Comput Stat (2009) 24:583–604
DOI 10.1007/s00180-008-0147-3

ORIGINAL PAPER

An exact approach to sparse principal component analysis

Alessio Farcomeni

Received: 12 April 2007 / Accepted: 29 December 2008 / Published online: 9 January 2009

Abstract We show a branch and bound approach to exactly find the best sparse dimension reduction of a matrix. We can choose between enforcing orthogonality of the coefficients and uncorrelation of the components, and can explicitly set the degree of sparsity. We suggest methods to choose the number of non-zero loadings for each component; illustrate and compare our approach with existing methods through a benchmark data set.

Keywords Branch and bound · Dimension reduction · Feature selection · Feature extraction · Interleaving eigenvalues theorem · Sparse principal components

1 Introduction

Principal component analysis (PCA) is a popular dimension reduction and descriptive multivariate technique (Chatfield and Collins 1980; Jolliffe 2002). Given an n by p matrix X, a new n by m matrix Y is built. The columns of Y are functions of the columns of the original data matrix, and $m << p$. It is well known that the highest possible variability is explained by Y if its columns (the principal components) are an affine linear combination of the original columns, with weights given by unit length eigenvectors of $X'X$ (the loadings). This is the well-known standard PCA. Furthermore (1) the columns of the derived matrix Y are uncorrelated and (2) the loadings are orthogonal.

In this paper we measure information in a variable by its variance or by its adjusted variance (Zou et al. 2004, see Sect. 3). We will specify which of the two measures is used when not clear from the context.

A. Farcomeni (✉)
Università di Roma "La Sapienza", Rome, Italy
e-mail: alessio.farcomeni@uniroma1.it

There is an impressive number of applications of PCA in biology, medicine, psychology, economics, finance, engineering, etc. In those applications usually X is a two-mode data matrix, in which the n rows represent subjects and the p columns represent numeric variables.

The main drawback of dimensionality reduction through PCA is that each principal component (PC) is in general a linear combination of all the p variables used in input. This complicates the interpretation of the PCs, and does not help the user in discarding less important variables. It is in general believed that sparseness of the loadings would be of great relevance in aiding the interpretation of the derived variables. If principal components were linear combinations of only a small number of original variables, with different variables being used by different components, the subjective interpretation step would be much easier. For this reason, principal components extraction is often followed by some kind of transformation which aims at making the interpretation easier. A common approach is to discard the smallest coefficients (hard thresholding) of the ordinary or rotated principal components. Such "simple thresholding" of the loadings is potentially misleading, and in general does not produce an optimal solution (Cadima and Jolliffe 1995). Principal components are often not easily interpreted even after rotation and thresholding (as it is known in the literature about the example of Sect. 6). Note that interpretability would not be the only advantage of sparseness in the loadings: sparse matrices are better stored and handled. Moreover, variables receiving a zero loading in all m components can be discarded (see also Sect. 3.1), thus performing an automatic feature selection.

Sparse principal components can be obtained also from more formal methods. SCoTLASS (Jolliffe et al. 2003) uses non-convex constrained optimization to derive modified principal components with possible zero loadings. A modification can impose orthogonality. The computational cost is often high, even if an efficient algorithm is proposed in Trendafilov and Jolliffe (2006). The PCA problem is seen as a regression-type optimization by Zou et al. (2004). They use the elastic net (Zou and Hastie 2005) to propose an sPCA algorithm which leads to sparse approximations of the ordinary principal components. The computational cost of sPCA is much lower than SCoTLASS, and often sPCA leads to the same amount of variability in Y with more sparsity; main drawback is that orthogonality of the loadings is not guaranteed.

In both approaches the degree of sparsity is controlled via a penalization parameter, and the choice of such parameter is an open problem. A related problem is that the user does not know in advance if and how sparse the loadings will be. Further, neither approach can lead to uncorrelated components, which are very important in certain settings.

The general goal is then to give non-zero weight to a pre-specified number of variables, with as little information loss as possible with respect to ordinary PCA. In this paper we view the sparse principal component analysis (sPCA) problem as a variability maximization problem and exactly find the optimal solution for prescribed degree of sparsity. We will also be able to choose whether to enforce orthogonality of the loadings or uncorrelation of the new variables. If orthogonality is enforced,

solutions in which different variables are used by different components are favored. There might then be very little overlap among the loading vectors, and the PCA be easily interpretable.

On other hand uncorrelation is useful in practice in many settings, like in principal components regression and whenever the components are used simultaneously. PCA in regression is in fact often used when there are multicollinearity problems. Further, a graphical representation on Cartesian axes, like a biplot, should be given with uncorrelated components. Correlated components do not produce a 90° angle on a graphical representation, and use of Cartesian axes may be misleading.

The main drawback of our exact approach, partly shared with the other methods, is that it is suitable only for small dimensional situations (say up to 30–40 variables). We are currently working on a solution for high-dimensional problems.

Our approach essentially corresponds to use an L_0 penalty for the objective function, as in Moghaddam et al. (2006) and d'Aspremont et al. (2007); while methods proposed by Jolliffe et al. (2003) and Zou et al. (2004) use an L_1 penalty. The L_1 penalty has many advantages over the L_0 penalty, leading to an optimization problem much easier to solve. Nevertheless with L_1 penalizations (1) the number of non-zero loadings is not known in advance and (2) the optimal solution will in general only approximate the optimal solution with L_0 penalty, which is the genuine optimum with fixed number of zeros. An elegant convex relaxation method to solve the sPCA problem (with L_0 penalty) is proposed in d'Aspremont et al. (2007), with the achievement of good approximations to the optimum. One of the goals of this paper is to show that in certain settings (namely, relatively low dimensions) there is no need for using L_1 penalty or look for approximations of the L_0 constrained problem.

Exact solutions of the L_0 constrained problem explain an amount of variability which is guaranteed to be not lower than the amount of variability explained by the other methods, with the same degree of sparsity (and possibly additional constraints).

A similar branch and bound algorithm, using the interlacing argument provided by Theorem 1, was proposed also in Moghaddam et al. (2006). In this paper we use one additional constraint for the new variables, enforcing either orthogonality or uncorrelation, while the approach in Moghaddam et al. (2006) will likely lack both properties. We also introduce the possibility of using a different objective function, namely the adjusted variance of Zou et al. (2004) instead of the variance. Finally, we suggest heuristic and formal strategies for choosing the degree of sparsity, and explicitly describe an efficient branch and bound algorithm. On other hand, Moghaddam et al. (2006) provide also other exact algorithms, perform a comparison also with simple thresholding, and suggest a simple renormalization step which can be used to improve the solution of approximate methods.

The remainder of the paper is organized as follows: in Sect. 2 we formalize the sPCA problem for the first sparse principal component (sPC). In Sect. 3 we show how to derive the other sparse components. In Sect. 4 we propose strategies to choose the degree of sparsity of each sPC. In Sect. 5 we illustrate the method with simulations and in Sect. 6 we will show a real data application. Finally, we give a short discussion in Sect. 7.

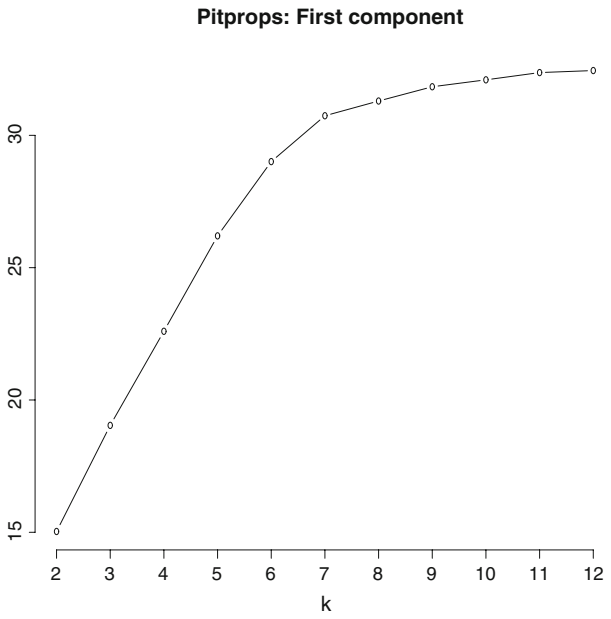

Fig. 1 Uphill test for the first sPC, Pitprops data

2 The sPCA problem

Let $\Sigma = \{\sigma_{ij}; i = 1, \ldots, p, j = 1, \ldots, p\}$ be the covariance matrix of X.

The sparse principal component problem is:

$$\begin{cases} \max_\beta \beta'\Sigma\beta \\ \beta'\beta = 1 \\ |\beta|_0 \leq k_1 \end{cases} \tag{1}$$

where $|x|_0$ denotes the cardinality (number of nonzero elements) of x.

The solution β_1 to Problem (1) is well known to be the first eigenvector of Σ whenever $k_1 \geq p$, and the maximum is the corresponding eigenvalue (the amount of information retained by the first principal component). If $k_1 < p$ there is a genuine sparsity constraint on the loadings vector β: at least $p - k_1$ variables will receive zero weight.

As k_1 is increased the constraint is less and less stringent and $\max_\beta \beta'\Sigma\beta$ grows, or at least does not decrease. This is also a direct implication of Theorem 1 below. For appropriately chosen k_1 very little information loss can happen with respect to the choice $k_1 = p$. For instance in the example of Sect. 6, where $p = 13$, setting $k_1 = 7$ leads to explain 30.7% of the total variability out of the 32.4% possible with $k_1 \geq 13$. For $k_1 = 8$, 31.3% is explained; and the proportion keeps increasing slowly until its maximum. See Fig. 1 for a graphical presentation of this phenomenon.

2.1 The first sparse principal component

We propose now a branch and bound algorithm to exactly solve the Problem (1). We call our method BB-sPCA. Branch and bound algorithms are a clever way to enumerate the possible solutions to a given (difficult) problem. The set of possible solutions is split into subsets by branching, and a criterion is used to bound the solutions into each branch. Finally, only branches for which the bound is bigger than the current maximum are explored. A nice review of branch and bound in statistical analysis is given in Hand (1981).

Algorithm 1. BB-sPCA algorithm

initialize
`activeset` is initialized as the set of all subsets of k_1 variables out of the p, which can be simplified as `activeset` $= \{1, \ldots, p\}$.
`bestset` $= \{1, \ldots, k_1\}$
$j = 1$
$\lambda_0 = \lambda_{\max}(\Sigma_{\texttt{bestset}})$
end initialize
while `activeset` $\neq \emptyset$ **do**
  choose following node $S \in$ `activeset`
  remove node S from `activeset`
  branching step: split node S into two branches S_1 and S_2, one of which is made by the set of groups of k_1 variables containing variable j and the other which is the set of groups of k_1 variables *not* containing variable j.
  $j = j + 1$
  for i=1,2 **do**
    bounding step: $\lambda_i = \lambda_{\max}(\Sigma_{S_i})$
    if $\lambda_i > \lambda_0$ **then**
      if $|S_i|_0 == k_1$ **then**
        $\lambda_0 = \lambda_i$
        `bestset` $= S_i$
      else
        `activeset` $= \{$`activeset`$, S_i\}$
      end if
    end if
  end for
end while

The proposed branch and bound algorithm is briefly described as Algorithm 2.1, and is illustrated in the following. We denote by $\lambda_{\max}(\cdot)$ the operator that computes the largest eigenvalue of a square symmetric matrix, and by Σ_S the covariance matrix obtained from a subset S of the variables. In what follows we assume that the variables are ordered so that the diagonal of the covariance matrix is in not-increasing order. Note that we initialize by setting the current maximum λ_0 as the largest eigenvalue of the first k_1 variables. Since the variables are ordered with respect to their variance, the

first k_1 are a good candidate for final optimality. We remind that k_1 is the chosen final cardinality for the loading vector. In our experience ordering with respect to $\sum_i (\sigma_{ij})^2$ may lead to lower run times (for instance when there are strong correlations).

The interleaving eigenvalues theorem, proposed and proved in Wilkinson (1965), is used for bounding:

Theorem 1 *Let A be a p by p real symmetric matrix and denote by $A^{(-j)}$ the matrix obtained by removing the j-th row and j-th column of A. Let ρ_i be the i-th eigenvalue of A and μ_i the i-th eigenvalue of $A^{(-j)}$. Then,*

$$\rho_1 \geq \mu_1 \geq \rho_2 \geq \mu_2 \geq \cdots \geq \mu_{p-1} \geq \rho_p.$$

Note that removing the jth row and column of a covariance matrix is equivalent to removing the jth variable from the data matrix.

Suppose $\lambda_0 = 100$, $k_1 = 5$ and there are still branches in `activeset`. Suppose the currently explored $|S_i|_0 > k_1$ and $\lambda_i = \lambda_{\max}(\Sigma_{S_i}) = 50$. By Theorem 1 all subsets of S_i will lead to a largest eigenvalue lower than the current maximum λ_0, and so they can be skipped.

In general, whenever a branch S_i leads to λ_i lower than the current maximum, all the subsets of length k_1 can be skipped. If this does not happen, then it is possible that a subset of size k_1 contains a better solution, and S_i is added to the `activeset` for further branching and investigation.

Note that the theorem also implies that the third constraint in Problem (1) is changed into an equality: no vector of cardinality lower than k_1 can lead to a better solution than the optimum with cardinality k_1.

The branching step is the same as the algorithms for best subset selection (Miller 1990): split the current set into subsets in which variables are removed one at a time. To fix the ideas suppose we have $p = 4$ variables, $k_1 = 2$, and $j = 1$. Then,

$$S_1 = \{\{X_1, X_2\}, \{X_1, X_3\}, \{X_1, X_4\}\}$$

and

$$S_2 = \{\{X_2, X_3\}, \{X_2, X_4\}, \{X_3, X_4\}\}$$

at the first branching.

It shall be noted that a single run of Algorithm 2.1 can produce all possible solutions from a chosen k_1 to p; and that the algorithm exactly solves the optimization problem.

3 The other principal components

It is straightforward to check that only the loadings resulting from ordinary PCA can be orthogonal and simultaneously yield uncorrelated components (Jolliffe 1995). We will have to choose then between enforcing orthogonality of the loadings and uncorrelation of the components, by adding a specific constraint into Problem (1). Let β_i be

the loadings of the ith sPC. Uncorrelation between the ith and jth sPC will be given by the constraint $\beta_i' \Sigma \beta_j = 0$, while orthogonality of the loadings is given by $\beta_i' \beta_j = 0$. The only vectors that can satisfy both constraints are the ones obtained through the classical PCA. As soon as constraints are put on the cardinality of any single β_i, only one of the two properties can be retained.

If we enforce orthogonality the resulting components will be correlated. Hence, the additional variability explained by the jth component, with $j > 1$, will in general be lower than its variance. An important choice for the objective function is then between maximization of the variance explained by each component, and the adjusted variance, an index of variability introduced by Zou et al. (2004) to cope with correlated components.

When the extracted features are used separately, for instance for descriptive purposes, it may be desirable to maximize the variance of each component (thereby setting the objective function $J(\Sigma, \beta) = \beta' \Sigma \beta$). In the other cases, Zou et al. (2004) devised how to measure the information/variability in a set of correlated variables, and defined the adjusted variance of the jth component as the square of the jth diagonal element R_{jj} of the upper triangular matrix in the QR decomposition of the new matrix Y (the objective function would then be $J(\Sigma, \beta) = R_{jj}^2$). See Zou et al. (2004) for computational strategies and further comments.

Maximization of the adjusted variance is more sensible in cases in which the extracted features are used jointly. Note that it makes sense to choose the objective function only under orthogonality, since when uncorrelation is enforced the adjusted variance and the variance coincide.

The sPCA problem for the second sPC is then:

$$\begin{cases} \max_\beta J(\Sigma, \beta) \\ \beta' \beta = 1 \\ |\beta|_0 \leq k_2 \\ C(\beta, \beta_1) = 0, \end{cases} \tag{2}$$

where $J(\Sigma, \beta)$ is either the variance or the adjusted variance of the new PC and $C(\beta, \beta_1)$ is either $\beta' \beta_1$ or $\beta' \Sigma \beta_1$. The solution to Problem (2) will be an sPC with k_2 non-zero elements. Note that now the maximum will not necessarily be an eigenvalue of a submatrix of Σ.

Algorithm 2.1 can be used to find a solution, just by substituting the $\lambda_{\max}(\cdot)$ operator with the operator that computes the solution of:

$$\begin{cases} \max_\beta J(\Sigma_{S_i}, \beta) \\ \beta' \beta = 1 \\ C(\beta, \beta_{1,S_i}) = 0, \end{cases} \tag{3}$$

where β_{1,S_i} are the loadings of the variables in subset S_i.

The bounding step is the same in light of the following generalization of the interleaving eigenvalues theorem:

Theorem 2 *Let λ be the maximum for Problem (3) for a given S_i and β_1. Let $\Sigma_{S_i^{(-j)}}$ be in which the jth row and column are removed from S_i, and $\beta_1^{(-j)}$ the vector β_1 in which the jth element is removed. Let μ be the maximum for Problem (3) in which Σ_{S_i} is substituted with $\Sigma_{S_i^{(-j)}}$ and β_1 with $\beta_1^{(-j)}$.*
Then, $\lambda \geq \mu$.

Proof Without loss of generality suppose we remove the last row and column from Σ_{S_i} and correspondingly the last element from β_1. Call y the solution of the reduced problem, with maximum μ. Let $x = \begin{pmatrix} y \\ 0 \end{pmatrix}$. It suffices to show that x satisfies the constraints of the enlarged problem, which can be checked easily. □

Maximization in (3) can be easily and quickly solved by quadratically constrained convex optimization (Gill et al. 1981). Fast routines are available even for large scale problems (Coleman and Li 1994). In order to use those algorithms one should just change the equality constraint $\beta'\beta = 1$ into an inequality $\beta'\beta \leq 1$. The maximum will be the same since it will lie on the boundary of the constraint set. The objective function is obviously convex and so will be the constraint set given by either $\{\beta : \beta'\beta \leq 1 \cap \beta'\beta_1 = 0\}$ or $\{\beta : \beta'\beta \leq 1 \cap \beta'\Sigma\beta_1 = 0\}$.

Further when $C(\beta, \beta_1) = \beta'\beta_1$, whenever the current set S_i is made of variables receiving a zero loading in the first sPC, the solution is the first eigenvalue of the reduced matrix.

For this reason, a good initialization of the algorithm may be given by the optimum of Problem (1) for the variables not used by the first sPC. If the variances are more or less homogeneous, it is reasonable to expect that the second sPC will be a linear combination of variables which receive a zero loading in the first sPC and an early pruning of the tree will happen.

After finding the second sPC, a further constraint can be added to find a third sPC in which only k_3 variables are used, and so on.

As a referee pointed out, imposing arbitrary sparsity can spoil the quality of the solution. In many cases, a solution satisfying the constraints may not even exist. A sufficient condition is $k_i \geq i, i = 2, \ldots, p$, but the solution may exist anyway also for $k_i < i$, for instance if $J(\beta, \beta_i) = \beta'\beta_i$ and at least k_i variables have not been used by the previous components. Nevertheless, so much sparsity is rarely needed, especially in the first sPCs.

3.1 Different criteria

The algorithm can be modified in order to solve other kinds of problems. So far each component was allowed to use any of the variables. We could further constrain the m sparse principal components to use the same k variables. This would provide m linear combinations in which just k variables are used (with $m \leq k \leq p$), performing simultaneous variable selection and dimension reduction. The optimal solution can be found with a single branch and bound run, in which the objective function is given by the sum

of the first m eigenvalues of the covariance matrix. In this case we would have finally applied ordinary PCA on a subset of the variables. See Sect. 6 for an example of this.

4 Choice of the degree of sparsity

The choice of the number of principal components is a problem shared with classical PCA. A popular approach is the scree-test (Cattell 1966), in which the proportion of explained variance is plotted against the number of components, and an "elbow" is looked for in the graph. It is proposed in Kaiser (1960); Horn (1965) to retain only components explaining more than the average variance of the original variables. A comparison and complete overview of the methods, also for factor analysis, is given in Zwick and Velicer (1986).

On other hand, a different problem is posed by the choice of the number of variables to use in each sPC. In this section we will propose heuristic and formal methods to choose the degree of sparsity of each sPC; which we will explore in the example below.

Unlike other methods, we can explicitly choose the number of variables to be used in computing the sparse principal component (sPC), namely, k_i.

A possibility is to exploit the same idea of the scree-test: the objective function for a given sPC can be plotted against each possible value of k_i, deriving an "uphill plot". An example is given in Fig. 1. Further, one may better judge the trade off between variance and simplicity by putting on the y-axis the percentage of variance with respect to the maximum achievable (i.e., with no sparsity).

The variance of the sPC increases with the number of non-zero loadings k_i, but from some k_i on, the growth may flatten markedly. This elbow phenomenon can be used to choose k_i as the minimum number of non-zero loadings for which adding one variable does not give a significant contribution.

More formal criteria can be considered: one can choose k as the maximizer in q of $J(\Sigma, \beta_{(q),i}) - \rho(i) f(q)$, where $\beta_{(q),i}$ are the loadings of the ith sPC (given the previous) with degree of sparsity q, $\rho(i)$ is a penalty parameter, and $f(\cdot)$ is a strictly monotone function (for instance, the identity or the logarithmic function). This is in parallel with likelihood penalization methods. The first term is the proportion of variability contained in the component, which is penalized by a function of the proportion of variables used. This approach will favor sparser and more interpretable results for reasonable choices of the penalty parameter $\rho(i)$, which may be proportional to the average variance $\bar{\sigma}^2 = 1/m \sum_i \sigma_{ii}^2$. If $\rho(i)$ is taken to be constant with respect to i, the first principal components will be less sparse than the others. We thereby suggest to decrease the penalty parameter as i increases.

In summary, we use the criterion:

$$\max_q \; J(\Sigma, \beta_{(q),i}) - \frac{\log(q)\bar{\sigma}^2}{i+1} \tag{4}$$

Note that the choice of the degree of sparsity is to be made sequentially, following the natural ordering of the sparse components: for a different choice of k_1 the second sPC will be different, possibly leading to a different "optimal" k_2.

5 Simulations

We use a simulation experiment to validate the formal criterion for choosing the degree of sparsity given by (4) and to study the computational complexity of the branch and bound approach.

In order to see what may happen in real situations, Table 1 shows the time needed to find the first sPC with degree of sparsity k for different data sets. The CPU time is computed on batch jobs on a Sun XFire 4100 computer with AMD dual-core opteron and 8Gb RAM, using non-optimized R (R Development Core Team 2007) code. Use of optimized code in a low-level programming language like C++ should lead to much lower running times and possibility to apply the approach to larger covariance matrices. The time needed before finding the optimum depends not only on the number of variables, but also on the degree of sparsity requested.

The Boston data (Harrison and Rubinfeld 1978) contains $n = 506$ observations on $p = 18$ variables. The Papers data set (Collins et al. 2004) contains $n = 77$ observations and $p = 18$ variables. Pitprops data will be described and analyzed in detail in the next section. The Spambase data set has $n = 4{,}601$ observations and $p = 57$ variables, data and a full description are available on the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html) Dermatology data and Optdigits data are available from the same source, they have respectively $n = 358$ observations and $p = 34$ variables, and $n = 5{,}620$ and $p = 62$ variables. The last three data sets have been prepared by removing the class (categorical) variable, by merging test and training data where separated, and by removing two constant variables for the Optdigits data. The solutions for Pitprops, Boston, Papers and Dermatology data can be found quickly for any degree of sparseness. Certain solutions for the Spambase data set require long run times, and for the Optdigits data certain values are infeasible.

We now use a simulated setting. We generate three hidden factors

$$V_1 \sim N(0.290), \quad V_2 \sim N(0.300)$$
$$V_3 = -0.3V_1 + 0.925V_2 + \varepsilon, \quad \varepsilon \sim N(0, 1)$$

with V_1, V_2 and ε independent.

Then we generate $p = p_1 + p_2 + p_3$ observed variables as

$$X_j = V_1 + \varepsilon_j, \quad j = 1, \ldots, p_1$$
$$X_j = V_2 + \varepsilon_j, \quad j = p_1 + 1, \ldots, p_2$$
$$X_j = V_3 + \varepsilon_j, \quad j = p_1 + p_2 + 1, \ldots, p$$

with ε_j independent for all j.

The variance of the three underlying factors is 290, 300 and 283.8, respectively. This simulation setting is the same as Zou et al. (2004).

We now use the exact covariance matrix of $X_1, \ldots, X_p$ to evaluate the run time of the branch and bound approach. Table 2 shows the CPU time needed to compute the first sPC with degree of sparsity $p_2 + p_3$, for different values of p_1, p_2 and p_3. In our experience in this simulated setting the most important feature is p_1, with low times

Table 1 Time needed to find the first sPC with cardinality k on different true data sets

Data	p	k	Time
Boston	18	2	0.004 s
Boston	18	4	0.004 s
Boston	18	8	0.010 s
Boston	18	10	0.008 s
Boston	18	12	0.005 s
Papers	18	2	0.002 s
Papers	18	4	0.015 s
Papers	18	8	0.038 s
Papers	18	10	0.022 s
Papers	18	12	0.009 s
Pitprops	13	2	0.02 s
Pitprops	13	4	0.08 s
Pitprops	13	6	0.12 s
Pitprops	13	8	0.08 s
Pitprops	13	10	0.01 s
Dermatology	34	5	1.59 s
Dermatology	34	10	59.60 s
Dermatology	34	15	24.63 s
Dermatology	34	20	1.14 s
Dermatology	34	25	0.30 s
Dermatology	34	30	0.06 s
Spambase	57	5	0.04 s
Spambase	57	8	3.99 s
Spambase	57	10	38.86 s
Spambase	57	12	5 min 1 s
Spambase	57	15	39 min 19 s
Spambase	57	20	8 h 09 min 23 s
Spambase	57	25	7 h 57 min 23 s
Spambase	57	30	37 min 39 s
Spambase	57	33	3 min 50 s
Spambase	57	35	45.78 s
Spambase	57	40	2.13 s
Spambase	57	45	0.13 s
Optdigits	62	3	4.14 s
Optdigits	62	5	1 min 18 s
Optdigits	62	8	1 h 47 min 8 s
Optdigits	62	44	18.49 s
Optdigits	62	45	1.17 s
Optdigits	62	50	0.72 s
Optdigits	62	55	0.14 s

Table 2 Time needed to find the first sPC with cardinality $p_2 + p_3$

(p_1, p_2, p_3)	p	Time (s)
(5, 5, 10)	20	0.01
(15, 3, 2)	20	3.66
(15, 5, 5)	25	8.47
(10, 15, 5)	30	0.01
(17, 6, 7)	30	9.11
(5, 15, 20)	40	0.03
(20, 5, 15)	40	0.21
(22, 6, 14)	42	16.28
(50, 30, 20)	100	7.03
(50, 50, 50)	150	1.25

for $p_1 \leq p/2$. With small p_1 even large dimensional problems can be tackled. When $p_1 > p/2$ run times get much higher, and the problem infeasible even for moderate p. Only few cases in which $p_1 > p/2$ are reported in Table 2. This happens not only because $k = p - p_1$, but also because of the correlation structure that is brought about by high p_1, which makes the optimum closer to many other possible solutions. In summary, run times will depend on the data at hand, with the problem infeasible even in moderate dimensional situations when there is strong correlation among many of the variables.

Zou et al. (2004) use the exact covariance matrix to show that if only p_2 nonzero loadings are used, both their approach and SCoTLASS lead to the correct choice of giving nonzero weight to $X_{p_1+1}, \ldots, X_{p_2}$, with $p_1 = p_2 = 4$, $p_3 = 2$, while simple thresholding incorrectly includes variables from the third group. BB-sPCA with $k_1 = 4$ in the same setting correctly recovers the factor V_2 using X_5, X_6, X_7, X_8, and also gives the correct loading -0.5 to each. The first sPC explains 40.9% of the total variance, exactly like sPCA of Zou et al. (2004).

We now validate (4), used to choose the cardinality of the loadings. We generate n observations from the proposed model $B = 250$ times and record the proportion of times the degree of sparsity $p_2 + p_3$ is chosen using our proposed formal criterion.

While if only p_2 loadings are used the algorithm should recover the first sPC only using $X_{p_1+1}, \ldots, X_{p_2}$; the "right" cardinality in this setting should be $p_2 + p_3$, due to the strong correlation between V_2 and V_3. In fact, with $p_1 = p_2 = 4$ and $p_3 = 2$ using $k_1 = 6$ instead of $k_1 = 4$ leads to an increment of 15.4% in the explained variance, which in general would be desirable. If $p_1 = p_2 = p_3 = 5$, the increment is 33.8%, and so on.

Table 3 shows the results for different values of n and (p_1, p_2, p_3). With $n = \infty$ we mean that the exact covariance matrix was used. There is an effect of n/p. Nevertheless if the number of observations is high enough the right sparsity is chosen with satisfactory probability.

In order to verify the effect of the third (correlated) hidden factor, we simulated data in a different setting in which $V_3 \sim N(0, 283.8)$ independently of V_1 and V_2. In this case, the correct cardinality of the first sPC is p_2. Table 4 shows the proportion of times the proposed criterion leads to choose $k_1 = p_2$. When the hidden factors

Table 3 Proportion of times degree of sparsity $p_2 + p_3$ is chosen using criterion given by (4)

(p_1, p_2, p_3)	$n = 250$	$n = 500$	$n = 1{,}000$	$n = 5{,}000$	$n = \infty$
(2, 4, 4)	1.00	1.00	1.00	1.00	1.00
(4, 4, 2)	0.95	1.00	1.00	1.00	1.00
(4, 2, 4)	0.77	0.88	0.94	1.00	1.00
(5, 5, 5)	0.71	0.80	0.91	1.00	1.00
(5, 10, 5)	0.81	0.90	0.97	1.00	1.00
(10, 15, 5)	0.76	0.82	0.93	1.00	1.00
(5, 15, 10)	0.58	0.64	0.69	0.80	1.00

$B = 250$ iterations

Table 4 Proportion of times degree of sparsity p_2 is chosen using criterion given by (4), with independent hidden factors

(p_1, p_2, p_3)	$n = 250$	$n = 500$	$n = 1{,}000$	$n = 5{,}000$	$n = \infty$
(2, 4, 4)	1.00	1.00	1.00	1.00	1.00
(4, 4, 2)	1.00	1.00	1.00	1.00	1.00
(4, 2, 4)	0.99	1.00	1.00	1.00	1.00
(5, 5, 5)	1.00	1.00	1.00	1.00	1.00
(5, 10, 5)	0.98	1.00	1.00	1.00	1.00
(10, 15, 5)	0.88	0.96	1.00	1.00	1.00
(5, 15, 10)	0.92	0.99	1.00	1.00	1.00

$B = 250$ iterations

are uncorrelated the correct degree of sparsity can be recovered with much higher probability.

6 Application to Pitprops data

The Pitprops data was first used by Jeffers (1967) as an example of the difficulty of interpreting principal components. The data set has 180 observations and 13 standardized variables, so that Σ reduces to the correlation matrix. In Jeffers (1967) it is suggested to use the first six principal components.

Sparse principal components analysis can be applied to this data to enhance interpretability. SCoTLASS produced results in Table 5, while method in Zou et al. (2004) produced results in Table 6. Tables are taken from Zou et al. (2004). The variance and cumulative variance of the ordinary PCA solution is reported only in Table 5 for reasons of space.

In Table 7 we show BB-sPCA for the same degree of sparsity of the first two methods, with orthogonal coefficients and maximization of the variance of each component. When we set $k_5 = k_6 = 1$, there is no feasible solution. We show the solution which is closest to satisfaction the constraints: for each candidate optimum we compute $\beta_5' \beta_j$ for $j = 1, \ldots, 4$ and take the average of the four values. The reported solution is

Table 5 sPCA of Pitprops data, SCoTLASS

Variable	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	0.546	0.047	−0.087	0.066	−0.046	0.000
Length	0.568	0.000	−0.076	0.117	−0.081	0.000
Moist	0.000	0.641	−0.187	−0.127	0.009	0.017
Testsg	0.000	0.641	0.000	−0.139	0.000	0.000
Ovensg	0.000	0.000	0.457	0.000	−0.614	−0.562
Ringtop	0.000	0.356	0.348	0.000	0.000	−0.045
Ringbut	0.279	0.000	0.325	0.000	0.000	0.000
Bowmax	0.132	−0.007	0.000	−0.589	0.000	0.000
bowdist	0.376	0.000	0.000	0.000	0.000	0.065
Whorls	0.376	−0.065	0.000	−0.067	0.189	−0.065
Clear	0.000	0.000	0.000	0.000	−0.659	0.725
Knots	0.000	0.206	0.000	0.771	0.040	0.003
Diaknot	0.000	0.000	−0.718	0.013	−0.379	−0.384
Number of nonzero loadings	6	7	7	8	8	8
Variance (%)	27.2	16.4	14.8	9.4	7.1	7.9
Adjusted variance (%)	27.2	15.3	14.4	7.1	6.7	7.5
Cumulative adjusted variance (%)	27.2	42.5	56.9	64.0	70.7	78.2
Variance of PCA solution (%)	32.4	18.3	14.4	8.5	7.0	6.3
Cumulative variance of PCA solution (%)	32.4	50.7	65.2	73.7	80.7	87.0

Table 6 sPCA of Pitprops data, method Zou et al. (2004)

Variable	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	−0.477	0.000	0.000	0	0	0
Length	−0.476	0.000	0.000	0	0	0
Moist	0.000	0.785	0.000	0	0	0
Testsg	0.000	0.620	0.000	0	0	0
Ovensg	0.177	0.000	0.640	0	0	0
Ringtop	0.000	0.000	0.589	0	0	0
Ringbut	−0.250	0.000	0.492	0	0	0
Bowmax	−0.344	−0.021	0.000	0	0	0
Bowdist	−0.416	0.000	0.000	0	0	0
Whorls	−0.400	0.000	0.000	0	0	0
Clear	0.000	0.000	0.000	−1	0	0
Knots	0.000	0.013	0.000	0	−1	0
Diaknot	0.000	0.000	−0.015	0	0	1
Number of nonzero loadings	7	4	4	1	1	1
Variance (%)	28.0	14.4	15.0	7.7	7.7	7.7
Adjusted variance (%)	28.0	14.0	13.3	7.4	6.8	6.2
Cumulative adjusted variance (%)	28.0	42.0	55.3	62.7	69.5	75.8

Table 7 BB-sPCA of Pitprops data, maximization of variance

Variable	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	−0.444	0.226	0.205	0.000	−0.093	0.000
Length	−0.453	0.000	−0.146	0.288	−0.235	0.216
Moist	0.000	0.604	0.000	0.167	0.222	−0.103
Testsg	0.000	0.623	0.000	0.000	0.276	0.358
Ovensg	0.000	0.000	−0.597	0.000	0.000	0.000
Ringtop	0.000	0.290	−0.182	−0.440	0.000	−0.479
Ringbut	−0.379	0.000	−0.088	−0.336	0.000	−0.469
Bowmax	−0.341	−0.154	0.000	0.000	0.294	0.000
Bowdist	−0.403	0.000	0.000	0.234	0.000	0.000
Whorls	−0.418	−0.114	0.000	−0.234	0.113	0.190
Clear	0.000	0.000	0.000	0.681	0.084	−0.559
Knots	0.000	0.271	0.018	0.000	−0.839	0.000
Diaknot	0.000	0.000	0.734	−0.092	0.000	−0.132
Number of nonzero loadings	6	7	7	8	8	8
Variance (%)	29.0	17.3	15.8	9.9	7.8	7.0
Adjusted variance (%)	29.0	16.7	10.6	7.5	6.5	2.1
Cumulative adjusted variance (%)	29.0	45.7	56.4	63.9	70.3	72.4
Topdiam	−0.423	0.000	0.000	0	0	0
Length	−0.430	0.000	0.000	0	0	0
Moist	0.000	0.676	0.000	0	0	0
Testsg	0.000	0.662	0.000	0	0	0
Ovensg	0.000	0.000	0.000	1	0	0
Ringtop	−0.268	0.000	0.000	0	0	0
Ringbut	−0.403	0.000	0.000	0	0	0
Bowmax	−0.313	0.000	0.558	0	0	0
Bowdist	−0.379	0.000	0.000	0	0	0
Whorls	−0.400	0.000	0.187	0	0	0
Clear	0.000	0.182	0.000	0	1	0
Knots	0.000	0.267	−0.679	0	0	0
Diaknot	0.000	0.000	−0.438	0	0	1
Number of nonzero loadings	7	4	4	1	1	1
Variance (%)	30.7	15.3	13.4	7.7	7.7	7.7
Adjusted variance (%)	30.7	15.0	7.5	7.5	7.0	4.9
Cumulative adjusted variance (%)	30.7	45.8	53.3	60.7	67.7	72.6

the one with average closest to zero. The same is done to compute β_6. In Table 9 we change the constraint and ask for uncorrelated components. The same comment as before applies for the fourth, fifth and sixth components when same sparsity as Zou et al. (2004) is requested. This is a good example of a case in which too strong sparsity is imposed, leading to unavailability of solutions satisfying orthogonality/

uncorrelation. In all other cases, in this example, a solution exists and the branch and bound can be used to find the optimal.

By comparing Tables 7 and 9 the effect of changing the constraints can be appreciated. When loadings are forced to be orthogonal the solutions are more easily interpretable (there is less overlap among the components), and the objective function may be a little higher. This comment applies when the variance is chosen as objective function, as in Tables 7 and 9, and in fact it is not true for the adjusted variance (which is not even maximized in Tables 7 and 9).

Orthogonal optima may be better than uncorrelated optima with respect to these criteria. On other hand uncorrelation of the components is sometimes of important practical use for graphical purposes and whenever the components are used simultaneously.

Information loss in using a sparse solution is in this example minimal, while at most 8 variables out of 13 are used by sparse principal components.

In Table 8 we maximize the adjusted variance. In all but one case (the sixth sPC with $k_6 = 8$) we manage to achieve higher objective functions than SCoTLASS. When compared with sPCA, BB-sPCA achieves higher objective functions in the first three sPC, even in presence of the further orthogonality constraint. In this example our approach to sPCA leads in conclusion to more compression of the information, at least in the first axes: for the first five axes the total variance/adjusted variance is always higher than the other methods, with the same degree of sparsity.

It is worth also noticing that in all cases both the variance and adjusted variance follow the right non-decreasing order, while this happens only for the adjusted variance with the other methods.

When coming to the interpretation of the axes, our methods produce the same or less overlap among components (especially when maximizing the adjusted variance). The second sPC with $k_2 = 4$ and the fourth with $k_4 = 1$ in Table 7 put zero weight to all of the previously used variables.

In Table 9 we change the constraint and ask for uncorrelated components. Even if the new variables are now uncorrelated, the objective functions are very close to the results of SCoTLASS and sPCA, and even higher in the first components. Note that not all of the new variables can be made uncorrelated: once again we turn to the solution closest to satisfaction of the constraints when $k_4 = k_5 = k_6 = 1$. Loadings are seldom orthogonal, but always very close to orthogonality.

To give an idea of the efficiency of the branch and bound algorithm we note that at most only 27% of the possible groupings of k variables were actually explored by the algorithm, in these examples. Even less are expected when the covariance matrix is used instead of the correlation matrix.

Figure 1 shows the uphill plot in which the variability of the first component is plotted as a function of k. Such test would probably lead to agreement with the previous methods by choosing k as 6 or 7. In Tables 10 and 11 results for degree of sparsity of each component chosen as the maximizer of (4) are shown. Note that the first three components can explain almost the same amount of variance as the first three of sPCA, using the same number of variables in the first component and more sparsity in the second and third component.

Not surprisingly all the non-zero loadings are far from zero, and additionally our loadings are orthogonal so that interpretation is easier.

Table 8 BB-sPCA of Pitprops data, maximization of adjusted variance

Variable	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	−0.444	0.235	−0.179	0.000	0.000	0.000
Length	−0.453	0.000	−0.179	0.000	−0.076	0.121
Moist	0.000	0.602	0.000	−0.118	0.304	0.148
Testsg	0.000	0.617	0.000	0.000	0.264	−0.070
Ovensg	0.000	0.000	0.477	0.000	0.000	0.000
Ringtop	0.000	0.268	0.439	0.000	−0.304	0.718
Ringbut	−0.379	0.000	0.424	0.139	−0.157	0.000
Bowmax	−0.341	−0.160	0.000	−0.265	0.274	−0.158
Bowdist	−0.403	0.000	0.000	−0.172	0.000	0.000
Whorls	−0.418	−0.118	0.000	0.256	0.000	−0.160
Clear	0.000	0.000	0.000	−0.874	−0.313	0.000
Knots	0.000	0.299	−0.254	0.195	−0.736	−0.103
Diaknot	0.000	0.000	−0.519	0.018	0.000	0.618
Number of nonzero loadings	6	7	7	8	8	8
Variance (%)	29.0	17.3	15.7	8.9	7.0	6.5
Adjusted variance (%)	29.0	17.2	15.3	8.8	6.8	6.4
Cumulative adjusted variance (%)	29.0	46.2	61.5	70.3	77.1	83.5
Topdiam	−0.423	0.000	0.000	0	0	0
Length	−0.430	0.000	0.000	0	0	0
Moist	0.000	0.676	0.000	0	0	0
Testsg	0.000	0.662	0.000	0	0	0
Ovensg	0.000	0.000	−0.659	0	0	0
Ringtop	−0.268	0.000	−0.435	0	0	0
Ringbut	−0.403	0.000	0.000	0	0	0
Bowmax	−0.313	0.000	0.000	0	0	0
Bowdist	−0.379	0.000	0.308	0	0	0
Whorls	−0.400	0.000	0.000	0	0	0
Clear	0.000	0.182	0.000	1	0	0
Knots	0.000	0.267	0.000	0	1	0
Diaknot	0.000	0.000	0.530	0	0	1
Number of nonzero loadings	7	4	4	1	1	1
Variance (%)	30.7	15.3	11.1	7.7	7.7	7.7
Adjusted variance (%)	30.7	15.0	10.9	7.1	5.9	4.1
Cumulative adjusted variance (%)	30.7	45.8	56.7	63.8	69.7	73.8

Table 12 shows loadings for uncorrelated components with automatically chosen sparsity, giving rise to the same comments.

Finally, suppose we want to perform a simultaneous variable selection and dimension reduction. This can be done by forcing the six principal components to use *the*

Table 9 BB-sPCA of Pitprops data, uncorrelated components [same sparsity as Zou et al. (2004) and SCoTLASS]

Variable	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	−0.423	0.000	0.000	0	0	0
Length	−0.430	0.000	0.431	0	0	0
Moist	0.000	0.654	0.000	0	0	0
Testsg	0.000	0.635	0.000	0	0	0
Ovensg	0.000	0.000	−0.682	0	0	0
Ringtop	−0.268	0.000	−0.551	0	0	0
Ringbut	−0.403	0.000	0.000	0	0	0
Bowmax	−0.313	0.000	0.000	0	0	0
Bowdist	−0.379	0.000	0.000	0	0	0
Whorls	−0.400	−0.222	0.000	0	0	0
Clear	0.000	0.000	0.000	1	0	0
Knots	0.000	0.345	0.000	0	1	0
Diaknot	0.000	0.000	0.214	0	0	1
Number of nonzero loadings	7	4	4	1	1	1
Variance (%)	30.7	15.3	10.5	7.7	7.7	7.7
Adjusted variance (%)	30.7	15.3	10.5	7.4	5.5	5.2
Cumulative adjusted variance (%)	30.7	46.0	56.6	64.0	69.4	74.6
Topdiam	−0.444	0.097	−0.285	0.000	0.000	−0.120
Length	−0.453	0.000	−0.301	0.000	0.000	−0.166
Moist	0.000	0.585	0.000	−0.078	0.272	0.373
Testsg	0.000	0.572	0.210	0.000	0.321	0.000
Ovensg	0.000	0.000	0.526	0.000	0.323	−0.591
Ringtop	0.000	0.129	0.456	0.000	−0.364	−0.116
Ringbut	−0.378	0.000	0.300	0.076	−0.275	0.000
Bowmax	−0.341	−0.328	0.000	−0.222	0.146	0.000
Bowdist	−0.403	0.000	0.000	−0.101	0.000	0.000
Whorls	−0.418	0.000	0.000	0.279	0.000	0.194
Clear	0.000	0.000	0.000	−0.880	−0.238	−0.145
Knots	0.000	0.394	0.000	0.169	−0.658	0.000
Diaknot	0.000	0.202	−0.456	0.220	0.000	−0.630
Number of nonzero loadings	6	7	7	8	8	8
Variance (%)	29.0	16.3	14.5	8.6	6.7	6.2
Adjusted variance (%)	29.0	16.3	14.5	8.6	6.7	6.2
Cumulative adjusted variance (%)	29.0	45.3	59.8	68.4	75.1	81.3

same variables, thereby changing the problem to

$$\begin{cases} \sum_{j=1}^{6} \max_{\beta_j} \beta_j' \Sigma \beta_j \\ \beta_{j_1}' \beta_{j_2} = I(j_1 = j_2) \\ |\beta_1 \odot \cdots \odot \beta_6|_0 \leq k \end{cases} \tag{5}$$

Table 10 BB-sPCA of Pitprops data, maximization of variance, automatically chosen sparsity

Variable	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	−0.423	0.000	0.000	0.000	0	0
Length	−0.430	0.000	0.000	0.000	0	0
Moist	0.000	0.707	0.000	0.000	0	0
Testsg	0.000	0.707	0.000	0.000	0	0
Ovensg	0.000	0.000	0.000	0.707	0	0
Ringtop	−0.268	0.000	0.488	0.000	0	0
Ringbut	−0.403	0.000	0.000	0.000	0	0
Bowmax	−0.313	0.000	−0.417	0.000	0	0
Bowdist	−0.379	0.000	0.000	0.000	0	1
Whorls	−0.400	0.000	0.000	0.000	0	0
Clear	0.000	0.000	0.000	0.000	1	0
Knots	0.000	0.000	0.766	0.000	0	0
Diaknot	0.000	0.000	0.000	−0.707	0	0
Number of nonzero loadings	7	2	3	2	1	1
Variance (%)	30.7	14.5	9.3	9.3	7.7	7.7
Adjusted variance (%)	30.7	13.9	8.2	9.0	7.5	3.1
Cumulative adjusted variance (%)	30.7	44.6	52.8	61.8	69.3	72.4

Table 11 BB-sPCA of Pitprops data, maximization of adjusted variance, automatically chosen sparsity

Variable	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	−0.423	0.000	0.000	0	0	0.000
Length	−0.430	0.000	−0.283	0	0	0.000
Moist	0.000	0.707	0.000	0	0	0.000
Testsg	0.000	0.707	0.000	0	0	0.000
Ovensg	0.000	0.000	0.600	0	0	0.704
Ringtop	−0.268	0.000	0.455	0	0	0.000
Ringbut	−0.403	0.000	0.000	0	0	0.000
Bowmax	−0.313	0.000	0.000	0	0	0.000
Bowdist	−0.379	0.000	0.000	0	0	0.000
Whorls	−0.400	0.000	0.000	0	0	0.000
Clear	0.000	0.000	0.000	1	0	0.000
Knots	0.000	0.000	0.000	0	1	0.000
Diaknot	0.000	0.000	−0.594	0	0	0.710
Number of nonzero loadings	7	2	4	1	1	2
Variance (%)	30.7	14.5	11.8	7.7	7.7	6.1
Adjusted variance (%)	30.7	13.9	11.7	7.5	6.8	5.9
Cumulative adjusted variance (%)	30.7	44.6	56.3	63.8	70.6	76.5

Table 12 BB-sPCA of Pitprops data, uncorrelated components, automatically chosen sparsity

Variable	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	−0.423	0.000	0.306	0.000	0.000	0.000
Length	−0.430	0.000	0.000	0.000	−0.202	0.000
Moist	0.000	−0.654	0.000	0.000	0.221	0.260
Testsg	0.000	−0.635	0.000	0.000	0.000	0.271
Ovensg	0.000	0.000	−0.867	0.000	−0.331	0.219
Ringtop	−0.268	0.000	−0.392	0.000	0.000	−0.374
Ringbut	−0.403	0.000	0.000	0.000	0.000	−0.255
Bowmax	−0.313	0.000	0.000	0.000	0.000	0.219
Bowdist	−0.379	0.000	0.000	0.000	0.000	0.000
Whorls	−0.400	0.222	0.000	0.093	0.000	0.000
Clear	0.000	0.000	0.000	−0.918	−0.223	−0.289
Knots	0.000	−0.345	0.000	0.328	0.000	−0.689
Diaknot	0.000	0.000	0.000	0.200	−0.867	0.000
Number of nonzero loadings	7	4	3	4	5	8
Variance (%)	30.7	15.3	9.5	7.9	6.5	6.9
Adjusted variance (%)	30.7	15.3	9.5	7.9	6.5	6.9
Cumulative adjusted variance (%)	30.7	46.0	55.6	63.5	70.0	76.9

where $I(C)$ is the indicator function of condition C and $\odot$ denotes elementwise product of two vectors. In words, we use six unit length orthogonal loading vectors which give zero weight to the same $p-k$ variables. In practice, the objective is now the sum of the largest to sixth largest eigenvalues of a covariance matrix made of k variables. If we use only $k = 10$ out of the 13 variables we can still explain 79% of the variance, and 61% can be explained by using only $k = 8$ variables.

7 Conclusions

Sparse principal component analysis can be seen as a dimension reduction technique that aims at interpretable (thus, sparse while orthogonal in the loadings or uncorrelated in the components) solutions.

On other hand, a parallel goal of sPCA is simultaneous feature extraction and selection. PCA falls into the category of feature extraction techniques: it builds new variables carrying a large part of the global variance. On other hand, feature selection techniques (Guyon and Elisseeff 2003; Miller 1990 etc.) find an appropriate subset of the original variables to represent the data. It may be desirable to combine feature selection and extraction, and a possibility is given by sparseness of the loadings in PCA. Additionally, we have seen for instance how to select the best subset of $k \leq p$ variables in order to extract $m \leq k$ linear combinations of those that carry the largest possible proportion of global information.

We showed a branch and bound approach as an exact alternative to popular approximate (but faster) methods.

Our approach is flexible and all its features (number of non-zero loadings, properties of the resulting components, objective function) can be chosen in advance.

We gave some guidelines to such choices. In real data applications different combinations can be tried and the sparse solution achieving exact orthogonality/uncorrelation and closest to uncorrelation/orthogonality adopted.

We remark once again that branch and bound techniques have exponential complexity, and thus are unsuitable for large-scale problems. Along the lines of Moghaddam et al. (2006), we can say that discrete algorithms are a complementary tool to continuous ones. In software implementation we may suggest to put a time limit to the branch and bound, whose current optimum can be used a starting solution for continuous methods if convergence was not achieved before the time limit.

In certain situations the algorithm is surprisingly fast, because of the availability of efficient maximization procedures for the simple problem at each node of the branch and bound algorithm. Still, even if our approach can efficiently handle large n data matrices, unfortunately when p is big the number of nodes can get too large to allow for an exploration of the entire tree. Due to the nature of the problem, we actually expect the method to be applicable also in high-dimensional situations when the covariance matrix is used and few variables have much higher variance than the remaining. If this is not the case, it still happens in few high-dimensional problems (like in DNA Microarray data analysis) that only less than 1% of the variables are expected to finally enter into the model, so we can moreover suggest a preliminary variable selection. The algorithm will be finally applied only to a subset of prospective relevant variables. Another possibility is to split the variables into groups, perform a BB-sPCA on the groups; and finally aggregate and perform a final BB-sPCA on the extracted variables. The results are not guaranteed to be optimal but certainly a genuine sPCA. The only viable method for high-dimensional datasets at the moment seems to be the sPCA of Zou et al. (2004), which furthermore does not rely on actually computing and storing the covariance matrix. Another promising approach is provided in d'Aspremont et al. (2007).

Acknowledgments The author is grateful to an Associate editor and two referees for constructive comments that lead to improvement of the paper and of the presentation. Acknowledgments go also to Nicola Apollonio for interesting discussions about the mathematical background.

References

Cadima J, Jolliffe IT (1995) Loadings and correlations in the interpretation of principal components. J Appl Stat 22:203–214

Cattell RB (1966) The meaning and strategic use of factor analysis. In: Handbook of multivariate experimental psychology. Springer, Heidelberg

Chatfield C, Collins AJ (1980) Introduction to multivariate analysis. Chapman and Hall, London

Coleman TF, Li Y (1994) On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds. Math Program 67:189–224

Collins J, Jaufer D, Vlachos P, Butler B, Suguru I (2004) Detecting collaborations in text comparing the authors' rhetorical language choices in The Federalist Papers. Comput Hum 38:15–36

d'Aspremont A, El Ghaoui L, Jordan M, Lanckriet GRG (2007) A direct formulation for sparse PCA using semidefinite programming. SIAM Rev 49:434–448

Gill PE, Murray W, Wright MH (1981) Practical optimization. Academic Press, London

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3: 1157–1182
Hand DJ (1981) Branch and bound in statistical data analysis. Statistician 30
Harrison D, Rubinfeld DL (1978) Hedonic prices and the demand for clean air. J Environ Econom Manage 5:81–102
Horn JL (1965) A rationale and test for the number of factors in factor analysis. Psychometrika 30:179–185
Jeffers J (1967) Two case studies in the application of principal components. Appl Stat 16:225–236
Jolliffe I (2002) Principal component analysis. Springer, Heidelberg
Jolliffe IT (1995) Rotation of principal components: choice of normalization constraints. J Appl Stat 22: 29–35
Jolliffe IT, Trendafilov NT, Uddin M (2003) A modified principal component technique based on the lasso. J Comput Graph Stat 12:531–547
Kaiser HF (1960) The application of electronic computers to factor analysis. Educational Psychol Measurement 20:141–151
Miller A (1990) Subset selection in regression. Chapman and Hall, London
Moghaddam B, Weiss Y, Avidan S (2006) Spectral bounds for sparse PCA: exact and greedy algorithms. Adv Neural Inf Process Syst 18
R Development Core Team (2007) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria
Trendafilov NT, Jolliffe IT (2006) Projected gradient approach to the numerical solution of the SCoTLASS. Comput Stat Data Anal 50:242–253
Wilkinson JH (1965) The algebraic eigenvalue problem. Oxford University Press, Oxford
Zou H, Hastie T (2005) Regression shrinkage and selection via the elastic net. J Royal Statistical Soc (Ser. B) 67:301–320
Zou H, Hastie T, Tibshirani R (2004) Sparse principal components analysis. Technical Report, Department of Statistics, Stanford University, USA
Zwick WR, Velicer WF (1986) Comparison of five rules for determining the number of components to retain. Psychol Bull 99:432–442