Sparse Principal Component Analysis

Hui Zou[*], Trevor Hastie[†], Robert Tibshirani[‡]

April 26, 2004

Abstract

Principal component analysis (PCA) is widely used in data processing and dimensionality reduction. However, PCA suffers from the fact that each principal component is a linear combination of all the original variables, thus it is often difficult to interpret the results. We introduce a new method called sparse principal component analysis (SPCA) using the *lasso* (*elastic net*) to produce modified principal components with sparse loadings. We show that PCA can be formulated as a regression-type optimization problem, then sparse loadings are obtained by imposing the lasso (elastic net) constraint on the regression coefficients. Efficient algorithms are proposed to realize SPCA for both regular multivariate data and gene expression arrays. We also give a new formula to compute the total variance of modified principal components. As illustrations, SPCA is applied to real and simulated data, and the results are encouraging.

Keywords: multivariate analysis, gene expression arrays, elastic net, lasso, singular value decomposition, thresholding

[*]Hui Zou is a Ph.D student in the Department of Statistics at Stanford University, Stanford, CA 94305. Email: hzou@stat.stanford.edu.

[†]Trevor Hastie is Professor, Department of Statistics and Department of Health Research & Policy, Stanford University, Stanford, CA 94305. Email: hastie@stat.stanford.edu.

[‡]Robert Tibshirani is Professor, Department of Health Research & Policy and Department of Statistics, Stanford University, Stanford, CA 94305. Email: tibs@stat.stanford.edu.

1 Introduction

Principal component analysis (PCA) (Jolliffe 1986) is a popular data processing and dimension reduction technique . As an un-supervised learning method, PCA has numerous applications such as handwritten zip code classification (Hastie et al. 2001) and human face recognition (Hancock et al. 1996). Recently PCA has been used in gene expression data analysis (Misra et al. 2002). Hastie et al. (2000) propose the so-called *Gene Shaving* techniques using PCA to cluster high variable and coherent genes in microarray data.

PCA seeks the linear combinations of the original variables such that the derived variables capture maximal variance. PCA can be done via the singular value decomposition (SVD) of the data matrix. In detail, let the data $\mathbf{X}$ be a $n \times p$ matrix, where n and p are the number of observations and the number of variables, respectively. Without loss of generality, assume the column means of $\mathbf{X}$ are all 0. Suppose we have the SVD of $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{1}$$

where T means transpose. $\mathbf{U}$ are the principal components (PCs) of unit length, and the columns of $\mathbf{V}$ are the corresponding loadings of the principal components. The variance of the ith PC is $D_{i,i}^2$. In gene expression data the PCs $\mathbf{U}$ are called the *eigen-arrays* and $\mathbf{V}$ are the *eigen-genes* (Alter et al. 2000). Usually the first q ($q \ll p$) PCs are chosen to represent the data, thus a great dimensionality reduction is achieved.

The success of PCA is due to the following two important optimal properties:

1. principal components sequentially capture the maximum variability among $\mathbf{X}$, thus guaranteeing minimal information loss;

2. principal components are uncorrelated, so we can talk about one principal component without

referring to others.

However, PCA also has an obvious drawback, i.e., each PC is a linear combination of all p variables and the loadings are typically nonzero. This makes it often difficult to interpret the derived PCs. Rotation techniques are commonly used to help practitioners to interpret principal components (Jolliffe 1995). Vines (2000) considered simple principal components by restricting the loadings to take values from a small set of allowable integers such as 0, 1 and -1.

We feel it is desirable not only to achieve the dimensionality reduction but also to reduce the size of explicitly used variables. An ad hoc way is to artificially set the loadings with absolute values smaller than a threshold to zero. This informal thresholding approach is frequently used in practice but can be potentially misleading in various respects (Cadima & Jolliffe 1995). McCabe (1984) presented an alternative to PCA which found a subset of *principal variables*. Jolliffe & Uddin (2003) introduced SCoTLASS to get modified principal components with possible zero loadings.

Recall the same interpretation issue arising in multiple linear regression, where the response is predicted by a linear combination of the predictors. Interpretable models are obtained via variable selection. The *lasso* (Tibshirani 1996) is a promising variable selection technique, simultaneously producing accurate and sparse models. Zou & Hastie (2003) propose the *elastic net*, a generalization of the lasso, to further improve upon the lasso. In this paper we introduce a new approach to get modified PCs with sparse loadings, which we call sparse principal component analysis (SPCA). SPCA is built on the fact that PCA can be written as a regression-type optimization problem, thus the lasso (elastic net) can be directly integrated into the regression criterion such that the resulting modified PCA produces sparse loadings.

In the next section we briefly review the lasso and the elastic net. The method details of SPCA are presented in Section 3. We first discuss a direct sparse approximation approach via the elastic net, which is a useful exploratory tool. We then show that finding the loadings of principal

components can be reformulated as estimating coefficients in a regression-type optimization problem. Thus by imposing the lasso (elastic net) constraint on the coefficients, we derive the modified principal components with sparse loadings. An efficient algorithm is proposed to realize SPCA. We also give a new formula, which justifies the correlation effects, to calculate the total variance of modified principal components. In Section 4 we consider a special case of the SPCA algorithm to efficiently handle gene expression arrays. The proposed methodology is illustrated by using real data and simulation examples in Section 5. Discussions are in Section 6. The paper ends up with an appendix summarizing technical details.

2 The Lasso and The Elastic Net

Consider the linear regression model. Suppose the data set has n observations with p predictors. Let $Y = (y_1, \ldots, y_n)^T$ be the response and $\mathbf{X}_j = (x_{1j}, \ldots, x_{nj})^T, i = 1, \ldots, p$ are the predictors. After a location transformation we can assume all $\mathbf{X}_j$ and Y are centered.

The lasso is a penalized least squares method, imposing a constraint on the L_1 norm of the regression coefficients. Thus the lasso estimates $\hat{\beta}_{lasso}$ are obtained by minimizing the lasso criterion

$$\hat{\beta}_{lasso} = \arg\min_{\beta} \left| Y - \sum_{j=1}^{p} \mathbf{X}_j \beta_j \right|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \,, \tag{2}$$

where λ is a non-negative value. The lasso was originally solved by quadratic programming (Tibshirani 1996). Efron et al. (2004) proved that the lasso estimates as a function of λ are piecewise linear, and proposed an algorithm called LARS to efficiently solve the whole lasso solution path in the same order of computations as a single least squares fit.

The lasso continuously shrinks the coefficients toward zero, thus gaining its prediction accuracy via the bias variance trade-off. Moreover, due to the nature of the L_1 penalty, some coefficients

will be shrunk to exact zero if λ_1 is large enough. Therefore the lasso simultaneously produces an accurate and sparse model, which makes it a favorable variable selection method. However, the lasso has several limitations as pointed out in Zou & Hastie (2003). The most relevant one to this work is that the number of selected variables by the lasso is limited by the number of observations. For example, if applied to the microarray data where there are thousands of predictors (genes) ($p > 1000$) with less than 100 samples ($n < 100$), the lasso can only select at most n genes, which is clearly unsatisfactory.

The elastic net (Zou & Hastie 2003) generalizes the lasso to overcome its drawbacks, while enjoying the similar optimal properties. For any non-negative λ_1 and λ_2, the elastic net estimates $\hat{\beta}_{en}$ are given as follows

$$\hat{\beta}_{en} = (1 + \lambda_2) \arg\min_{\beta} \left| Y - \sum_{j=1}^{p} \mathbf{X}_j \beta_j \right|^2 + \lambda_2 \sum_{j=1}^{p} |\beta_j|^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| \,. \tag{3}$$

Hence the elastic net penalty is a convex combination of ridge penalty and the lasso penalty . Obviously, the lasso is a special case of the elastic net with $\lambda_2 = 0$. Given a fixed λ_2, the LARS-EN algorithm (Zou & Hastie 2003) efficiently solves the elastic net problem for all λ_1 with the computation cost as a single least squares fit. When $p > n$, we choose some $\lambda_2 > 0$. Then the elastic net can potentially include all variables in the fitted model, so the limitation of the lasso is removed. An additional benefit offered by the elastic net is its grouping effect, that is, the elastic net tends to select a group of highly correlated variables once one variable among them is selected. In contrast, the lasso tends to select only one out of the grouped variables and does not care which one is in the final model. Zou & Hastie (2003) compare the elastic net with the lasso and discuss the application of the elastic net as a gene selection method in microarray analysis.

3 Motivation and Method Details

In both lasso and elastic net, the sparse coefficients are a direct consequence of the L_1 penalty, not depending on the squared error loss function. Jolliffe & Uddin (2003) proposed SCoTLASS by directly putting the L_1 constraint in PCA to get sparse loadings. SCoTLASS successively maximizes the variance

$$a_k^T(\mathbf{X}^T\mathbf{X})a_k \tag{4}$$

subject to

$$a_k^T a_k = 1 \quad \text{and (for} \quad k \geq 2) \quad a_h^T a_k = 0, \quad h < k; \tag{5}$$

and the extra constraints

$$\sum_{j=1}^{p} |a_{k,j}| \leq t \tag{6}$$

for some tuning parameter t. Although sufficiently small t yields some exact zero loadings, SCoTLASS seems to lack of a guidance to choose an appropriate t value. One might try several t values, but the high computational cost of SCoTLASS makes it an impractical solution. The high computational cost is due to the fact that SCoTLASS is not a convex optimization problem. Moreover, the examples in Jolliffe & Uddin (2003) show that the obtained loadings by SCoTLASS are not sparse enough when requiring a high percentage of explained variance.

We consider a different approach to modify PCA, which can more directly make good use of the lasso. In light of the success of the lasso (elastic net) in regression, we state our strategy

> *We seek a regression optimization framework in which PCA is done exactly. In addition, the regression framework should allow a direct modification by using the lasso (elastic net) penalty such that the derived loadings are sparse.*

3.1 Direct sparse approximations

We first discuss a simple regression approach to PCA. Observe that each PC is a linear combination of the p variables, thus its loadings can be recovered by regressing the PC on the p variables.

Theorem 1 *$\forall i$, denote $Y_i = \mathbf{U}_i D_i$. Y_i is the i-th principal component. $\forall\ \lambda > 0$, suppose $\hat{\beta}_{ridge}$ is the ridge estimates given by*

$$\hat{\beta}_{ridge} = \arg\min_{\beta} |Y_i - \mathbf{X}\beta|^2 + \lambda |\beta|^2 . \tag{7}$$

Let $\hat{v} = \frac{\hat{\beta}_{ridge}}{|\hat{\beta}_{ridge}|}$, then $\hat{v} = \mathbf{V}_i$.

The theme of this simple theorem is to show the connection between PCA and a regression method is possible. Regressing PCs on variables was discussed in Cadima & Jolliffe (1995), where they focused on approximating PCs by a subset of k variables. We extend it to a more general ridge regression in order to handle all kinds of data, especially the gene expression data. Obviously when $n > p$ and $\mathbf{X}$ is a full rank matrix, the theorem does not require a positive λ. Note that if $p > n$ and $\lambda = 0$, ordinary multiple regression has no unique solution that is exactly $\mathbf{V}_i$. The same story happens when $n > p$ and $\mathbf{X}$ is not a full rank matrix. However, PCA always gives a unique solution in all situations. As shown in theorem 1, this discrepancy is eliminated by the positive ridge penalty ($\lambda |\beta|^2$). Note that after normalization the coefficients are independent of λ, therefore the ridge penalty is not used to penalize the regression coefficients but to ensure the reconstruction of principal components. Hence we keep the ridge penalty term throughout this paper.

Now let us add the L_1 penalty to (7) and consider the following optimization problem

$$\hat{\beta} = \arg\min_{\beta} |Y_i - \mathbf{X}\beta|^2 + \lambda |\beta|^2 + \lambda_1 |\beta|_1 . \tag{8}$$

We call $\hat{V}_i = \frac{\hat{\beta}}{|\hat{\beta}|}$ an approximation to $\mathbf{V}_i$, and $\mathbf{X}\hat{\mathbf{V}}_i$ the ith approximated principal component. (8) is called *naive* elastic net (Zou & Hastie 2003) which differs from the elastic net by a scaling factor $(1+\lambda)$. Since we are using the normalized fitted coefficients, the scaling factor does not affect $\hat{V}_i$. Clearly, large enough λ_1 gives a sparse $\hat{\beta}$, hence a sparse $\hat{V}_i$. Given a fixed λ, (8) is efficiently solved for all λ_1 by using the LARS-EN algorithm (Zou & Hastie 2003). Thus we can flexibly choose a sparse approximation to the ith principal component.

3.2 Sparse principal components based on SPCA criterion

Theorem 1 depends on the results of PCA, so it is not a *genuine* alternative. However, it can be used in a two-stage exploratory analysis: first perform PCA, then use (8) to find suitable sparse approximations.

We now present a "self-contained" regression-type criterion to derive PCs. We first consider the leading principal component.

Theorem 2 *Let $\mathbf{X}_i$ denote the ith row vector of the matrix $\mathbf{X}$. For any $\lambda > 0$, let*

$$\begin{aligned}(\hat{\alpha}, \hat{\beta}) &= \arg\min_{\alpha,\beta} \sum_{i=1}^{n} \left|\mathbf{X}_i - \alpha\beta^T\mathbf{X}_i\right|^2 + \lambda|\beta|^2 \\ &\quad \textit{subject to} \quad |\alpha|^2 = 1.\end{aligned} \tag{9}$$

Then $\hat{\beta} \propto \mathbf{V}_1$.

The next theorem extends theorem 2 to derive the whole sequence of PCs.

Theorem 3 *Suppose we are considering the first k principal components. Let α and β be $p \times k$ matrices. $\mathbf{X}_i$ denote the i-th row vector of the matrix $\mathbf{X}$. For any $\lambda > 0$, let*

$$(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha,\beta} \sum_{i=1}^{n} \left|\mathbf{X}_i - \alpha\beta^T\mathbf{X}_i\right|^2 + \lambda\sum_{j=1}^{k}|\beta_j|^2 \tag{10}$$

$$\textit{subject to} \quad \alpha^T\alpha = I_k.$$

Then $\hat{\beta}_i \propto \mathbf{V}_i$ *for* $i = 1, 2, \ldots, k$.

Theorem 3 effectively transforms the PCA problem to a regression-type problem. The critical element is the object function $\sum_{i=1}^{n} \left|\mathbf{X}_i - \alpha\beta^T\mathbf{X}_i\right|^2$. If we restrict $\beta = \alpha$, then $\sum_{i=1}^{n} \left|\mathbf{X}_i - \alpha\beta^T\mathbf{X}_i\right|^2 = \sum_{i=1}^{n} \left|\mathbf{X}_i - \alpha\alpha^T\mathbf{X}_i\right|^2$, whose minimizer under the orthonormal constraint on α is exactly the first k loading vectors of ordinary PCA. This is actually an alternative derivation of PCA other than the maximizing variance approach, e.g. Hastie et al. (2001). Theorem 3 shows that we can still have exact PCA while relaxing the restriction $\beta = \alpha$ and adding the ridge penalty term. As can be seen later, these generalizations enable us to flexibly modify PCA.

To obtain sparse loadings, we add the lasso penalty into the criterion (10) and consider the following optimization problem

$$\begin{aligned} (\hat{\alpha}, \hat{\beta}) &= \arg\min_{\alpha,\beta} \sum_{i=1}^{n} \left|\mathbf{X}_i - \alpha\beta^T\mathbf{X}_i\right|^2 + \lambda \sum_{j=1}^{k} |\beta_j|^2 + \sum_{j=1}^{k} \lambda_{1,j} |\beta_j|_1 \\ &\text{subject to} \quad \alpha^T\alpha = I_k. \end{aligned} \tag{11}$$

Whereas the same λ is used for all k components, different $\lambda_{1,j}$s are allowed for penalizing the loadings of different principal components. Again, if $p > n$, a positive λ is required in order to get exact PCA when the sparsity constraint (the lasso penalty) vanishes ($\lambda_{1,j} = 0$). (11) is called the SPCA criterion hereafter.

3.3 Numerical solution

We propose an alternatively minimization algorithm to minimize the SPCA criterion. From the proof of theorem 3 (see appendix for details) we get

$$\begin{aligned} & \textstyle\sum_{i=1}^{n} \left|\mathbf{X}_i - \alpha\beta^T\mathbf{X}_i\right|^2 + \lambda \textstyle\sum_{j=1}^{k} |\beta_j|^2 + \textstyle\sum_{j=1}^{k} \lambda_{1,j} |\beta_j|_1 \\ = \quad & \mathrm{Tr}\mathbf{X}^T\mathbf{X} + \textstyle\sum_{j=1}^{k} \left(\beta_j^T(\mathbf{X}^T\mathbf{X} + \lambda)\beta_j - 2\alpha_j^T\mathbf{X}^T\mathbf{X}\beta_j + \lambda_{1,j}|\beta_j|_1\right). \end{aligned} \tag{12}$$

Hence if given α, it amounts to solve k independent elastic net problems to get $\hat{\beta}_j$ for $j = 1, 2, \ldots, k$.

On the other hand, we also have (details in appendix)

$$\begin{aligned} & \textstyle\sum_{i=1}^{n} \left|\mathbf{X}_i - \alpha\beta^T\mathbf{X}_i\right|^2 + \lambda \textstyle\sum_{j=1}^{k} |\beta_j|^2 + \textstyle\sum_{j=1}^{k} \lambda_{1,j} |\beta_j|_1 \\ = \quad & \mathrm{Tr}\mathbf{X}^T\mathbf{X} - 2\mathrm{Tr}\alpha^T\mathbf{X}^T\mathbf{X}\beta + \mathrm{Tr}\beta^T(\mathbf{X}^T\mathbf{X} + \lambda)\beta + \textstyle\sum_{j=1}^{k} \lambda_{1,j} |\beta_j|_1 . \end{aligned} \tag{13}$$

Thus if β is fixed, we should maximize $\mathrm{Tr}\alpha^T(\mathbf{X}^T\mathbf{X})\beta$ subject to $\alpha^T\alpha = I_k$, whose solution is given by the following theorem.

Theorem 4 *Let α and β be $m \times k$ matrices and β has rank k. Consider the constrained maximization problem*

$$\hat{\alpha} = \arg\max_{\alpha} Tr\left(\alpha^T\beta\right) \quad \textit{subject to} \quad \alpha^T\alpha = I_k. \tag{14}$$

Suppose the SVD of β is $\beta = UDV^T$, then $\hat{\alpha} = UV^T$.

Here are the steps of our numerical algorithm to derive the first k sparse PCs.

General SPCA Algorithm

1. Let α start at $\mathbf{V}[, 1:k]$, the loadings of first k ordinary principal components.

2. Given fixed α, solve the following naive elastic net problem for $j = 1, 2, \ldots, k$

$$\beta_j = \arg\min_{\beta^*} \beta^{*T}(\mathbf{X}^T\mathbf{X} + \lambda)\beta^* - 2\alpha_j^T\mathbf{X}^T\mathbf{X}\beta^* + \lambda_{1,j}\left|\beta^*\right|_1 . \tag{15}$$

3. For each fixed β, do the SVD of $\mathbf{X}^T\mathbf{X}\beta = UDV^T$, then update $\alpha = UV^T$.

4. Repeat steps 2-3, until β converges.

5. Normalization: $\hat{V}_j = \frac{\beta_j}{|\beta_j|}$, $j = 1, \ldots, k$.

Some remarks:

1. Empirical evidence indicates that the outputs of the above algorithm vary slowly as λ changes. For $n > p$ data, the default choice of λ can be zero. Practically λ is a small positive number to overcome potential collinearity problems of $\mathbf{X}$. Section 4 discusses the default choice of λ for the data with thousands of variables, such as gene expression arrays.

2. In principle, we can try several combinations of $\{\lambda_{1,j}\}$ to figure out a good choice of the tunning parameters, since the above algorithm converges quite fast. There is a shortcut provided by the direct sparse approximation (8). The LARS-EN algorithm efficiently deliver a whole sequence of sparse approximations for each PC and the corresponding values of $\lambda_{1,j}$. Hence we can pick a $\lambda_{1,j}$ which gives a good compromise between variance and sparsity. In this selection, variance has a higher priority than sparsity, thus we tend to be conservative in pursuing sparsity.

3. Both PCA and SPCA depend on $\mathbf{X}$ only through $\mathbf{X}^T\mathbf{X}$. Note that $\frac{\mathbf{X}^T\mathbf{X}}{n}$ is actually the sample covariance matrix of variables $(\mathbf{X}_i)$. Therefore if Σ, the covariance matrix of $(\mathbf{X}_i)$, is known, we can replace $\mathbf{X}^T\mathbf{X}$ with Σ and have a population version of PCA or SPCA. If $\mathbf{X}$ is

standardized beforehand, then PCA or SPCA uses the (sample) correlation matrix, which is preferred when the scales of the variables are different.

3.4 Adjusted total variance

The ordinary principal components are uncorrelated and their loadings are orthogonal. Let $\hat{\Sigma} = \mathbf{X}^T\mathbf{X}$, then $\mathbf{V}^T\mathbf{V} = I_k$ and $\mathbf{V}^T\hat{\Sigma}\mathbf{V}$ is diagonal. It is easy to check that only the loadings of ordinary principal components can satisfy both conditions. In Jolliffe & Uddin (2003) the loadings were forced to be orthogonal, so the uncorrelated property was sacrificed. SPCA does not explicitly impose the uncorrelated components condition too.

Let $\hat{\mathbf{U}}$ be the modified PCs. Usually the total variance explained by $\hat{\mathbf{U}}$ is calculated by $trace(\hat{\mathbf{U}}^T\hat{\mathbf{U}})$. This is unquestionable when $\hat{\mathbf{U}}$ are uncorrelated. However, if they are correlated, the computed total variance is too optimistic. Here we propose a new formula to compute the total variance explained by $\hat{\mathbf{U}}$, which takes into account the correlations among $\hat{\mathbf{U}}$.

Suppose $(\hat{\mathbf{U}}_i, i = 1, 2, \ldots, k)$ are the first k modified PCs by any method. Denote $\hat{\mathbf{U}}_{j\cdot 1,\ldots,j-1}$ the reminder of $\hat{\mathbf{U}}_j$ after adjusting the effects of $\hat{\mathbf{U}}_1, \ldots, \hat{\mathbf{U}}_{j-1}$, that is

$$\hat{\mathbf{U}}_{j\cdot 1,\ldots,j-1} = \hat{\mathbf{U}}_j - H_{1,\ldots,j-1}\hat{\mathbf{U}}_j, \tag{16}$$

where $H_{1,\ldots,j-1}$ is the projection matrix on $\hat{\mathbf{U}}_i$ $i = 1, 2, \ldots, j-1$. Then the adjusted variance of $\hat{\mathbf{U}}_j$ is $\left|\hat{\mathbf{U}}_{j\cdot 1,\ldots,j-1}\right|^2$, and the total explained variance is given by $\sum_{j=1}^{k}\left|\hat{\mathbf{U}}_{j\cdot 1,\ldots,j-1}\right|^2$. When the modified PCs $\hat{\mathbf{U}}$ are uncorrelated, then the new formula agrees with $trace(\hat{\mathbf{U}}^T\hat{\mathbf{U}})$. Note that the above computations depend on the order of $\hat{\mathbf{U}}_i$. However, since we have a natural order in PCA, ordering is not an issue here.

Using the QR decomposition, we can easily compute the adjusted variance. Suppose $\hat{\mathbf{U}} = QR$,

where $\mathbf{Q}$ is orthonormal and $\mathbf{R}$ is upper triangular. Then it is straightforward to see that

$$\left|\hat{\mathbf{U}}_{j\cdot 1,\ldots,j-1}\right|^2 = \mathbf{R}_{j,j}^2. \tag{17}$$

Hence the explained total variance is equal to $\sum_{j=1}^{k} \mathbf{R}_{j,j}^2$.

3.5 Computation complexity

PCA is computationally efficient for both $n > p$ or $p \gg n$ data. We separately discuss the computational cost of the general SPCA algorithm for $n > p$ and $p \gg n$.

1. $n > p$. Traditional multivariate data fit in this category. Note that although the SPCA criterion is defined using $\mathbf{X}$, it only depends on $\mathbf{X}$ via $\mathbf{X}^T\mathbf{X}$. A trick is to first compute the $p \times p$ matrix $\hat{\Sigma} = \mathbf{X}^T\mathbf{X}$ once for all, which requires np^2 operations. Then the same $\hat{\Sigma}$ is used at each step within the loop. Computing $\mathbf{X}^T\mathbf{X}\beta$ costs p^2k and the SVD of $\mathbf{X}^T\mathbf{X}\beta$ is of order $O(pk^2)$. Each elastic net solution requires at most $O(p^3)$ operations. Since $k \leq p$, the total computation cost is at most $np^2 + mO(p^3)$, where m is the number of iterations before convergence. Therefore the SPCA algorithm is able to efficiently handle data with huge n, as long as p is small (say $p < 100$).

2. $p \gg n$. Gene expression arrays are typical examples of this $p \gg n$ category. The trick of $\hat{\Sigma}$ is no longer applicable, because $\hat{\Sigma}$ is a huge matrix $(p \times p)$ in this case. The most consuming step is solving each elastic net, whose cost is of order $O(pJ^2)$ for a positive finite λ, where J is the number of nonzero coefficients. Generally speaking the total cost is of order $mO(pJ^2k)$, which is expensive for a large J. Fortunately, as shown in Section 4, there exits a special SPCA algorithm for efficiently dealing with $p \gg n$ data.

4 SPCA for $p \gg n$ and Gene Expression Arrays

Gene expression arrays are a new type of data where the number of variables (genes) are much bigger than the number of samples. Our general SPCA algorithm still fits this situation using a positive λ. However the computation cost is expensive when requiring a large number of nonzero loadings. It is desirable to simplify the general SPCA algorithm to boost the computation.

Observe that theorem 3 is valid for all $\lambda > 0$, so in principle we can use any positive λ. It turns out that a thrifty solution emerges if $\lambda \to \infty$. Precisely, we have the following theorem.

Theorem 5 *Let* $\hat{\mathbf{V}}_i(\lambda) = \frac{\hat{\beta}_i}{|\hat{\beta}_i|}$ *be the loadings derived from criterion (11). Define* $(\hat{\alpha}^*, \hat{\beta}^*)$ *as the solution of the optimization problem*

$$\begin{aligned} (\hat{\alpha}^*, \hat{\beta}^*) &= \arg\min_{\alpha,\beta} -2\,Tr\alpha^T\mathbf{X}^T\mathbf{X}\beta + \sum_{j=1}^{k}\beta_j^2 + \sum_{j=1}^{k}\lambda_{1,j}\,|\beta_j|_1 \\ &\quad \textit{subject to} \quad \alpha^T\alpha = I_k. \end{aligned} \tag{18}$$

When $\lambda \to \infty$, $\hat{\mathbf{V}}_i(\lambda) \to \frac{\hat{\beta}_i^*}{|\hat{\beta}_i^*|}$.

By the same statements in Section 3.3, criterion (18) is solved by the following algorithm, which is a special case of the general SPCA algorithm with $\lambda = \infty$.

Gene Expression Arrays SPCA Algorithm

Replacing step 2 in the general SPCA algorithm with

Step 2*: Given fixed α, for $j = 1, 2, \ldots, k$

$$\beta_j = \left(\left|\alpha_j^T\mathbf{X}^T\mathbf{X}\right| - \frac{\lambda_{1,j}}{2}\right)_+ Sign(\alpha_j^T\mathbf{X}^T\mathbf{X}). \tag{19}$$

The operation in (19) is called soft-thresholding. Figure 1 gives an illustration of how the soft-thresholding rule operates. Recently soft-thresholding has become increasingly popular in

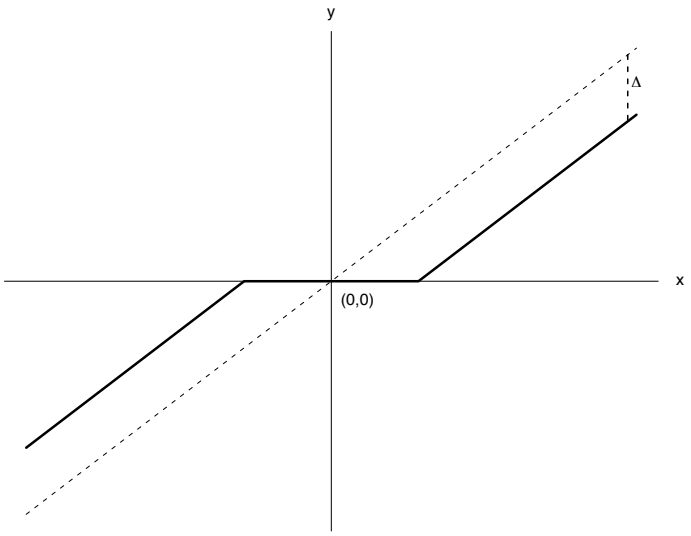

Figure 1: *An illstration of soft-thresholding rule* $y = (|x| - \Delta)_+ Sign(x)$ *with* $\Delta = 1$.

the literature. For example, nearest shrunken centroids (Tibshirani et al. 2002) adopts the soft-thresholding rule to simultaneously classify samples and select important genes in microarrays.

5 Examples

5.1 Pitprops data

The pitprops data first introduced in Jeffers (1967) has 180 observations and 13 measured variables. It is the classic example showing the difficulty of interpreting principal components. Jeffers (1967) tried to interpret the first 6 PCs. Jolliffe & Uddin (2003) used their SCoTLASS to find the modified PCs. Table 1 presents the results of PCA, while Table 2 presents the modified PCs loadings by SCoTLASS and the adjusted variance computed using (17).

As a demonstration, we also considered the first 6 principal components. Since this is a usual

$n \gg p$ data set, we set $\lambda = 0$. $\lambda_1 = (0.06, 0.16, 0.1, 0.5, 0.5, 0.5)$ were chosen according to Figure 2 such that each sparse approximation explained almost the same amount of variance as the ordinary PC did. Table 3 shows the obtained sparse loadings and the corresponding adjusted variance. Compared with the modified PCs by SCoTLASS, PCs by SPCA account for nearly the same amount of variance (75.8% vs. 78.2%) but with a much sparser loading structure. The important variables associated with the 6 PCs do not overlap, which further makes the interpretations easier and clearer. It is interesting to note that in Table 3 even though the variance does not strictly monotonously decrease, the adjusted variance follows the right order. However, Table 2 shows this is not true in SCoTLASS. It is also worthy to mention that the whole computation of SPCA was done in seconds in `R`, while the implementation of SCoTLASS for each t was expensive (Jolliffe & Uddin 2003). Optimizing SCoTLASS over several values of t is even a more difficult computational challenge.

Although the informal thresholding method, which is referred to as simple thresholding henceforth, has various drawbacks, it may serve as the benchmark for testing sparse PCs methods. An variant of simple thresholding is soft-thresholding. We found that used in PCA, soft-thresholding performs very similarly to simple thresholding. Thus we omitted the results of soft-thresholding in this paper. Both SCoTLASS and SPCA were compared with simple thresholding. Table 4 presents the loadings and the corresponding explained variance by simple thresholding. To make fair comparisons, we let the numbers of nonzero loadings by simple thresholding match the results of SCoTLASS and SPCA. In terms of variance, it seems that simple thresholding is better than SCoTLASS and worse than SPCA. Moreover, the variables with non-zero loadings by SPCA are very different to that chosen by simple thresholding for the first three PCs; while SCoTLASS seems to create a similar sparseness pattern as simple thresholding does, especially in the leading PC.

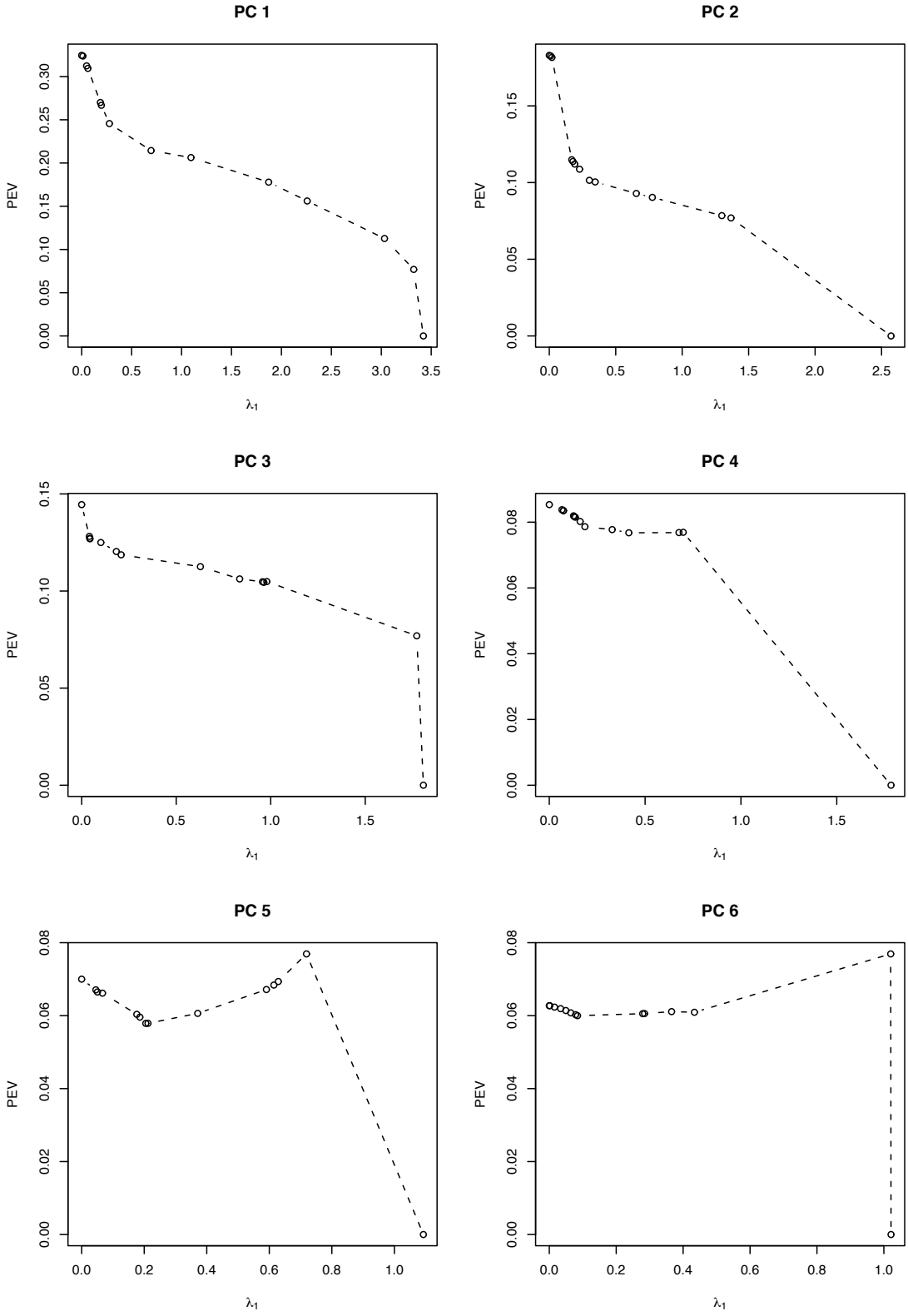

Figure 2: *Pitprops data: The sequences of sparse approximations to the first 6 principal components. Plots show the percentage of explained variance (PEV) as a function of* λ_1.

Table 1: *Pitprops data: loadings of the first 6 principal components*

Variable	PC1	PC2	PC3	PC4	PC5	PC6
topdiam	-0.404	0.218	-0.207	0.091	-0.083	0.120
length	-0.406	0.186	-0.235	0.103	-0.113	0.163
moist	-0.124	0.541	0.141	-0.078	0.350	-0.276
testsg	-0.173	0.456	0.352	-0.055	0.356	-0.054
ovensg	-0.057	-0.170	0.481	-0.049	0.176	0.626
ringtop	-0.284	-0.014	0.475	0.063	-0.316	0.052
ringbut	-0.400	-0.190	0.253	0.065	-0.215	0.003
bowmax	-0.294	-0.189	-0.243	-0.286	0.185	-0.055
bowdist	-0.357	0.017	-0.208	-0.097	-0.106	0.034
whorls	-0.379	-0.248	-0.119	0.205	0.156	-0.173
clear	0.011	0.205	-0.070	-0.804	-0.343	0.175
knots	0.115	0.343	0.092	0.301	-0.600	-0.170
diaknot	0.113	0.309	-0.326	0.303	0.080	0.626
Variance (%)	32.4	18.3	14.4	8.5	7.0	6.3
Cumulative Variance (%)	32.4	50.7	65.1	73.6	80.6	86.9

Table 2: *Pitprops data: loadings of the first 6 modified PCs by SCoTLASS*

$t = 1.75$						
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

Table 3: *Pitprops data: loadings of the first 6 sparse PCs by SPCA*

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

5.2 A simulation example

We first created three *hidden* factors

$$V_1 \sim N(0, 290), \qquad V_2 \sim N(0, 300)$$

$$V_3 = -0.3V_1 + 0.925V_2 + \epsilon, \qquad \epsilon \sim N(0, 1)$$

$$V_1, V_2 \text{ and } \epsilon \quad \text{are independent.}$$

Then 10 observed variables were generated as the follows

$$X_i = V_1 + \epsilon_i^1, \quad \epsilon_i^1 \sim N(0, 1), \quad i = 1, 2, 3, 4,$$

$$X_i = V_2 + \epsilon_i^2, \quad \epsilon_i^2 \sim N(0, 1), \quad i = 5, 6, 7, 8,$$

$$X_i = V_3 + \epsilon_i^3, \quad \epsilon_i^3 \sim N(0, 1), \quad i = 9, 10,$$

$$\{\epsilon_i^j\} \text{ are independent}, \qquad j = 1, 2, 3 \qquad i = 1, \cdots, 10.$$

Table 4: *Pitprops data: loadings of the first 6 modified PCs by simple thresholding*

Variable	PC1	PC2	PC3	PC4	PC5	PC6
topdiam	-0.439	0.234	0.000	0.092	0.000	0.120
length	-0.441	0.000	-0.253	0.104	0.000	0.164
moist	0.000	0.582	0.000	0.000	0.361	-0.277
testsg	0.000	0.490	0.379	0.000	0.367	0.000
ovensg	0.000	0.000	0.517	0.000	0.182	0.629
ringtop	0.000	0.000	0.511	0.000	-0.326	0.000
ringbut	-0.435	0.000	0.272	0.000	-0.222	0.000
bowmax	-0.319	0.000	-0.261	-0.288	0.191	0.000
bowdist	-0.388	0.000	0.000	-0.098	0.000	0.000
whorls	-0.412	-0.267	0.000	0.207	0.000	-0.174
clear	0.000	0.221	0.000	-0.812	-0.354	0.176
knots	0.000	0.369	0.000	0.304	-0.620	-0.171
diaknot	0.000	0.332	-0.350	0.306	0.000	0.629
Number of nonzero loadings	6	7	7	8	8	8
Variance (%)	28.9	16.6	14.2	8.6	6.9	6.3
Adjusted Variance (%)	28.9	16.5	14.0	8.5	6.7	6.2
Cumulative Adjusted Variance (%)	28.9	45.4	59.4	67.9	74.6	80.8
Variance	PC1	PC2	PC3	PC4	PC5	PC6
topdiam	-0.420	0.000	0.000	0	0	0
length	-0.422	0.000	0.000	0	0	0
moist	0.000	0.640	0.000	0	0	0
testsg	0.000	0.540	0.425	0	0	0
ovensg	0.000	0.000	0.580	0	0	0
ringtop	-0.296	0.000	0.573	0	0	0
ringbut	-0.416	0.000	0.000	0	0	0
bowmax	-0.305	0.000	0.000	0	0	0
bowdist	-0.370	0.000	0.000	0	0	0
whorls	-0.394	0.000	0.000	0	0	0
clear	0.000	0.000	0.000	-1	0	0
knots	0.000	0.406	0.000	0	-1	0
diaknot	0.000	0.365	-0.393	0	0	1
Number of nonzero loadings	7	4	4	1	1	1
Variance (%)	30.7	14.8	13.6	7.7	7.7	7.7
Adjusted Variance (%)	30.7	14.7	11.1	7.6	5.2	3.6
Cumulative Adjusted Variance (%)	30.7	45.4	56.5	64.1	68.3	71.9

To avoid the simulation randomness, we used the exact covariance matrix of $(X_1, \ldots, X_{10})$ to perform PCA, SPCA and simple thresholding. In other words, we compared their performances using an infinity amount of data generated from the above model.

The variance of the three underlying factors is 290, 300 and 283.8, respectively. The numbers of variables associated with the three factors are 4, 4 and 2. Therefore V_2 and V_1 are almost equally important, and they are much more important than V_3. The first two PCs together explain 99.6% of the total variance. These facts suggest that we only need to consider two derived variables with *'right'* sparse representations. Ideally, the first derived variable should recover the factor V_2 only using (X_5, X_6, X_7, X_8), and the second derived variable should recover the factor V_1 only using (X_1, X_2, X_3, X_4). In fact, if we sequentially maximize the variance of the first two derived variables under the orthonormal constraint, while restricting the numbers of nonzero loadings to four, then the first derived variable uniformly assigns nonzero loadings on (X_5, X_6, X_7, X_8); and the second derived variable uniformly assigns nonzero loadings on (X_1, X_2, X_3, X_4).

Both SPCA ($\lambda = 0$) and simple thresholding were carried out by using the oracle information that the ideal sparse representations use only four variables. Table 5 summarizes the comparison results. Clearly, SPCA correctly identifies the sets of important variables. As a matter of fact, SPCA delivers the ideal sparse representations of the first two principal components. Mathematically, it is easy to show that if $t = 2$ is used, SCoTLASS is also able to find the same sparse solution. In this example, both SPCA and SCoTLASS produce the ideal sparse PCs, which may be explained by the fact that both methods explicitly use the lasso penalty.

In contrast, simple thresholding wrongly includes X_9, X_{10} in the most important variables. The explained variance by simple thresholding is also lower than that by SPCA, although the relative difference is small (less than 5%). Due to the high correlation between V_2 and V_3, variables X_9, X_{10} gain loadings which are even higher than that of the true important varaibles (X_5, X_6, X_7, X_8). Thus

Table 5: *Results of the simulation example: loadings and variance*

	PCA			SPCA ($\lambda = 0$)		Simple	Thresholding
	PC1	PC2	PC3	PC1	PC2	PC1	PC2
X_1	0.116	-0.478	-0.087	0.0	0.5	0.000	-0.5
X_2	0.116	-0.478	-0.087	0.0	0.5	0.000	-0.5
X_3	0.116	-0.478	-0.087	0.0	0.5	0.000	-0.5
X_4	0.116	-0.478	-0.087	0.0	0.5	0.000	-0.5
X_5	-0.395	-0.145	0.270	0.5	0.0	0.000	0.0
X_6	-0.395	-0.145	0.270	0.5	0.0	0.000	0.0
X_7	-0.395	-0.145	0.270	0.5	0.0	-0.497	0.0
X_8	-0.395	-0.145	0.270	0.5	0.0	-0.497	0.0
X_9	-0.401	0.010	-0.582	0.0	0.0	-0.503	0.0
X_{10}	-0.401	0.010	-0.582	0.0	0.0	-0.503	0.0
Adjusted Variance (%)	60.0	39.6	0.08	40.9	39.5	38.8	38.6

the truth is disguised by the high correlation. On the other hand, simple thresholding correctly discovers the second factor, because V_1 has a low correlation with V_3.

5.3 Ramaswamy data

Ramaswamy data (Ramaswamy et al. 2001) has 16063 ($p = 16063$) genes and 144 ($n = 144$) samples. Its first principal component explains 46% of the total variance. In a typical microarray data like this, it appears that SCoTLASS cannot be practically useful. We applied SPCA ($\lambda = \infty$) to find the sparse leading PC. A sequence of λ_1 were used such that the number of nonzero loadings varied in a rather wide range. As displayed in Figure 3, the percentage of explained variance decreases at a slow rate, as the sparsity increase. As few as 2.5% of these 16063 genes can sufficiently construct the leading principal component with little loss of explained variance (from 46% to 40%). Simple thresholding was also applied to this data. It seems that when using the same number of genes, simple thresholding always explains slightly higher variance than SPCA does. Among the same number of selected genes by SPCA and simple thresholding, there are about 2% different genes, and this difference rate is quite consistent.

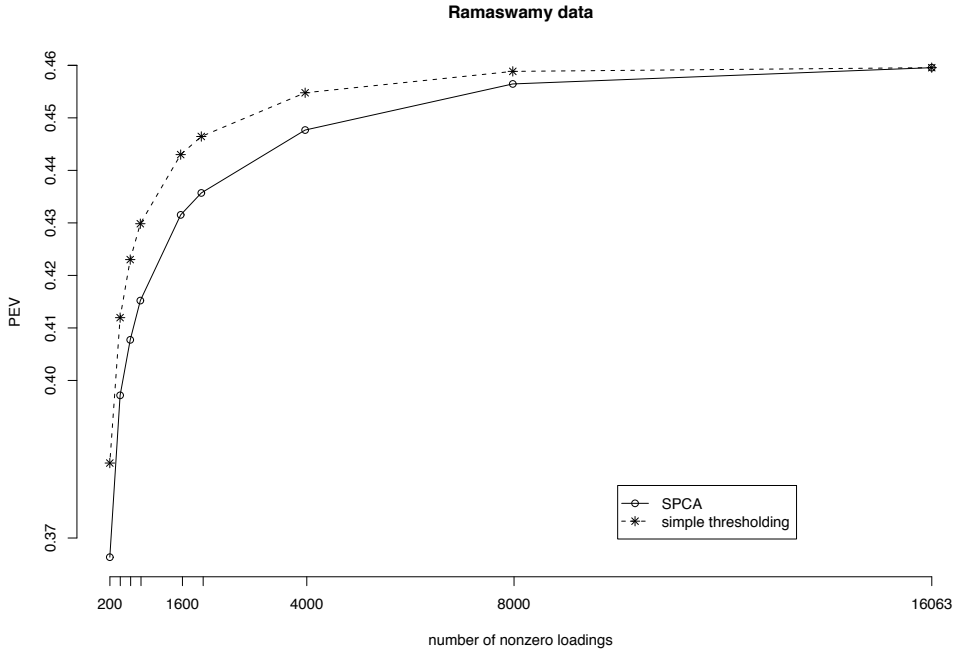

Figure 3: *The sparse leading principal component: percentage of explained variance versus sparsity. Simple thresholding and SPCA have similar performances. However, there still exists consistent difference in the selected genes (the ones with nonzero loadings).*

6 Discussion

It has been a long standing interest to have a formal approach to derive principal components with sparse loadings. From a practical point of view, a good method to achieve the sparseness goal should (at least) possess the following properties.

- Without any sparsity constraint, the method should reduce to PCA.
- It should be computationally efficient for both small p and big p data.
- It should avoid mis-identifying the important variables.

The frequently used simple thresholding is not criterion based. However, this informal ad hoc method seems to have the first two of the good properties listed above. If the explained variance and sparsity are the only concerns, simple thresholding is not such a bad choice, and it is extremely convenient. We have shown that simple thresholding can work pretty well in gene expression

arrays. The serious problem with simple thresholding is that it can mis-identify the real important variables. Nevertheless, simple thresholding is regarded as a benchmark for any potentially better method.

Using the lasso constraint in PCA, SCoTLASS successfully derives sparse loadings. However, SCoTLASS is not computationally efficient, and it lacks a good rule to pick its tunning parameter. In addition, it is not feasible to apply SCoTLASS to gene expression arrays, while in which PCA is a quite popular tool.

In this work we have developed SPCA using the SPCA criterion. The new SPCA criterion gives exact PCA results when its sparsity (lasso) penalty term vanishes. SPCA allows a quite flexible control on the sparse structure of the resulting loadings. Unified efficient algorithms have been proposed to realize SPCA for both regular multivariate data and gene expression arrays. As a principled procedure, SPCA enjoys advantages in several aspects, including computational efficiency, high explained variance and ability of identifying important variables.

7 Appendix: proofs

Theorem 1 proof: Using $\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^T$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, we have

$$\begin{aligned} \hat{\beta}_{ridge} &= \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T(\mathbf{X}\mathbf{V}_i) \\ &= \mathbf{V}\left(\frac{\mathbf{D}^2}{\mathbf{D}^2 + \lambda\mathbf{I}}\right)\mathbf{V}^T\mathbf{V}_i \\ &= \mathbf{V}_i\frac{\mathbf{D}_i^2}{\mathbf{D}_i^2 + \lambda}. \end{aligned} \tag{20}$$

□

Theorem 2 proof: Note that

$$\begin{aligned}\sum_{i=1}^{n}\left|\mathbf{X}_i-\alpha\beta^T\mathbf{X}_i\right|^2 &= \sum_{i=1}^{n}\mathrm{Tr}\mathbf{X}_i^T(I-\beta\alpha^T)(I-\alpha\beta^T)\mathbf{X}_i\\ &= \sum_{i=1}^{n}\mathrm{Tr}(I-\beta\alpha^T)(I-\alpha\beta^T)\mathbf{X}_i\mathbf{X}_i^T\\ &= \mathrm{Tr}(I-\beta\alpha^T)(I-\alpha\beta^T)(\sum_{i=1}^{n}\mathbf{X}_i\mathbf{X}_i^T)\\ &= \mathrm{Tr}(I-\beta\alpha^T-\alpha\beta^T+\beta\alpha^T\alpha\beta^T)\mathbf{X}^T\mathbf{X}\\ &= \mathrm{Tr}\mathbf{X}^T\mathbf{X}+\mathrm{Tr}\beta^T\mathbf{X}^T\mathbf{X}\beta-2\mathrm{Tr}\alpha^T\mathbf{X}^T\mathbf{X}\beta. \qquad (21)\end{aligned}$$

Since $\alpha^T\mathbf{X}^T\mathbf{X}\beta$ and $\beta^T\mathbf{X}^T\mathbf{X}\beta$ are both scalars, we get

$$\begin{aligned}&\sum_{i=1}^{n}\left|\mathbf{X}_i-\alpha\beta^T\mathbf{X}_i\right|^2+\lambda\left|\beta\right|^2\\ =\ &\mathrm{Tr}\mathbf{X}^T\mathbf{X}-2\alpha^T\mathbf{X}^T\mathbf{X}\beta+\beta^T(\mathbf{X}^T\mathbf{X}+\lambda)\beta. \qquad (22)\end{aligned}$$

For a fixed α, the above quantity is minimized at

$$\beta=\left(\mathbf{X}^T\mathbf{X}+\lambda\right)^{-1}\mathbf{X}^T\mathbf{X}\alpha. \qquad (23)$$

Substituting (23) into (22) gives

$$\begin{aligned}&\sum_{i=1}^{n}\left|\mathbf{X}_i-\alpha\beta^T\mathbf{X}_i\right|^2+\lambda\left|\beta\right|^2\\ =\ &\mathrm{Tr}\mathbf{X}^T\mathbf{X}-2\alpha^T\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda)^{-1}\mathbf{X}^T\mathbf{X}\alpha. \qquad (24)\end{aligned}$$

Therefore

$$\hat{\alpha} = \arg\max_{\alpha}\alpha^T\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda)^{-1}\mathbf{X}^T\mathbf{X}\alpha \qquad (25)$$

$$\text{subject to} \quad \alpha^T\alpha = 1.$$

And $\hat{\beta} = \left(\mathbf{X}^T\mathbf{X} + \lambda\right)^{-1}\mathbf{X}^T\mathbf{X}\hat{\alpha}$.

By $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, we have

$$\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda)^{-1}\mathbf{X}^T\mathbf{X} = \mathbf{V}\frac{\mathbf{D}^4}{\mathbf{D}^2 + \lambda}\mathbf{V}^T. \tag{26}$$

Hence $\hat{\alpha} = s\mathbf{V}_1$ with s=1 or -1. Then $\hat{\beta} = s\frac{\mathbf{D}_1^2}{\mathbf{D}_1^2+\lambda}\mathbf{V}_1$.

□

Theorem 3 proof: By the same steps in the proof of theorem 2 we derive (22) as long as $\alpha^T\alpha = I_k$. Hence we have

$$\begin{aligned} & \textstyle\sum_{i=1}^n \left|\mathbf{X}_i - \alpha\beta^T\mathbf{X}_i\right|^2 + \lambda\sum_{j=1}^k |\beta_j|^2 \\ = \quad & \mathrm{Tr}\mathbf{X}^T\mathbf{X} - 2\mathrm{Tr}\alpha^T\mathbf{X}^T\mathbf{X}\beta + \mathrm{Tr}\beta^T(\mathbf{X}^T\mathbf{X} + \lambda)\beta \qquad (27) \\ = \quad & \mathrm{Tr}\mathbf{X}^T\mathbf{X} + \textstyle\sum_{j=1}^k \left(\beta_j^T(\mathbf{X}^T\mathbf{X} + \lambda)\beta_j - 2\alpha_j^T\mathbf{X}^T\mathbf{X}\beta_j\right) \qquad (28) \end{aligned}$$

Thus given a fixed α, the above quantity is minimized at $\beta_j = \left(\mathbf{X}^T\mathbf{X} + \lambda\right)^{-1}\mathbf{X}^T\mathbf{X}\alpha_j$ for $j = 1, 2, \ldots, k$; or equivalently

$$\beta = \left(\mathbf{X}^T\mathbf{X} + \lambda\right)^{-1}\mathbf{X}^T\mathbf{X}\alpha. \tag{29}$$

Therefore

$$\begin{aligned} \hat{\alpha} \quad = \quad & \arg\max_{\alpha} \mathrm{Tr}\alpha^T\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda)^{-1}\mathbf{X}^T\mathbf{X}\alpha \qquad (30) \\ & \text{subject to} \quad \alpha^T\alpha = I_k. \end{aligned}$$

This is an eigen-analysis problem whose solution is $\hat{\alpha}_j = s_j\mathbf{V}_j$ with s_j=1 or -1 for $j = 1, 2, \ldots, k$,

because the eigenvectors of $\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda)^{-1}\mathbf{X}^T\mathbf{X}$ are $\mathbf{V}$. Hence (29) gives $\hat{\beta}_j = s_j \frac{\mathbf{D}_j^2}{\mathbf{D}_j^2+\lambda}\mathbf{V}_j$ for $j = 1, 2, \ldots, k$.

□

Theorem 4 proof: By assumption $\beta = UDV^T$ with $U^TU = I_k$ and $VV^T = V^TV = I_k$. The constraint $\alpha^T\alpha = I_k$ is equivalent to $\frac{k(k+1)}{2}$ constraints

$$\alpha_i^T\alpha_i = 1, \quad i = 1, 2 \ldots, k \tag{31}$$

$$\alpha_i^T\alpha_j = 0, \quad j > i. \tag{32}$$

Using Lagrangian multipliers method, we define

$$L = -\sum_{i=1}^{k}\beta_i^T\alpha_i + \sum_{i=1}^{k}\frac{1}{2}\lambda_{i,i}(\alpha_i^T\alpha_i - 1) + \sum_{j>i}^{k}\lambda_{i,j}(\alpha_i^T\alpha_j). \tag{33}$$

Setting $\frac{\partial L}{\partial \alpha_i} = 0$ gives $\beta_i = \lambda_{i,i}\hat{\alpha}_i + \lambda_{i,j}\hat{\alpha}_j$; or in a matrix form $\beta = \hat{\alpha}\Lambda$, where $\Lambda_{i,j} = \lambda_{j,i}$. Both β and α are full rank, so Λ is invertible and $\alpha = \beta\Lambda^{-1}$. We have

$$\mathrm{Tr}\hat{\alpha}^T\beta = \mathrm{Tr}\Lambda^{-1}\beta^T\beta = \mathrm{Tr}\left(\Lambda^{-1,T}VD^2V^T\right), \tag{34}$$

$$I_k = \hat{\alpha}^T\hat{\alpha} = \Lambda^{-1,T}\beta^T\beta\Lambda^{-1} = \Lambda^{-1,T}VD^2V^T\Lambda^{-1}. \tag{35}$$

Let $A = V^T\Lambda^{-1}V$, observe

$$\mathrm{Tr}\left(\Lambda^{-1}VD^2V^T\right) = \mathrm{Tr}\left(V^T\Lambda^{-1}VD^2\right) = \mathrm{Tr}A^TD^2 = \sum_{j=1}^{k}A_{jj}D_{jj}^2, \tag{36}$$

$$A^TD^2A = I_k. \tag{37}$$

Since $A_{jj}^2 D_{jj}^2 \leq 1$,

$$\sum_{j=1}^{k} A_{jj} D_{jj}^2 \leq \sum_{j=1}^{k} D_{jj}. \tag{38}$$

The "=" is taken if only if A is diagonal and $A_{jj} = D_{jj}^{-1}$. Therefore $\Lambda^{-1} = VAV^T = VD^{-1}V^T$, and $\hat{\alpha} = \beta\Lambda = UDV^T VD^{-1}V^T = UV^T$.

□

Theorem 5 proof: Let $\hat{\beta}^* = (1+\lambda)\hat{\beta}$, then we observe $\hat{\mathbf{V}}_i(\lambda) = \frac{\hat{\beta}_i^*}{|\hat{\beta}_i^*|}$. On the other hand, $\hat{\beta} = \frac{\hat{\beta}^*}{1+\lambda}$ means

$$\begin{aligned} (\hat{\alpha}, \hat{\beta}^*) &= \arg\min_{\alpha,\beta} \sum_{i=1}^{n} \left| \mathbf{X}_i - \alpha \frac{\beta^T}{1+\lambda} \mathbf{X}_i \right|^2 + \lambda \sum_{j=1}^{k} \left| \frac{\beta_j}{1+\lambda} \right|^2 + \sum_{j=1}^{k} \lambda_{1,j} \left| \frac{\beta_j}{1+\lambda} \right|_1 \\ &\text{subject to} \quad \alpha^T \alpha = I_k. \end{aligned} \tag{39}$$

Then by (12), we have

$$\begin{aligned} & \textstyle\sum_{i=1}^{n} \left| \mathbf{X}_i - \alpha \frac{\beta^T}{1+\lambda} \mathbf{X}_i \right|^2 + \lambda \sum_{j=1}^{k} \left| \frac{\beta_j}{1+\lambda} \right|^2 + \sum_{j=1}^{k} \lambda_{1,j} \left| \frac{\beta_j}{1+\lambda} \right|_1 \\ = \quad & \mathrm{Tr}\mathbf{X}^T\mathbf{X} + \tfrac{1}{1+\lambda} \left(\sum_{j=1}^{k} \left(\beta_j^T \tfrac{\mathbf{X}^T\mathbf{X}+\lambda}{1+\lambda} \beta_j - 2\alpha_j^T \mathbf{X}^T \mathbf{X} \beta_j + \lambda_{1,j} |\beta_j|_1 \right) \right) \\ = \quad & \mathrm{Tr}\mathbf{X}^T\mathbf{X} + \tfrac{1}{1+\lambda} \left(\sum_{j=1}^{k} \left(\beta_j^T \tfrac{\mathbf{X}^T\mathbf{X}+\lambda}{1+\lambda} \beta_j + \lambda_{1,j} |\beta_j|_1 \right) - 2\mathrm{Tr}\alpha^T \mathbf{X}^T \mathbf{X} \beta \right). \end{aligned} \tag{40}$$

$$\begin{aligned} (\hat{\alpha}, \hat{\beta}^*) &= \arg\min_{\alpha,\beta} -2\mathrm{Tr}\alpha^T \mathbf{X}^T \mathbf{X} \beta + \sum_{j=1}^{k} \beta_j^T \frac{\mathbf{X}^T\mathbf{X}+\lambda}{1+\lambda} \beta_j + \sum_{j=1}^{k} \lambda_{1,j} |\beta_j|_1 \\ &\text{subject to} \quad \alpha^T \alpha = I_k. \end{aligned} \tag{41}$$

As $\lambda \to \infty$, (41) approaches (18). Thus the conclusion follows.

□

References

Alter, O., Brown, P. & Botstein, D. (2000), 'Singular value decomposition for genome-wide expression data processing and modeling', *Proceedings of the National Academy of Sciences* **97**, 10101–10106.

Cadima, J. & Jolliffe, I. (1995), 'Loadings and correlations in the interpretation of principal components', *Journal of Applied Statistics* **22**, 203–214.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), 'Least angle regression', *The Annals of Statistics* **32**, In press.

Hancock, P., Burton, A. & Bruce, V. (1996), 'Face processing: human perception and principal components analysis', *Memory and Cognition* **24**, 26–40.

Hastie, T., Tibshirani, R., Eisen, M., Brown, P., Ross, D., Scherf, U., Weinstein, J., Alizadeh, A., Staudt, L. & Botstein, D. (2000), ''gene shaving' as a method for identifying distinct sets of genes with similar expression patterns', *Genome Biology* **1**, 1–21.

Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The Elements of Statistical Learning; Data mining, Inference and Prediction*, Springer Verlag, New York.

Jeffers, J. (1967), 'Two case studies in the application of principal component', *Applied Statistics* **16**, 225–236.

Jolliffe, I. (1986), *Principal component analysis*, Springer Verlag, New York.

Jolliffe, I. (1995), 'Rotation of principal components: choice of normalization constraints', *Journal of Applied Statistics* **22**, 29–35.

Jolliffe, I. T. & Uddin, M. (2003), 'A modified principal component technique based on the lasso', *Journal of Computational and Graphical Statistics* **12**, 531–547.

McCabe, G. (1984), 'Principal variables', *Technometrics* **26**, 137–144.

Misra, J., Schmitt, W., Hwang, D., Hsiao, L., Gullans, S., Stephanopoulos, G. & Stephanopoulos, G. (2002), 'Interactive exploration of microarray gene expression patterns in a reduced dimensional space', *Genome Research* **12**, 1112–1120.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukheriee, S., Yeang, C., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J., Poggio, T., Gerald, W., Loda, M., Lander, E. & Golub, T. (2001), 'Multiclass cancer diagnosis using tumor gene expression signature', *Proceedings of the National Academy of Sciences* **98**, 15149–15154.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society, Series B* **58**, 267–288.

Tibshirani, R., Hastie, T., Narasimhan, B. & Chu, G. (2002), 'Diagnosis of multiple cancer types by shrunken centroids of gene', *Proceedings of the National Academy of Sciences* **99**, 6567–6572.

Vines, S. (2000), 'Simple principal components', *Applied Statistics* **49**, 441–451.

Zou, H. & Hastie, T. (2003), Regression shrinkage and selection via the elastic net, with applications to microarrays, Technical report, Department of Statistics, Stanford University. Available at http://www-stat.stanford.edu/~hastie/pub.htm.