

→ PCA is hard to interpret  
⇒ Sparsity ⇒ 희소성 ! 해석하기  
간단하게

# Sparse Principal Component Analysis

Hui ZOU, Trevor HASTIE, and Robert TIBSHIRANI

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 first show that PCA can be formulated as a regression-type optimization problem; sparse loadings are then obtained by imposing the lasso (elastic net) constraint on the regression coefficients. Efficient algorithms are proposed to fit our SPCA models 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 with encouraging results.

**Key Words:** Arrays; Gene expression; Lasso/elastic net; Multivariate analysis; Singular value decomposition; Thresholding.

→ Interpretation

## 1. INTRODUCTION

Principal component analysis (PCA) (Jolliffe 1986) is a popular data-processing and dimension-reduction technique, with numerous applications in engineering, biology, and social science. Some interesting examples include handwritten zip code classification (Hastie, Tibshirani, and Friedman 2001) and human face recognition (Hancock, Burton, and Bruce 1996). Recently PCA has been used in gene expression data analysis (Alter, Brown, and Botstein 2000). Hastie et al. (2000) proposed the so-called *gene shaving* techniques using PCA to cluster highly variable and coherent genes in microarray datasets.

PCA seeks the linear combinations of the original variables such that the derived variables capture maximal variance. PCA can be computed 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

---

Hui Zou is Assistant Professor, School of Statistics, University of Minnesota, Minneapolis, MN 55455 (E-mail: [hzou@stat.umn.edu](mailto:hzou@stat.umn.edu)). Trevor Hastie is Professor, Department of Statistics, Stanford University, Stanford, CA 94305 (E-mail: [hastie@stat.stanford.edu](mailto:hastie@stat.stanford.edu)). Robert Tibshirani is Professor, Department of Health Research Policy, Stanford University, Stanford, CA 94305 (E-mail: [tibs@stat.stanford.edu](mailto:tibs@stat.stanford.edu)).

©2006 American Statistical Association, Institute of Mathematical Statistics,  
and Interface Foundation of North America

*Journal of Computational and Graphical Statistics*, Volume 15, Number 2, Pages 265–286  
DOI: 10.1198/106186006X113430

number of observations and the number of variables, respectively. Without loss of generality, assume the column means of  $\mathbf{X}$  are all 0. Let the SVD of  $\mathbf{X}$  be

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T. \quad (1.1)$$

$\mathbf{Z} = \mathbf{U}\mathbf{D}$  are the principal components (PCs), and the columns of  $\mathbf{V}$  are the corresponding loadings of the principal components. The sample variance of the  $i$ th PC is  $\mathbf{D}_{ii}^2/n$ . In gene expression data the standardized PCs  $\mathbf{U}$  are called the *eigen-arrays* and  $\mathbf{V}$  are the *eigen-genes* (Alter, Brown, and Botstein 2000). Usually the first  $q$  ( $q \ll \min(n, p)$ ) 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 the columns of  $\mathbf{X}$ , 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, that is, 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 number of explicitly used variables. An ad hoc way to achieve this 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 and Jolliffe 1995). McCabe (1984) presented an alternative to PCA which found a subset of *principal variables*. Jolliffe, Trendafilov, and Uddin (2003) introduced SCoTLASS to get modified principal components with possible zero loadings.

The same interpretation issues arise 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 and Hastie (2005) proposed the *elastic net*, a generalization of the lasso, which has some advantages. In this article we introduce a new approach for estimating 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, with a quadratic penalty; the lasso penalty (via the elastic net) can then be directly integrated into the regression criterion, leading to a modified PCA with sparse loadings.

In the next section we briefly review the lasso and the elastic net. The methodological details of SPCA are presented in Section 3. We present an efficient algorithm for fitting the SPCA model. We also derive an appropriate expression for representing the variance explained by modified principal components. In Section 4 we consider a special case of the SPCA algorithm for handling gene expression arrays efficiently. The proposed methodology

is illustrated by using real data and simulation examples in Section 5. Discussions are in Section 6. The article ends with an Appendix summarizing technical details.

## 2. THE LASSO AND THE ELASTIC NET

Consider the linear regression model with  $n$  observations and  $p$  predictors. Let  $Y = (y_1, \dots, y_n)^T$  be the response vector and  $\mathbf{X} = [X_1, \dots, X_p]$ ,  $j = 1, \dots, p$  the predictors, where  $X_j = (x_{1j}, \dots, x_{nj})^T$ . After a location transformation we can assume all the  $X_j$  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  $\hat{\beta}_{\text{lasso}}$  are obtained by minimizing the lasso criterion

$$\hat{\beta}_{\text{lasso}} = \arg \min_{\beta} \|Y - \sum_{j=1}^p X_j \beta_j\|^2 + \lambda \sum_{j=1}^p |\beta_j|, \quad (2.1)$$

where  $\lambda$  is non-negative. The lasso was originally solved by quadratic programming (Tibshirani 1996). Efron, Hastie, Johnstone, and Tibshirani (2004) showed that the lasso estimates  $\hat{\beta}$  are piecewise linear as a function of  $\lambda$ , and proposed an algorithm called LARS to efficiently solve the entire lasso solution path in the same order of computations as a single least squares fit. The piecewise linearity of the lasso solution path was first proved by Osborne, Presnell, Turlach (2000) where a different algorithm was proposed to solve the entire lasso solution path.

The lasso continuously shrinks the coefficients toward zero, and achieves its prediction accuracy via the bias variance trade-off. Due to the nature of the  $L_1$  penalty, some coefficients will be shrunk to exact zero if  $\lambda_1$  is large enough. Therefore the lasso simultaneously produces both an accurate and sparse model, which makes it a favorable variable selection method. However, the lasso has several limitations as pointed out by Zou and Hastie (2005). The most relevant one to this work is that the number of variables selected by the lasso is limited by the number of observations. For example, if applied to 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 and Hastie 2005) generalizes the lasso to overcome these drawbacks, while enjoying its other favorable properties. For any non-negative  $\lambda_1$  and  $\lambda_2$ , the elastic net estimates  $\hat{\beta}_{\text{en}}$  are given as follows

$$\hat{\beta}_{\text{en}} = (1 + \lambda_2) \left\{ \arg \min_{\beta} \|Y - \sum_{j=1}^p X_j \beta_j\|^2 + \lambda_2 \sum_{j=1}^p |\beta_j|^2 + \lambda_1 \sum_{j=1}^p |\beta_j| \right\}. \quad (2.2)$$

The elastic net penalty is a convex combination of the ridge and lasso penalties. Obviously, the lasso is a special case of the elastic net when  $\lambda_2 = 0$ . Given a fixed  $\lambda_2$ , the LARS-EN algorithm (Zou and Hastie 2005) efficiently solves the elastic net problem for all  $\lambda_1$  with the computational cost of 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 this particular limitation of the lasso is removed. Zou and Hastie (2005) compared the elastic net with the lasso and discussed the application of the elastic net as a gene selection method in microarray analysis.

### 3. MOTIVATION AND DETAILS OF SPCA

In both lasso and elastic net, the sparse coefficients are a direct consequence of the  $L_1$  penalty, and do not depend on the squared error loss function. Jolliffe, Trendafilov, and Uddin (2003) proposed SCoTLASS, an interesting procedure that obtains sparse loadings by directly imposing an  $L_1$  constraint on PCA. SCoTLASS successively maximizes the variance

$$a_k^T (\mathbf{X}^T \mathbf{X}) a_k, \tag{3.1}$$

subject to

$$a_k^T a_k = 1 \quad \text{and (for } k \geq 2) \quad a_h^T a_k = 0, \quad h < k; \tag{3.2}$$

and the extra constraints

$$\sum_{j=1}^p |a_{kj}| \leq t \tag{3.3}$$

for some tuning parameter  $t$ . Although sufficiently small  $t$  yields some exact zero loadings, there is not much guidance with SCoTLASS in choosing an appropriate value for  $t$ . One could try several  $t$  values, but the high computational cost of SCoTLASS makes this an impractical solution. This high computational cost is probably due to the fact that SCoTLASS is not a convex optimization problem. Moreover, the examples in Jolliffe, Trendafilov, and Uddin (2003) showed that the loadings obtained by SCoTLASS are not sparse enough when one requires a high percentage of explained variance.

We consider a different approach to modifying PCA. We first show how PCA can be recast exactly in terms of a (ridge) regression problem. We then introduce the lasso penalty by changing this ridge regression to an elastic-net regression.

#### 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.** For each  $i$ , denote by  $Z_i = \mathbf{U}_i \mathbf{D}_{ii}$  the  $i$ th principal component. Consider a positive  $\lambda$  and the ridge estimates  $\hat{\beta}_{\text{ridge}}$  given by

$$\hat{\beta}_{\text{ridge}} = \arg \min_{\beta} \|Z_i - \mathbf{X}\beta\|^2 + \lambda \|\beta\|^2. \tag{3.4}$$

$$\begin{aligned} X &= U D^{1/2} & U &= U D^{1/2} \\ X^T &= U^T D^{1/2} & &= X^T \end{aligned}$$

Let  $\hat{v} = \frac{\hat{\beta}_{\text{ridge}}}{\|\hat{\beta}_{\text{ridge}}\|}$ , then  $\hat{v} = V_i$ .

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

Now let us add the  $L_1$  penalty to (3.4) and consider the following optimization problem

$$\hat{\beta} = \arg \min_{\beta} \|Z_i - \mathbf{X}\beta\|^2 + \lambda\|\beta\|^2 + \lambda_1\|\beta\|_1, \tag{3.5}$$

where  $\|\beta\|_1 = \sum_{j=1}^p |\beta_j|$  is the 1-norm of  $\beta$ . We call  $\hat{V}_i = \frac{\hat{\beta}}{\|\hat{\beta}\|}$  an approximation to  $V_i$ , and  $\mathbf{X}\hat{V}_i$  the  $i$ th approximated principal component. Zou and Hastie (2005) called (3.5) *naive elastic net* 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  $\lambda_1$  gives a sparse  $\hat{\beta}$ , hence a sparse  $\hat{V}_i$ . Given a fixed  $\lambda$ , (3.5) is efficiently solved for all  $\lambda_1$  by using the LARS-EN algorithm (Zou and Hastie 2005). Thus, we can flexibly choose a sparse approximation to the  $i$ th principal component.

### 3.2 SPARSE PRINCIPAL COMPONENTS BASED ON THE 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 (3.5) to find suitable sparse approximations.

We now present a “self-contained” regression-type criterion to derive PCs. Let  $\mathbf{x}_i$  denote the  $i$ th row vector of the matrix  $\mathbf{X}$ . We first consider the leading principal component.

**Theorem 2.** For any  $\lambda > 0$ , let

$$\begin{aligned} (\hat{\alpha}, \hat{\beta}) &= \arg \min_{\alpha, \beta} \sum_{i=1}^n \|\mathbf{x}_i - \alpha\beta^T \mathbf{x}_i\|^2 + \lambda\|\beta\|^2 \\ &\text{subject to } \|\alpha\|^2 = 1. \end{aligned} \tag{3.6}$$

Then  $\hat{\beta} \propto V_1$ .

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  $\mathbf{A}_{p \times k} =$

$[\alpha_1, \dots, \alpha_k]$  and  $\mathbf{B}_{p \times k} = [\beta_1, \dots, \beta_k]$ . For any  $\lambda > 0$ , let

$$\begin{aligned} (\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) &= \arg \min_{\mathbf{A}, \mathbf{B}} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T \mathbf{x}_i\|^2 + \lambda \sum_{j=1}^k \|\beta_j\|^2 \\ &\text{subject to } \mathbf{A}^T \mathbf{A} = I_{k \times k}. \end{aligned} \quad (3.7)$$

Then  $\hat{\beta}_j \propto V_j$  for  $j = 1, 2, \dots, k$ .

Theorems 2 and 3 effectively transform the PCA problem to a regression-type problem. The critical element is the objective function  $\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T \mathbf{x}_i\|^2$ . If we restrict  $\mathbf{B} = \mathbf{A}$ , then

$$\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T \mathbf{x}_i\|^2 = \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A}\mathbf{A}^T \mathbf{x}_i\|^2,$$

whose minimizer under the orthonormal constraint on  $\mathbf{A}$  is exactly the first  $k$  loading vectors of ordinary PCA. This formulation arises in the ‘‘closest approximating linear manifold’’ derivation of PCA (e.g., Hastie, Tibshirani, and Friedman 2001). Theorem 3 shows that we can still have exact PCA while relaxing the restriction  $\mathbf{B} = \mathbf{A}$  and adding the ridge penalty term. As can be seen later, these generalizations enable us to flexibly modify PCA.

The proofs of Theorems 2 and 3 are given in the Appendix; here we give an intuitive explanation. Note that

$$\checkmark \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T \mathbf{x}_i\|^2 = \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2, \quad \underline{\mathbf{A}\mathbf{A}^T = \mathbf{I}} \quad (3.8)$$

Since  $\mathbf{A}$  is orthonormal, let  $\mathbf{A}_\perp$  be any orthonormal matrix such that  $[\mathbf{A}; \mathbf{A}_\perp]$  is  $p \times p$  orthonormal. Then we have

$$\checkmark \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2 = \checkmark \|\mathbf{X}\mathbf{A}_\perp\|^2 + \|\mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\|^2 \quad (3.9)$$

$$= \checkmark \|\mathbf{X}\mathbf{A}_\perp\|^2 + \sum_{j=1}^k \|\mathbf{X}\alpha_j - \mathbf{X}\beta_j\|^2. \quad (3.10)$$

Suppose  $\mathbf{A}$  is given, then the optimal  $\mathbf{B}$  minimizing (3.7) should minimize

$$\checkmark \arg \min_{\mathbf{B}} \sum_{j=1}^k \{\|\mathbf{X}\alpha_j - \mathbf{X}\beta_j\|^2 + \lambda \|\beta_j\|^2\} \quad (3.11)$$

which is equivalent to  $k$  independent ridge regression problems. In particular, if  $\mathbf{A}$  corresponds to the ordinary PCs, that is,  $\mathbf{A} = \mathbf{V}$ , then by Theorem 1, we know that  $\mathbf{B}$  should be proportional to  $\mathbf{V}$ . Actually, the above view points out an effective algorithm for solving (3.7), which is revisited in the next section.

We carry on the connection between PCA and regression, and use the lasso approach to produce sparse loadings (‘‘regression coefficients’’). For that purpose, we add the lasso penalty into the criterion (3.7) and consider the following optimization problem

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) = \arg \min_{\mathbf{A}, \mathbf{B}} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T \mathbf{x}_i\|^2 + \lambda \sum_{j=1}^k \|\beta_j\|^2 + \sum_{j=1}^k \lambda_{1,j} \|\beta_j\|_1$$

$$\text{subject to } \mathbf{A}^T \mathbf{A} = I_{k \times k}.$$

(3.12)

Whereas the same  $\lambda$  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  $\lambda$  is required in order to get exact PCA when the sparsity constraint (the lasso penalty) vanishes ( $\lambda_{1,j} = 0$ ). We call (3.12) the SPCA criterion hereafter.

### 3.3 NUMERICAL SOLUTION

We propose an alternating algorithm to minimize the SPCA criterion (3.12).

**A given A:** For each  $j$ , let  $Y_j^* = \mathbf{X}\alpha_j$ . By the same analysis used in (3.9)–(3.11), we know that  $\hat{\mathbf{B}} = [\hat{\beta}_1, \dots, \hat{\beta}_k]$ , where each  $\hat{\beta}_j$  is an elastic net estimate

$$\hat{\beta}_j = \arg \min_{\beta_j} \|Y_j^* - \mathbf{X}\beta_j\|^2 + \lambda\|\beta_j\|^2 + \lambda_{1,j}\|\beta_j\|_1. \tag{3.13}$$

**A given B:** On the other hand, if  $\mathbf{B}$  is fixed, then we can ignore the penalty part in (3.12) and only try to minimize  $\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T \mathbf{x}_i\|^2 = \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2$ , subject to  $\mathbf{A}^T \mathbf{A} = \mathbf{I}_{k \times k}$ . The solution is obtained by a reduced rank form of the *Procrustes rotation*, given in Theorem 4 below. We compute the SVD

$$(\mathbf{X}^T \mathbf{X})\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{3.14}$$

and set  $\hat{\mathbf{A}} = \mathbf{U}\mathbf{V}^T$ .

**Theorem 4. Reduced Rank Procrustes Rotation.** Let  $\mathbf{M}_{n \times p}$  and  $\mathbf{N}_{n \times k}$  be two matrices. Consider the constrained minimization problem

$$\hat{\mathbf{A}} = \arg \min_{\mathbf{A}} \|\mathbf{M} - \mathbf{N}\mathbf{A}^T\|^2 \quad \text{subject to} \quad \mathbf{A}^T \mathbf{A} = \mathbf{I}_{k \times k}. \tag{3.15}$$

Suppose the SVD of  $\mathbf{M}^T \mathbf{N}$  is  $\mathbf{U}\mathbf{D}\mathbf{V}^T$ , then  $\hat{\mathbf{A}} = \mathbf{U}\mathbf{V}^T$ .

The usual Procrustes rotation (e.g., Marcia, Kent, and Bibby, 1979) has  $\mathbf{N}$  the same size as  $\mathbf{M}$ .

It is worth pointing out that to solve (3.13), we only need to know the Gram matrix  $\mathbf{X}^T \mathbf{X}$ , because

$$\begin{aligned} & \|Y_j^* - \mathbf{X}\beta_j\|^2 + \lambda\|\beta_j\|^2 + \lambda_{1,j}\|\beta_j\|_1 \\ &= (\alpha_j - \beta_j)^T \mathbf{X}^T \mathbf{X} (\alpha_j - \beta_j) + \lambda\|\beta_j\|^2 + \lambda_{1,j}\|\beta_j\|_1. \end{aligned} \tag{3.16}$$

The same is true of (3.14).

Now  $\frac{1}{n} \mathbf{X}^T \mathbf{X}$  is the sample covariance matrix of  $\mathbf{X}$ . Therefore if  $\Sigma$ , the covariance matrix of  $\mathbf{X}$ , is known, we can replace  $\mathbf{X}^T \mathbf{X}$  with  $\Sigma$  in (3.16) and have a population version of SPCA. If  $\mathbf{X}$  is standardized beforehand, then we use the (sample) correlation matrix, which is preferred when the scales of the variables are different.

Although (3.16) (with  $\Sigma$  instead of  $\mathbf{X}^T \mathbf{X}$ ) is not quite an elastic net problem, we can easily turn it into one. Create the artificial response  $Y^{**}$  and  $\mathbf{X}^{**}$  as follows

$$Y^{**} = \Sigma^{\frac{1}{2}} \alpha_j \quad \mathbf{X}^{**} = \Sigma^{\frac{1}{2}}, \quad (3.17)$$

then it is easy to check that

$$\hat{\beta}_j = \arg \min_{\beta} \|Y^{**} - \mathbf{X}^{**} \beta\|^2 + \lambda \|\beta\|^2 + \lambda_{1,j} \|\beta\|_1. \quad (3.18)$$

Algorithm 1 summarizes the steps of our SPCA procedure outlined above.

### Algorithm 1. General SPCA Algorithm

1. Let  $\mathbf{A}$  start at  $\mathbf{V}[1 : k]$ , the loadings of the first  $k$  ordinary principal components.
2. Given a fixed  $\mathbf{A} = [\alpha_1, \dots, \alpha_k]$ , solve the following elastic net problem for  $j = 1, 2, \dots, k$

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

3. For a fixed  $\mathbf{B} = [\beta_1, \dots, \beta_k]$ , compute the SVD of  $\mathbf{X}^T \mathbf{X} \mathbf{B} = \mathbf{U} \mathbf{D} \mathbf{V}^T$ , then update  $\mathbf{A} = \mathbf{U} \mathbf{V}^T$ .
4. Repeat Steps 2–3, until convergence.
5. Normalization:  $\hat{V}_j = \frac{\beta_j}{\|\beta_j\|}$ ,  $j = 1, \dots, k$ .

Some remarks:

1. Empirical evidence suggests that the output of the above algorithm does not change much as  $\lambda$  is varied. For  $n > p$  data, the default choice of  $\lambda$  can be zero. Practically  $\lambda$  is chosen to be a small positive number to overcome potential collinearity problems in  $\mathbf{X}$ . Section 4 discusses the default choice of  $\lambda$  for 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 tuning parameters, since the above algorithm converges quite fast. There is a shortcut provided by the direct sparse approximation (3.5). The LARS-EN algorithm efficiently delivers a whole sequence of sparse approximations for each PC and the corresponding values of  $\lambda_{1,j}$ . Hence we can pick a  $\lambda_{1,j}$  that gives a good compromise between variance and sparsity. When facing the variance-sparsity trade-off, we let variance have a higher priority.

## 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  $\mathbf{V}^T \mathbf{V} = \mathbf{I}_k$  and  $\mathbf{V}^T \hat{\Sigma} \mathbf{V}$  is diagonal. It is easy to check that it is only for ordinary principal components the the loadings can satisfy both conditions. In Jolliffe, Trendafilov, and Uddin (2003) the loadings were forced to be orthogonal, so the uncorrelated property was sacrificed. SPCA does not explicitly impose the uncorrelated components condition either.

Let  $\widehat{\mathbf{Z}}$  be the modified PCs. Usually the total variance explained by  $\widehat{\mathbf{Z}}$  is calculated by  $\text{tr}(\widehat{\mathbf{Z}}^T \widehat{\mathbf{Z}})$ . This is reasonable when  $\widehat{\mathbf{Z}}$  are uncorrelated. However, if they are correlated,  $\text{tr}(\widehat{\mathbf{Z}}^T \widehat{\mathbf{Z}})$  is too optimistic for representing the total variance. Suppose  $(\hat{Z}_i, i = 1, 2, \dots, k)$  are the first  $k$  modified PCs by any method, and the  $(k+1)$ th modified PC  $\hat{Z}_{k+1}$  is obtained. We want to compute the total variance explained by the first  $k+1$  modified PCs, which should be the sum of the explained variance by the first  $k$  modified PCs and the additional variance from  $\hat{Z}_{k+1}$ . If  $\hat{Z}_{k+1}$  is correlated with  $(\hat{Z}_i, i = 1, 2, \dots, k)$ , then its variance contains contributions from  $(\hat{Z}_i, i = 1, 2, \dots, k)$ , which should not be included into the total variance given the presence of  $(\hat{Z}_i, i = 1, 2, \dots, k)$ .

Here we propose a new formula to compute the total variance explained by  $\widehat{\mathbf{Z}}$ , which takes into account the correlations among  $\widehat{\mathbf{Z}}$ . We use regression projection to remove the linear dependence between correlated components. Denote  $\hat{Z}_{j \cdot 1, \dots, j-1}$  the residual after adjusting  $\hat{Z}_j$  for  $\hat{Z}_1, \dots, \hat{Z}_{j-1}$ , that is

$$\hat{Z}_{j \cdot 1, \dots, j-1} = \hat{Z}_j - \mathbf{H}_{1, \dots, j-1} \hat{Z}_j, \quad (3.19)$$

where  $\mathbf{H}_{1, \dots, j-1}$  is the projection matrix on  $\{\hat{Z}_i\}_1^{j-1}$ . Then the adjusted variance of  $\hat{Z}_j$  is  $\|\hat{Z}_{j \cdot 1, \dots, j-1}\|^2$ , and the total explained variance is defined as  $\sum_{j=1}^k \|\hat{Z}_{j \cdot 1, \dots, j-1}\|^2$ . When the modified PCs  $\widehat{\mathbf{Z}}$  are uncorrelated, the new formula agrees with  $\text{tr}(\widehat{\mathbf{Z}}^T \widehat{\mathbf{Z}})$ .

Note that the above computations depend on the order of  $\hat{Z}_i$ . 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  $\widehat{\mathbf{Z}} = \mathbf{Q}\mathbf{R}$ , where  $\mathbf{Q}$  is orthonormal and  $\mathbf{R}$  is upper triangular. Then it is straightforward to see that

$$\|\hat{Z}_{j \cdot 1, \dots, j-1}\|^2 = \mathbf{R}_{jj}^2. \quad (3.20)$$

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

### 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  $\widehat{\mathbf{\Sigma}} = \mathbf{X}^T \mathbf{X}$  once for all, which requires  $np^2$  operations. Then the same  $\widehat{\mathbf{\Sigma}}$  is used at each step within the loop. Computing  $\mathbf{X}^T \mathbf{X} \beta$  costs  $p^2 k$  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 in this  $p \gg n$  category. The trick of using  $\widehat{\mathbf{\Sigma}}$  is no longer applicable, because  $\widehat{\mathbf{\Sigma}}$  is a huge matrix ( $p \times p$ ) in this case. The most consuming step is solving each elastic net, whose cost is of order

$O(pnJ + J^3)$  for a positive finite  $\lambda$ , where  $J$  is the number of nonzero coefficients. Generally speaking the total cost is of order  $mkO(pJn + J^3)$ , which can be expensive for large  $J$  and  $p$ . Fortunately, as shown in the next section, there exists a special SPCA algorithm for efficiently dealing with  $p \gg n$  data.

#### 4. SPCA FOR $p \gg n$ AND GENE EXPRESSION ARRAYS

For gene expression arrays the number of variables (genes) is typically much bigger than the number of samples (e.g.,  $n = 10,000$ ,  $p = 100$ ). Our general SPCA algorithm still fits this situation using a positive  $\lambda$ . However the computational 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  $\lambda$ . It turns out that a thrifty solution emerges if  $\lambda \rightarrow \infty$ . Precisely, we have the following theorem.

**Theorem 5.** Let  $\widehat{V}_j(\lambda) = \frac{\widehat{\beta}_j}{\|\widehat{\beta}_j\|}$  ( $j = 1, \dots, k$ ) be the loadings derived from criterion (3.12). Let  $(\widehat{\mathbf{A}}, \widehat{\mathbf{B}})$  be the solution of the optimization problem

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) = \arg \min_{\mathbf{A}, \mathbf{B}} -2\text{tr}(\mathbf{A}^T \mathbf{X}^T \mathbf{X} \mathbf{B}) + \sum_{j=1}^k \|\beta_j\|^2 + \sum_{j=1}^k \lambda_{1,j} \|\beta_j\|_1 \quad (4.1)$$

subject to  $\mathbf{A}^T \mathbf{A} = \mathbf{I}_{k \times k}$ .

When  $\lambda \rightarrow \infty$ ,  $\widehat{V}_j(\lambda) \rightarrow \frac{\beta_j}{\|\beta_j\|}$ .

We can use the same alternating algorithm in Section 3.3 to solve (4.1), where we only need to replace the general elastic net problem with its special case ( $\lambda = \infty$ ). Note that given  $\mathbf{A}$ ,

$$\widehat{\beta}_j = \arg \min_{\beta_j} -2\alpha_j^T (\mathbf{X}^T \mathbf{X}) \beta_j + \|\beta_j\|^2 + \lambda_{1,j} \|\beta_j\|_1, \quad (4.2)$$

which has an explicit form solution given in (4.3).

**Gene Expression Arrays SPCA Algorithm.** Replacing Step 2 in the general SPCA algorithm with

**Step 2\*:** for  $j = 1, 2, \dots, k$

$$\beta_j = \left( \left| \alpha_j^T \mathbf{X}^T \mathbf{X} \right| - \frac{\lambda_{1,j}}{2} \right)_+ \text{Sign}(\alpha_j^T \mathbf{X}^T \mathbf{X}). \quad (4.3)$$

The operation in (4.3) is called soft-thresholding. Figure 1 gives an illustration of how the soft-thresholding rule operates. Recently soft-thresholding has become increasingly popular in the literature. For example, nearest shrunken centroids (Tibshirani, Hastie, Narasimhan, and Chu 2002) adopts the soft-thresholding rule to simultaneously classify samples and select important genes in microarrays.

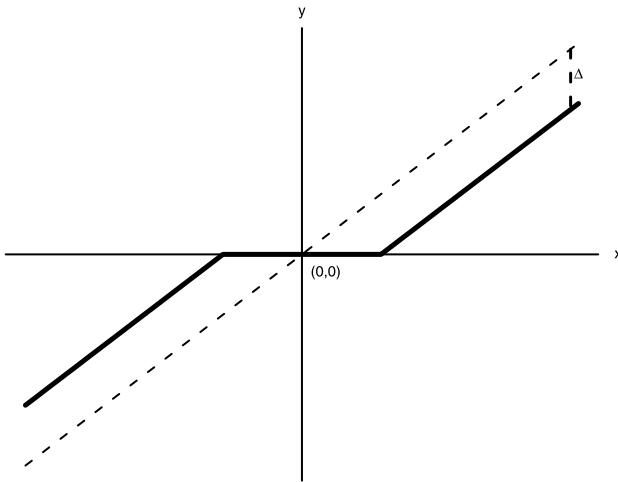


Figure 1. An illustration of soft-thresholding rule  $y = (|x| - \Delta)_+ \text{Sign}(x)$  with  $\Delta = 1$ .

## 5. EXAMPLES

### 5.1 PITPROPS DATA

The pitprops data first introduced by Jeffers (1967) has 180 observations and 13 measured variables. It is a classic example showing the difficulty of interpreting principal components. Jeffers (1967) tried to interpret the first six PCs. Jolliffe, Trendafilov, and Uddin (2003) used their SCoTLASS to find the modified PCs. Table 1 presents the results of PCA, while Table 2 presents the modified PC loadings as computed by SCoTLASS and the adjusted variance computed using (3.20).

As a demonstration, we also considered the first six principal components. Since this is a usual  $n \gg p$  dataset, 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 of SCoTLASS, PCs by SPCA account for a larger amount of variance (75.8% vs. 69.3%) with a much sparser loading structure. The important variables associated with the six 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 that this is not true in SCoTLASS. It is also worthy of mention that the entire SPCA computation was done in seconds in R, while the implementation of SCoTLASS for each  $t$  was expensive (Jolliffe, Trendafilov, and Uddin 2003). Optimizing SCoTLASS over several values of  $t$  is an even more difficult computational challenge.

Although the informal thresholding method, which we henceforth refer to as simple

Table 1. Pitprops Data: Loadings of the First Six Principal Components

<i>Variable</i>	<i>PC1</i>	<i>PC2</i>	<i>PC3</i>	<i>PC4</i>	<i>PC5</i>	<i>PC6</i>
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 Six Modified PCs by SCoTLASS. Empty cells have zero loadings.

<i>t</i> = 1.75 <i>Variable</i>	<i>PC1</i>	<i>PC2</i>	<i>PC3</i>	<i>PC4</i>	<i>PC5</i>	<i>PC6</i>
topdiam	0.664			-0.025	0.002	-0.035
length	0.683	-0.001		-0.040	0.001	-0.018
moist		0.641	0.195		0.180	-0.030
testsg		0.701	0.001			-0.001
ovensg					-0.887	-0.056
ringtop		0.293	-0.186		-0.373	0.044
ringbut	0.001	0.107	-0.658		-0.051	0.064
bowmax	0.001			0.735	0.021	-0.168
bowdist	0.283					-0.001
whorls	0.113		-0.001	0.388	-0.017	0.320
clear						-0.923
knots		0.001		-0.554	0.016	0.004
diaknot			0.703	0.001	-0.197	0.080
Number of nonzero loadings	6	6	6	6	10	13
Variance (%)	19.6	16.0	13.1	13.1	9.2	9.0
Adjusted variance (%)	19.6	13.8	12.4	8.0	7.1	8.4
Cumulative adjusted variance (%)	19.6	33.4	45.8	53.8	60.9	69.3

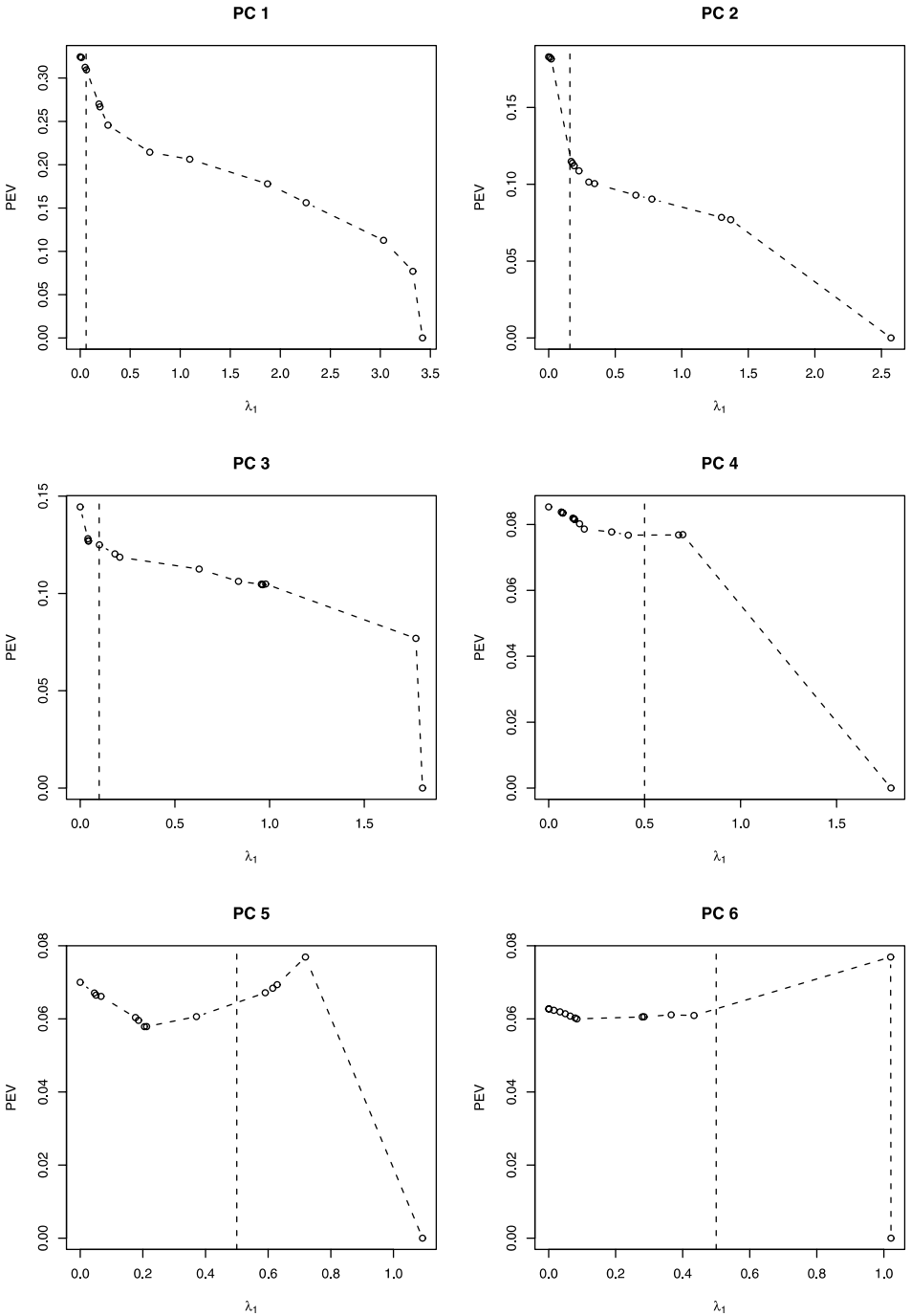


Figure 2. Pitprops data: The sequences of sparse approximations to the first six principal components. The curves show the percentage of explained variance (PEV) as a function of  $\lambda_1$ . The vertical broken lines indicate the choice of  $\lambda_1$  used in our SPCA analysis.

Table 3. Pitprops Data: Loadings of the First Six Sparse PCs by SPCA. Empty cells have zero loadings.

Variable	PC1	PC2	PC3	PC4	PC5	PC6
topdiam	-0.477					
length	-0.476					
moist		0.785				
testsg		0.620				
ovensg	0.177		0.640			
ringtop			0.589			
ringbut			0.492			
bowmax	-0.250	-0.021				
bowdist	-0.344					
bowdist	-0.416					
whorls	-0.400					
clear				-1		
knots		0.013			-1	
diaknot			-0.015			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

thresholding, has various drawbacks, it may serve as the benchmark for testing sparse PCs methods. A variant of simple thresholding is soft-thresholding. We found that when used in PCA, soft-thresholding performs very similarly to simple thresholding. Thus, we omitted the results of soft-thresholding in this article. Both SCoTLASS and SPCA were compared with simple thresholding. Table 4 presents the loadings and the corresponding variance explained by simple thresholding. To make the comparisons fair, we let the numbers of nonzero loadings obtained by simple thresholding match the results of SCoTLASS and SPCA, as shown in the top and bottom parts of Table 4, respectively. In terms of variance, it seems that simple thresholding is better than SCoTLASS and worse than SPCA. Moreover, the variables with nonzero loadings by SPCA are 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.

## 5.2 A SYNTHETIC EXAMPLE

Our synthetic example has three *hidden* factors

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

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

and

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

Then 10 observable variables are constructed as follows

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

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

We used the exact covariance matrix of  $(X_1, \dots, X_{10})$  to perform PCA, SPCA and simple thresholding (in the population setting).

Table 4. Pitprops Data: Loadings of the First Six Modified PCs by Simple Thresholding. Empty cells have zero loadings.

<i>Variable</i>	<i>PC1</i>	<i>PC2</i>	<i>PC3</i>	<i>PC4</i>	<i>PC5</i>	<i>PC6</i>
<i>Simple thresholding vs. SCoTLASS</i>						
topdiam	-0.439	0.240				0.120
length	-0.441			0.105	-0.114	0.163
moist		0.596			0.354	-0.276
testsg		0.503	0.391		0.360	-0.054
ovensg			0.534		0.178	0.626
ringtop			0.528		-0.320	0.052
ringbut	-0.435		0.281		-0.218	0.003
bowmax	-0.319		-0.270	-0.291	0.188	-0.055
bowdist	-0.388					0.034
whorls	-0.412	-0.274		0.209	0.158	-0.173
clear				-0.819	-0.347	0.175
knots		0.378		0.307	-0.608	-0.170
diaknot		0.340	-0.362	0.309		0.626
Number of nonzero loadings	6	6	6	6	10	13
Variance (%)	28.9	16.1	15.4	8.4	7.1	6.3
Adjusted Variance (%)	28.9	16.1	13.9	8.2	6.9	6.2
Cumulative adjusted variance (%)	28.9	45.0	58.9	67.1	74.0	80.2
<i>Simple thresholding vs. SPCA</i>						
topdiam	-0.420					
length	-0.422					
moist		0.640				
testsg		0.540	0.425			
ovensg			0.580			
ringtop	-0.296		0.573			
ringbut	-0.416					
bowmax	-0.305					
bowdist	-0.370					
whorls	-0.394					
clear					-1	
knots		0.406				
diaknot		0.365	-0.393		-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

Table 5. Results of the Simulation Example: Loadings and Variance

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

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 “correct” 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. In 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 incorrectly includes  $X_9, X_{10}$  in the most important variables. The variance explained 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  $X_9, X_{10}$  achieve loadings which are even higher than those of the true important variables  $(X_5, X_6, X_7, X_8)$ . Thus 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

An important task in microarray analysis is to find a set of genes which are biologically

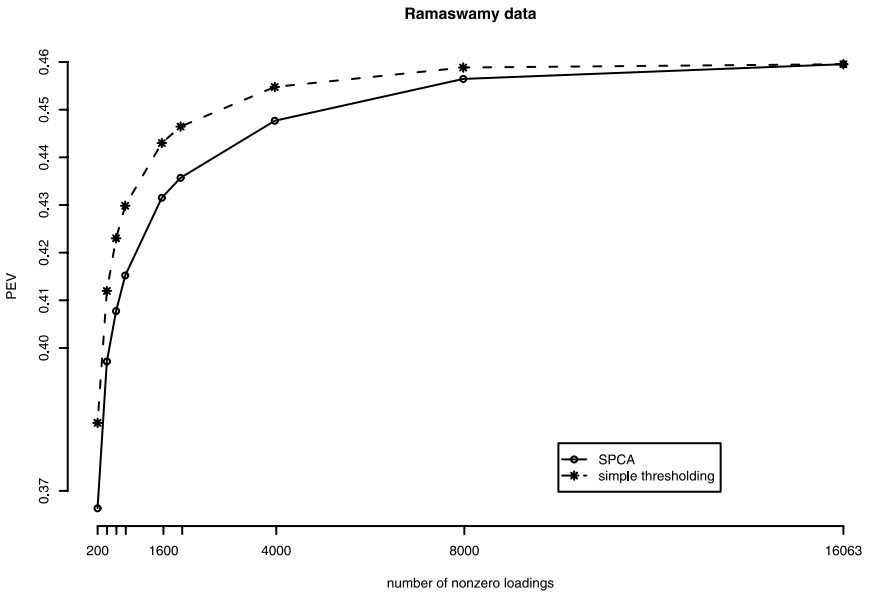


Figure 3. The sparse leading principal component: percentage of explained variance versus sparsity. Simple thresholding and SPCA have similar performances. However, there still exists consistent difference in the selected genes (the ones with nonzero loadings).

relevant to the outcome (e.g. tumor type or survival time). PCA (or SVD) has been a popular tool for this purpose. Many gene-clustering methods in the literature use PCA (or SVD) as a building block. For example, *gene shaving* (Hastie et al. 2000) uses an iterative principal component shaving algorithm to identify subsets of coherent genes. Here we consider another approach to gene selection through SPCA. The idea is intuitive: if the (sparse) principal component can explain a large part of the total variance of gene expression levels, then the subset of genes representing the principal component are considered important.

We illustrate the sparse PC selection method on Ramaswamy's data (Ramaswamy et al. 2001) which has 16,063 ( $p = 16,063$ ) genes and 144 ( $n = 144$ ) samples. Its first principal component explains 46% of the total variance. For microarray data like this, it appears that SCoTLASS cannot be practically useful for finding sparse PCs. We applied SPCA ( $\lambda = \infty$ ) to find the leading sparse PC. A sequence of values for  $\lambda_1$  were used such that the number of nonzero loadings varied over a wide range. As displayed in Figure 3, the percentage of explained variance decreases at a slow rate, as the sparsity increases. As few as 2.5% of these 16,063 genes can sufficiently construct the leading principal component with an affordable 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.

## 6. DISCUSSION

It has been an interesting research topic for years 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 misidentifying the important variables.

The often-used simple thresholding approach is not criterion based. However, this informal method seems to possess the first two of the desirable properties listed above. If the explained variance and sparsity are the only concerns, simple thresholding is a reasonable approach, and it is extremely convenient. We have shown that simple thresholding can work well with gene expression arrays. The serious problem with simple thresholding is that it can misidentify 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 tuning parameter. In addition, it is not feasible to apply SCOTLASS to gene expression arrays, where PCA is a quite popular tool.

In this work we have developed SPCA using our SPCA criterion (3.12). This new criterion gives exact PCA results when its sparsity (lasso) penalty term vanishes. SPCA allows flexible control on the sparse structure of the resulting loadings. Unified efficient algorithms have been proposed to compute SPCA solutions 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 an ability in identifying important variables.

Software in R for fitting the SPCA model (and elastic net models) is available in the CRAN contributed package `elasticnet`.

## APPENDIX: PROOFS

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

$$\hat{\beta}_{\text{ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T (\mathbf{X} \mathbf{V}_i) = V_i \frac{\mathbf{D}_{ii}^2}{\mathbf{D}_{ii}^2 + \lambda}. \quad (A.1)$$

$\rightarrow$  이번엔  $\lambda$ 가 0이면  $\hat{\beta}_{\text{ridge}} = V_i$ 로 수렴함  
 $\rightarrow$   $\lambda$ 가 클수록  $\hat{\beta}_{\text{ridge}}$ 가 0에 가까워짐  
 $\rightarrow$  Constant

Hence  $\hat{v} = V_i$ .  $\square$

Note that since Theorem 2 is a special case of Theorem 3, we will not prove it separately. We first provide a lemma.

**Lemma 1.** Consider the ridge regression criterion

$$C_\lambda(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|^2.$$

Then if  $\hat{\beta} = \arg \min_{\beta} C_{\lambda}(\beta)$ ,

$$C_{\lambda}(\hat{\beta}) = \mathbf{y}^T (\mathbf{I} - \mathbf{S}_{\lambda}) \mathbf{y},$$

where  $\mathbf{S}_{\lambda}$  is the ridge operator

$$\mathbf{S}_{\lambda} = \mathbf{X}(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T.$$

(beta) < (loading과 관계가 있다.) -> normal을 이루면 된다.

**Proof of Lemma 1:** Differentiating  $C_{\lambda}$  with respect to  $\beta$ , we get that

$$-\mathbf{X}^T (\mathbf{y} - \mathbf{X}\hat{\beta}) + \lambda \hat{\beta} = 0.$$

Premultiplication by  $\hat{\beta}^T$  and re-arrangement gives  $\lambda \|\hat{\beta}\|^2 = (\mathbf{y} - \mathbf{X}\hat{\beta})^T \mathbf{X}\hat{\beta}$ . Since

$$\|\mathbf{y} - \mathbf{X}\hat{\beta}\|^2 = (\mathbf{y} - \mathbf{X}\hat{\beta})^T \mathbf{y} - (\mathbf{y} - \mathbf{X}\hat{\beta})^T \mathbf{X}\hat{\beta},$$

$C_{\lambda}(\hat{\beta}) = (\mathbf{y} - \mathbf{X}\hat{\beta})^T \mathbf{y}$ . The result follows since the "fitted values"  $\mathbf{X}\hat{\beta} = \mathbf{S}_{\lambda} \mathbf{y}$ . □

**Proof of Theorem 3.** We use the notation introduced in Section 3:  $\mathbf{A} = [\alpha_1, \dots, \alpha_k]$  and  $\mathbf{B} = [\beta_1, \dots, \beta_k]$ . Let

$$C_{\lambda}(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T \mathbf{x}_i\|^2 + \lambda \sum_{j=1}^k \|\beta_j\|^2.$$

As in (3.9) we have

$$\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T \mathbf{x}_i\|^2 = \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2 \tag{A.2}$$

$$= \|\mathbf{X}\mathbf{A}_{\perp}\|^2 + \|\mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\|^2. \tag{A.3}$$

Hence, with  $\mathbf{A}$  fixed, solving

when A is fixed -

$$\arg \min_{\mathbf{B}} C_{\lambda}(\mathbf{A}, \mathbf{B})$$

is equivalent to solving the series of ridge regressions

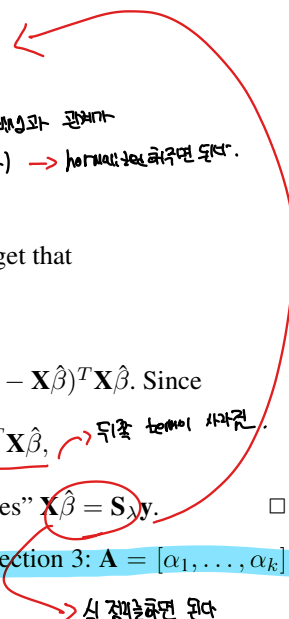
$$\arg \min_{\{\beta_j\}_1^k} \sum_{j=1}^k \{\|\mathbf{X}\alpha_j - \mathbf{X}\beta_j\|^2 + \lambda \|\beta_j\|^2\}.$$

It is easy to show that

$$\hat{\mathbf{B}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{X} \mathbf{A}, \tag{A.4}$$

and using Lemma 1 and (A.2) we have that the partially optimized penalized criterion is given by

$$C_{\lambda}(\mathbf{A}, \hat{\mathbf{B}}) = \|\mathbf{X}\mathbf{A}_{\perp}\|^2 + \text{tr}((\mathbf{X}\mathbf{A})^T (\mathbf{I} - \mathbf{S}_{\lambda}) (\mathbf{X}\mathbf{A})). \tag{A.5}$$



이것이

math-x-version

Rearranging the terms, we get

$$C_\lambda(\mathbf{A}, \hat{\mathbf{B}}) = \text{tr}(\mathbf{X}^T \mathbf{X}) - \text{tr}(\mathbf{A}^T \mathbf{X}^T \mathbf{S}_\lambda \mathbf{X} \mathbf{A}), \tag{A.6}$$

which must be minimized with respect to  $\mathbf{A}$  with  $\mathbf{A}^T \mathbf{A} = \mathbf{I}$ . Hence  $\mathbf{A}$  should be taken to be the largest  $k$  eigenvectors of  $\mathbf{X}^T \mathbf{S}_\lambda \mathbf{X}$ . If the SVD of  $\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^T$ , it is easy to show that  $\mathbf{X}^T \mathbf{S}_\lambda \mathbf{X} = \mathbf{V} \mathbf{D}^2 (\mathbf{D}^2 + \lambda \mathbf{I})^{-1} \mathbf{D}^2 \mathbf{V}^T$ , hence  $\hat{\mathbf{A}} = \mathbf{V}[1 : k]$ . Likewise, plugging the SVD of  $\mathbf{X}$  into (A.4), we see that each of the  $\hat{\beta}_j$  are scaled elements of the corresponding  $V_j$ .

**Proof of Theorem 4:** We expand the matrix norm

$$\|\mathbf{M} - \mathbf{N} \mathbf{A}^T\|^2 = \text{tr}(\mathbf{M}^T \mathbf{M}) - 2\text{tr}(\mathbf{M}^T \mathbf{N} \mathbf{A}^T) + \text{tr}(\mathbf{A} \mathbf{N}^T \mathbf{N} \mathbf{A}^T). \tag{A.7}$$

Since  $\mathbf{A}^T \mathbf{A} = \mathbf{I}$ , the last term is equal to  $\text{tr}(\mathbf{N}^T \mathbf{N})$ , and hence we need to maximize (minus half) the middle term. With the SVD  $\mathbf{M}^T \mathbf{N} = \mathbf{U} \mathbf{D} \mathbf{V}^T$ , this middle term becomes

$$\text{tr}(\mathbf{M}^T \mathbf{N} \mathbf{A}^T) = \text{tr}(\mathbf{U} \mathbf{D} \mathbf{V}^T \mathbf{A}^T) \tag{A.8}$$

$$= \text{tr}(\mathbf{U} \mathbf{D} \mathbf{A}^{*T}) \tag{A.9}$$

$$= \text{tr}(\mathbf{A}^{*T} \mathbf{U} \mathbf{D}), \tag{A.10}$$

where  $\mathbf{A}^* = \mathbf{A} \mathbf{V}$ , and since  $\mathbf{V}$  is  $k \times k$  orthonormal,  $\mathbf{A}^{*T} \mathbf{A}^* = \mathbf{I}$ . Now since  $\mathbf{D}$  is diagonal, (A.10) is maximized when the diagonal of  $\mathbf{A}^{*T} \mathbf{U}$  is positive and maximum. By Cauchy-Schwartz inequality, this is achieved when  $\mathbf{A}^* = \mathbf{U}$ , in which case the diagonal elements are all 1. Hence  $\hat{\mathbf{A}} = \mathbf{U} \mathbf{V}^T$ .  $\square$

**Proof of Theorem 5:** Let  $\hat{\mathbf{B}}^* = [\hat{\beta}_1^*, \dots, \hat{\beta}_k^*]$  with  $\hat{\beta}^* = (1 + \lambda)\hat{\beta}$ , then  $\hat{V}_i(\lambda) = \frac{\hat{\beta}_i^*}{\|\hat{\beta}_i^*\|}$ .

On the other hand,  $\hat{\beta} = \frac{\hat{\beta}^*}{1 + \lambda}$  means that

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}^*) = \arg \min_{\mathbf{A}, \mathbf{B}} C_{\lambda, \lambda_1}(\mathbf{A}, \mathbf{B}) \quad \text{subject to} \quad \mathbf{A}^T \mathbf{A} = \mathbf{I}_{k \times k}, \tag{A.11}$$

where

$$C_{\lambda, \lambda_1}(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A} \frac{\mathbf{B}^T}{1 + \lambda} \mathbf{x}_i\|^2 + \lambda \sum_{j=1}^k \left\| \frac{\beta_j}{1 + \lambda} \right\|^2 + \sum_{j=1}^k \lambda_{1,j} \left\| \frac{\beta_j}{1 + \lambda} \right\|_1. \tag{A.12}$$

Since

$$\sum_{j=1}^k \left\| \frac{\beta_j}{1 + \lambda} \right\|^2 = \frac{1}{(1 + \lambda)^2} \text{tr}(\mathbf{B}^T \mathbf{B}),$$

and

$$\begin{aligned} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{A} \frac{\mathbf{B}^T}{1 + \lambda} \mathbf{x}_i\|^2 &= \text{tr} \left( (\mathbf{X} - \mathbf{X} \frac{\mathbf{B}}{1 + \lambda} \mathbf{A}^T)^T (\mathbf{X} - \mathbf{X} \frac{\mathbf{B}}{1 + \lambda} \mathbf{A}^T) \right) \\ &= \text{tr}(\mathbf{X}^T \mathbf{X}) + \frac{1}{(1 + \lambda)^2} \text{tr}(\mathbf{B}^T \mathbf{X}^T \mathbf{X} \mathbf{B}) - \frac{2}{1 + \lambda} \text{tr}(\mathbf{A}^T \mathbf{X}^T \mathbf{X} \mathbf{B}). \end{aligned}$$

Thus we have

$$C_{\lambda, \lambda_1}(\mathbf{A}, \mathbf{B}) = \text{tr}(\mathbf{X}^T \mathbf{X}) + \frac{1}{1 + \lambda} \left( \text{tr} \left( \mathbf{B}^T \frac{\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}}{1 + \lambda} \mathbf{B} \right) - 2\text{Tr} \left( \mathbf{A}^T \mathbf{X}^T \mathbf{X} \mathbf{B} \right) + \sum_{j=1}^k \lambda_{1,j} \|\beta_j\|_1 \right),$$

which implies that

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}^*) = \arg \min_{\mathbf{A}, \mathbf{B}} \text{tr} \left( \mathbf{B}^T \frac{\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}}{1 + \lambda} \mathbf{B} \right) - 2\text{tr} \left( \mathbf{A}^T \mathbf{X}^T \mathbf{X} \mathbf{B} \right) + \sum_{j=1}^k \lambda_{1,j} \|\beta_j\|_1, \quad (\text{A.13})$$

subject to  $\mathbf{A}^T \mathbf{A} = \mathbf{I}_{k \times k}$ . As  $\lambda \rightarrow \infty$ ,  $\text{tr} \left( \mathbf{B}^T \frac{\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}}{1 + \lambda} \mathbf{B} \right) \rightarrow \text{tr}(\mathbf{B}^T \mathbf{B}) = \sum_{j=1}^k \|\beta_j\|^2$ . Thus (A.13) approaches (4.1) and the conclusion of Theorem 5 follows.  $\square$

## ACKNOWLEDGMENTS

We thank the editor, an associate editor, and referees for helpful comments and suggestions which greatly improved the manuscript.

[Received April 2004. Revised June 2005.]

## REFERENCES

- Alter, O., Brown, P., and Botstein, D. (2000), "Singular Value Decomposition for Genome-Wide Expression Data Processing and Modeling," in *Proceedings of the National Academy of Sciences*, 97, pp. 10101–10106.
- Cadima, J., and 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., and Tibshirani, R. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 407–499.
- Hancock, P., Burton, A., and Bruce, V. (1996), "Face Processing: Human Perception and Principal Components Analysis," *Memory and Cognition*, 24, 26–40.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning; Data mining, Inference and Prediction*, New York: Springer Verlag.
- Hastie, T., Tibshirani, R., Eisen, M., Brown, P., Ross, D., Scherf, U., Weinstein, J., Alizadeh, A., Staudt, L., and Botstein, D. (2000), "'gene Shaving' as a Method for Identifying Distinct Sets of Genes With Similar Expression Patterns," *Genome Biology*, 1, 1–21.
- Jeffers, J. (1967), "Two Case Studies in the Application of Principal Component," *Applied Statistics*, 16, 225–236.
- Jolliffe, I. (1986), *Principal Component Analysis*, New York: Springer Verlag.
- (1995), "Rotation of Principal Components: Choice of Normalization Constraints," *Journal of Applied Statistics*, 22, 29–35.
- Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. (2003), "A Modified Principal Component Technique Based on the Lasso," *Journal of Computational and Graphical Statistics*, 12, 531–547.
- Mardia, K., Kent, J., and Bibby, J. (1979), *Multivariate Analysis*, New York: Academic Press.
- McCabe, G. (1984), "Principal Variables," *Technometrics*, 26, 137–144.

- Osborne, M. R., Presnell, B., and Turlach, B. A. (2000), "A New Approach to Variable Selection in Least Squares Problems," *IMA Journal of Numerical Analysis*, 20, 389–403.
- Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J., Poggio, T., Gerald, W., Loda, M., Lander, E., and Golub, T. (2001), "Multiclass Cancer Diagnosis using Tumor Gene Expression Signature," in *Proceedings of the National Academy of Sciences*, 98, pp. 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., and Chu, G. (2002), "Diagnosis of Multiple Cancer Types by Shrunken Centroids of Gene," in *Proceedings of the National Academy of Sciences*, 99, 6567–6572.
- Vines, S. (2000), "Simple Principal Components," *Applied Statistics*, 49, 441–451.
- Zou, H., and Hastie, T. (2005), "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society, Series B*, 67, 301–320.