidefinite optimization, *Math. Program.* **110** (2007), 245–259.
[17] V. D. M. Nhat and S. Lee, *Two-dimensional weighted PCA algorithm for face recognition*, in: Computational Intelligence in Robotics and Automation, 2005, 219–223.
[18] K. Pearson, On lines and planes of closest fit to systems of points in space, *Phil. Mag.* **2** (1901), 559–572.
[19] D. Skočaj and A. Leonardis, *Weighted and robust incremental method for subspace learning*, in: Ninth IEEE international Conference on Computer Vision (ICCV 2003 II), 2003, 1494–1501.
[20] J. Sturm, Using SEDUMI 1.0x, a MATLAB toolbox for optimization over symmetric cones, *Optim. Methods Softw.* **11** (1999), 625–653.
[21] K. C. Toh, M. J. Todd, and R. H. Tutuncu, SDPT3 - a MA TLAB software package for semidefinite programming, *Optim. Methods Softw.* **11** (1999), 545–581.
[22] S. Vines, Simple principal components, *Appl. Statist.* **49** (2000), 441–451.
[23] Z. Zhang, H. Zha, and H. Simon, Low-rank approximations with sparse factors I: Basic algorithms and error analysis, *SIAM J. Matrix Anal. Appl.* **23** (2002), 706–727.
[24] Z. Zhang, H. Zha, and H. Simon, Low-rank approximations with sparse factors II: Penalized methods with discrete Newton-like iterations, *SIAM J. Matrix Anal. Appl.* **25** (2004), 901–920.
[25] H. Zou, T. Hastie, and R. Tibshirani, Sparse principal component analysis, *J. Comput. Graphical Statist.* **15** (2006), 265–286.

Thanh D. X. Duong was born in Vietnam, in 1982. He received Bachelors Degree in Mathematics and Computer Science with third-class honors in 2003, a Masters Degree in Mathematics in 2007 and a Masters Degree in Computer Science in 2008, and a Doctors Degree in Applied Mathematics in 2009 with high distinction from the University of Science, VNU HCM. He is currently an Assistant Professor at Ton Duc Thang University at HoChiMinh City, Vietnam.

Vu N. Duong is currently Senior Scientific Advisor at EUROCONTROL France, the technical center of the European Organization for the Safety of Air Navigation. Before joining EUROCONTROL in 1995, he had been a research scientist and technical advisor at Schlumberger Ltd. Concurrently, Vu is an Affiliate Professor of Computer Science and Engineering at University of Technology and University of Science, Vietnam National University at HMC. He also teaches at Paris Institute of Technology and is a Charge de Conferences at École Pratique des Hautes Études, Sorbonne Paris. He holds a Masters degree in Engineering and a PhD in Artificial Intelligence both from École National des Ponts et Chaussees, Paris Institute of Technology, France.