Principal Component Analysis with Sparse Fused Loadings

Frank Jian Guo, Gareth James, Elizaveta Levina, George Michailidis and Ji Zhu

September 16, 2009

Abstract

In this paper, we propose a new method for principal component analysis (PCA), whose main objective is to capture natural "blocking" structures in the variables. Further, the method beyond selecting different variables for different components, also encourages the loadings of highly correlated variables to have the same magnitude. These two features often help in interpreting the principal components. To achieve these goals, a fusion penalty is introduced and the resulting optimization problem solved by an alternating block optimization algorithm. The method is applied to a number of simulated and real datasets and is shown that it achieves the stated objectives.

Keywords: Principal component analysis; Sparsity; Variable selection; Fusion penalty; Local quadratic approximation

1 Introduction

Principal component analysis (PCA) is a widely used data analytic technique that aims to reduce the dimensionality of the data for simplifying further analysis and visualization. It achieves its goal by constructing a sequence of *orthogonal linear combinations* of the original variables, called the principal components (PC), that have maximum variance. The technique is often used in exploratory mode and hence good interpretability of the resulting principal components is an important goal. However, it is often hard to achieve this in practice, since PCA tends to produce principal components that involve *all* the variables. Further, the orthogonality requirement often determines the signs of the variable loadings (coefficients) beyond the first few components, which makes meaningful interpretation challenging.

Various alternatives to ordinary PCA have been proposed in the literature to aid interpretation, including rotations of the components (Jollife, 1995), restrictions for their loadings

to take values in the set $\{-1, 0, 1\}$ (Vines, 2000), and construction of components based on a subset of the original variables (McCabe, 1984). More recently, variants of PCA that attempt to select different variables for different components have been proposed and are based on a regularization framework that penalizes some norm of the PC vectors. Such variants include

SCoTLASS (Jollife et al., 2003) that imposes an ℓ_1 penalty on the ordinary PCA loadings and a recent sparse PCA technique (Zou et al., 2006) that extends the elastic net (Zou and Hastie, 2005) procedure by relaxing the PCs orthogonality requirement.

In this paper, we propose another version of PCA with sparse components motivated by the following empirical considerations. In many application areas, some variables are highly correlated and form natural "blocks". For example, in the meat spectra example discussed in Section 4, the spectra exhibit high correlations within the high and low frequency regions, thus giving rise to such a block structure. Something analogous occurs in the image data, where the background forms one natural block, and the foreground one or more such blocks. In such cases, the loadings of the block tend to be of similar magnitude. The proposed technique is geared towards exploring such block structures and producing sparse principal components whose loadings are of the same sign and magnitude, thus significantly aiding interpretation of the results. We call this property *fusion* and introduce a penalty that forces "fusing" of loadings of highly correlated variables in addition to forcing small loadings to zero. We refer to this method as sparse fused PCA (SFPCA).

The remainder of the paper is organized as follows: the technical development and computing algorithm for our method are presented in Section 2. An illustration of the method based on simulated data is given in Section 3. In Section 4, we apply the new method to several real datasets. Finally, some concluding remarks are drawn in Section 5.

2 The Model and its Estimation

2.1 Preliminaries and Sparse Variants of PCA

Let $X = (x_{i,j})_{n\times p}$ be a data matrix comprised of n observations and p variables, whose columns are assumed to be centered. As noted above, PCA reduces the dimensionality of the data by constructing linear combinations of the original variables that have maximum variance; i. e.,

for $k = 1, \cdots, p$, define

$$\alpha_k = \arg\max_{\alpha} Var(X\alpha), \quad \text{subject to } \alpha_k'\alpha_k = 1, \alpha_k'\alpha_j = 0 \ \text{ for all } j \neq k, \tag{1}$$

where α_k is a p-dimensional vector called *factor loadings* (PC vectors). The projection of the data $Z_k = X\alpha_k$ is called the k-th principal component. The technique proves most successful if one can use a small number $k \ll p$ of components to account for most of the variance and thus provide a relatively simple explanation of the underlying data structure. Some algebra shows that the factor loadings can be obtained by solving the following optimization problem

$$\widehat{\alpha}_k = \arg \max_{\alpha \perp \alpha_1, \ldots, \alpha_{k-1}} \alpha^T \widehat{\Sigma} \alpha \tag{2}$$

where $\widehat{\Sigma} = 1/n(X^T X)$ denotes the sample covariance of the data. The solution of (2) is given by the eigenvector corresponding to the k-th largest eigenvalue of $\widehat{\Sigma}$. An alternative way to derive the PC vectors, which proves useful in subsequent developments, is to solve the following constrained least squares problem:

$$\min_{A} \|X - XAA^T\|_F^2, \quad \text{subject to } \ A^T A = I_K \ , \tag{3}$$

where I_K denotes a $K \times K$ identity matrix, $\|M\|_F$ is the Frobenius norm of a matrix M ($\|M\|_F^2 = \sum_{i,j} M_{ij}^2$), and $A = [\alpha_1, \ldots, \alpha_K]$ is a $p \times K$ matrix with orthogonal columns. The estimate $\widehat{A}$ contains the first K PC vectors, and $\widehat{Z} = X\widehat{A}$ the first K principal components.

To impose sparsity on the PC vectors, Jollife et al. (2003) proposed SCoTLASS, which adds an ℓ_1-norm constraint to objective function (2), i.e., for any $1 \leq k \leq K$, solve:

$$\max_{\alpha \perp \alpha_1, \ldots, \alpha_{k-1}} \alpha^T \widehat{\Sigma} \alpha \ , \ \text{subject to } \ \|\alpha\|_1 \leq t \ , \tag{4}$$

where $\|\alpha\|_1 = \sum_{j=1}^{p} |\alpha_j|$ is the ℓ_1 norm of the vector α. Due to the singularity property of the ℓ_1 norm, the constraint $\|\alpha\|_1 \leq t$ shrinks some components of α to zero for small enough values of t. Therefore, objective function (2) produces sparse PC vectors. However, Zou et al. (2006) noted that in many cases, SCoTLASS fails to achieve sufficient sparsity, thus complicating the interpretation of the results. One possible explanation stems from the orthogonality constraint of the PC vectors that is not fully compatible with the desired sparsity condition. Hence, Zou et al. (2006) proposed an alternative way to estimate sparse PC vectors, by relaxing the orthogonality requirement. Their procedure amounts to solving the following

regularized regression problem:

$$\begin{aligned} \arg\min_{A,B} \quad & \|X - XBA^T\|_F^2 + \lambda_1 \sum_{k=1}^{K} \|\beta_k\|_1 + \lambda_2 \sum_{k=1}^{K} \|\beta_k\|_2^2 \\ \text{subject to} \quad & A^T A = I_K \ , \end{aligned} \tag{5}$$

where β_k is a p-dimensional column vector and $B = [\beta_1, \beta_2, \ldots, \beta_K]$. The l_2 penalty $\sum_{k=1}^{K} \|\beta_k\|_2^2$ regularizes the loss function to avoid singular solutions, whenever $n < p$. If $\lambda_1 = 0$, objective function (5) reduces to the ordinary PCA problem and the columns of $\widehat{B}$ are proportional to the first K ordinary PC vectors (Zou et al., 2006); otherwise, the ℓ_1 penalty $\|\beta_k\|_1$ imposes sparsity on the elements of $\widehat{B}$, i.e., it shrinks some loadings exactly to zero. In addition, the first term in (5) can be written as

$$\begin{aligned} \|X - XBA^T\|_F^2 &= \|XA - XB\|_F^2 + \|A_\perp\|_F^2 \\ &= \sum_{k=1}^{K} \|X\alpha_k - X\beta_k\|_F^2 + \|A_\perp\|_F^2 \\ &= n \sum_{k=1}^{K} (\alpha_k - \beta_k)^T \widehat{\Sigma} (\alpha_k - \beta_k) + \|A_\perp\|_F^2 \end{aligned} \tag{6}$$

where $A_\perp$ is any orthonormal matrix such that $[A, A_\perp]$ is a $p \times p$ orthonormal matrix. The quantity $(\alpha_k - \beta_k)^T \widehat{\Sigma} (\alpha_k - \beta_k)$, $1 \leq k \leq p$ measures the difference between α_k and β_k. Therefore, although there is no direct constraint on the column orthogonality in B, the loss function shrinks the difference between A and B and this results in the columns of B becoming closer to orthogonal. Numerical examples in Zou et al. (2006) indicate that sparse PCA produces more zero loadings than SCoTLASS. However, both techniques cannot accommodate block structures in the variables, as the numerical results in Section 3 suggest. Next, we introduce a variant of sparse PCA called sparse fused PCA (SFPCA) that addresses this issue.

2.2 Sparse Fused Loadings

Our proposal is based on solving the following optimization problem:

$$\begin{aligned} \min_{A,B} \quad & \|X - XBA^T\|_F^2 + \lambda_1 \sum_{k=1}^{K} \|\beta_k\|_1 + \lambda_2 \sum_{k=1}^{K} \sum_{s<t} |\rho_{s,t}| |\beta_{s,k} - sign(\rho_{s,t}) \beta_{t,k}| \ , \\ \text{subject to} \quad & A^T A = I_K \ , \end{aligned} \tag{7}$$

where $\|X - XBA^T\|_F^2 = \sum_{i=1}^{n} \|x_i - AB^T x_i\|_2^2$; $\rho_{s,t}$ denotes the sample correlation between variables X_s and X_t and $sign(\cdot)$ the sign function. The first penalty in (7) is the sum of l_1

norms of the K PC vectors. It aims to shrink the elements of the PC vectors to zero, thus ensuring sparsity of the resulting solution. The second penalty is a linear combination of K generalized *fusion* penalties. This penalty shrinks the difference between $\beta_{s,k}$ and $\beta_{t,k}$, if the correlation between variables X_s and X_t is positive; the higher the correlation, the heavier the penalty for on the difference of coefficients. If the correlation is negative, the penalty encourages $\beta_{s,k}$ and $\beta_{t,k}$ to have similar magnitudes, but different signs. It is natural to encourage the loadings of highly correlated variables to be close: two perfectly correlated variables with the same variance have equal loadings, so for highly correlated variables on the same scale pushing the loadings to the same value has the same effect as setting small regression coefficients to 0 in the lasso: fitted model accuracy is not affected much, but interpretation is improved and overfitting avoided.

The effect of the fusion penalty, due to the singularity property of the ℓ_1 norm, is that some terms in the sum are shrunken exactly to zero, resulting in some loadings having identical magnitudes. Therefore, the penalty aims at blocking the loadings into groups and "fusing" similar variables together for ease of interpretation. Finally, if $\rho_{s,t} = 0$ for any $|t - s| > 1$ and $\rho_{s,s+1}$ is a constant for all s, then the generalized fusion penalty reduces to the fusion penalty (Land and Friedman, 1996; Tibshirani et al., 2005).

Note that one can use other types of weights in the generalized fusion penalty, including partial correlations or other similarity measures Li and Li (2008).

2.3 Optimization of the Objective Function

We discuss next how to optimize the posited objective function. It is achieved through alternating optimization over A and B, analagously to the sparse PCA algorithm. Overall, the algorithm proceeds as follows.

The Algorithm

Step 1. Initialize $\widehat{A}$ by setting it to the ordinary PCA solution.

Step 2. Given A, minimizing the objective function (7) over B is equivalent to solving the following K separate problems:

$$\min_{\beta_k} \|Y_k^* - X\beta_k\|^2 + \lambda_1\|\beta_k\|_1 + \lambda_2 \sum_{s<t} |\rho_{s,t}||\beta_{s,k} - sign(\rho_{s,t})\beta_{t,k}| \tag{8}$$

where $Y_k^* = X\alpha_k$. The solution to (8) is nontrivial, and is discussed in Section 2.4. This step updates the estimate $\widehat{B}$.

Step 3. Given the value of B, minimizing (7) over A is equivalent to solving

$$\arg\min_A \|X - XBA^T\|^2 \ , \text{ subject to } \ A^TA = I_K \ . \tag{9}$$

The solution can be derived by a reduced rank Procrustes rotation (Zou et al., 2006). Specifically, we compute the singular value decomposition (SVD) of $X^TXB = UDV^T$ and the solution to (9) is given by $\widehat{A} = UV^T$. This step updates the estimate $\widehat{A}$.

Step 4. Repeat Steps 2-3 until convergence.

2.4 Estimation of B Given A

Objective function (8) can be solved by quadratic programming. However, this approach can be inefficient in practice; thus, we propose a more efficient algorithm — local quadratic approximation (LQA) (Fan and Li, 2001). This method has been employed in a number of variable selection procedures for regression and its convergence properties have been studied by Fan and Li (2001) and Hunter and Li (2005). The LQA method approximates the objective function locally via a quadratic form. Notice that

$$\begin{aligned} & \sum_{s<t} |\rho_{s,t}||\beta_{s,k} - sign(\rho_{s,t})\beta_{t,k}| \\ = & \sum_{s<t} \frac{|\rho_{s,t}|}{|\beta_{s,k} - sign(\rho_{s,t})\beta_{t,k}|}(\beta_{s,k} - sign(\rho_{s,t})\beta_{t,k})^2 \\ = & \sum_{s<t} |w_{s,t}^{(k)}|(\beta_{s,k} - sign(w_{s,t})\beta_{t,k})^2 \end{aligned} \tag{10}$$

where $w_{s,t}^{(k)} = \rho_{s,t}/|\beta_{s,k} - sign(\rho_{s,t})\beta_{t,k}|$ and consequently $sign(w_{s,t}^{(k)}) = sign(\rho_{s,t})$.

After some algebra, one can show that (10) can be written as $\beta^T L^{(k)} \beta$, where $L^{(k)} = D^{(k)} - W^{(k)}$, $W^{(k)} = (w_{s,t})_{p\times p}$ with diagonal elements equal to zero, and $D^{(k)} = diag(\sum_{t\neq 1}|w_{1,t}|, \ldots, \sum_{t\neq p}|w_{p,t}|)$.

Similarly, we have $\|\beta_k\|_1 = \sum_{j=1}^p |\beta_{j,k}| = \sum_{j=1}^p \omega_j^{(k)} \beta_{j,k}^2 = \beta^T \Omega^{(k)} \beta$, where $\omega_j^{(k)} = 1/|\beta_{j,k}|$ and $\Omega^{(k)} = diag(\omega_1^{(k)}, \ldots, \omega_p^{(k)})$. Then, (8) can be written as

$$\min_{\beta_k} \|Y_k^* - X\beta_k\|_2^2 + \lambda_1 \beta^T \Omega^{(k)} \beta + \lambda_2 \beta^T L^{(k)} \beta. \tag{11}$$

Notice that (11) takes the form of a least squares problem involving two generalized ridge penalties; hence, its closed form solution is given by

$$\widehat{\beta}_k = (X^TX + \lambda_1 \Omega^{(k)} + \lambda_2 L^{(k)})^{-1} X^T Y_k^*. \tag{12}$$

Notice that both $\Omega^{(k)}$ and $L^{(k)}$ depend on the unknown parameter β_k. Specifically, LQA iteratively updates β_k, $L^{(k)}$ and $\Omega^{(k)}$ as follows, which constitute Step 2 of the algorithm.

Step 2(a). Given $\widehat{\beta}_k$ from the previous iteration, update $\widehat{\Omega}^{(k)}$ and $\widehat{L}^{(k)}$.

Step 2(b). Given $\widehat{\Omega}^{(k)}$ and $\widehat{L}^{(k)}$, update $\widehat{\beta}_k$ by formula (12).

Step 2(c). Repeat Steps 2(a) and 2(b) until convergence.

Step 2(d). Scale $\widehat{\beta}_k$ to have unit l_2-norm.

Note that to calculate $L^{(k)}$ in step 2(a), we need to calculate $w_{s,t} = \rho_{s,t}/|\beta_{k,s} - sign(\rho_{s,t})\beta_{k,t}|$. When the values of $\beta_{k,s}$ and $sign(\rho_{s,t})\beta_{k,t}$ are extremely close, $w_{s,t}$ is numerically singular. In this case, we replace $|\beta_{k,s} - sign(\rho_{s,t})\beta_{k,t}|$ by a very small positive number (e.g. 10^{-10}); similarly, we replace $|\beta_{j,k}|$ by a very small positive number if its value is extremely close to 0.

With the new Step 2, the algorithm has two nested loops. However, the inner loop in Step 2 can be effectively approximated by a one step update (Hunter and Li, 2005), i.e., by removing step 2(c). In our numerical experiments, we found that this one step update can lead to significant computational savings without minor sacrifices in terms of numerical accuracy.

2.5 Selection of Tuning Parameters

The proposed procedure involves two tuning parameters. One can always use cross-validation to select the optimal values, but it can be computationally expensive. We discuss next an alternative approach for tuning parameter selection based on the Bayesian information criterion (BIC), which we use in simulations in Section 3. In general, we found solutions from cross-validation and BIC to be comparable, but BIC solutions tend to be sparser.

Let $A^{\lambda_1,\lambda_2} = [\alpha_1^{\lambda_1,\lambda_2}, \ldots, \alpha_K^{\lambda_1,\lambda_2}]$ and $B^{\lambda_1,\lambda_2} = [\beta_1^{\lambda_1,\lambda_2}, \ldots, \beta_K^{\lambda_1,\lambda_2}]$ be the estimates of A and B in (7), obtained using tuning parameters λ_1 and λ_2. Let $\widehat{\sigma}_\epsilon^2 = 1/n \sum_{i=1}^n \|X - X\widehat{A}\widehat{A}^T\|_F^2$, where the columns of $\widehat{A}$ contain the first K ordinary PC vectors of X. We define the BIC for sparse PCA as follows:

$$BIC(\lambda_1, \lambda_2) = \|X - XB^{\lambda_1,\lambda_2}(A^{\lambda_1,\lambda_2})^T\|_F^2/\widehat{\sigma}_\epsilon^2 + \log(n)df^{SPCA} \tag{13}$$

and analogously for SFPCA

$$BIC(\lambda_1, \lambda_2) = \|X - XB^{\lambda_1,\lambda_2}(A^{\lambda_1,\lambda_2})^T\|_F^2/\widehat{\sigma}_\epsilon^2 + \log(n)df^{SFPCA} \tag{14}$$

where df^{SPCA} and df^{SFPCA} denote the degrees of freedom of sparse and sparse-fused PCA defined as the number of all nonzero/nonzero-distinct elements in B^{λ_1,λ_2}, respectively. These

definitions are similar to *df* defined for Lasso and fused Lasso (Zou et al., 2007; Tibshirani et al., 2005).

2.6 Computational Complexity and Convergence

Since X^TX only depends on the data, it is calculated once and requires np^2 operations. The estimation of A by solving an SVD takes $O(pK^2)$. Calculation of Ω and L in (11) requires $O(p^2)$ operations, while the inverse in (12) is of order $O(p^3)$. Therefore, each update in LQA is of order $O(p^3K)$, and the total computational cost is $O(np^2) + O(p^3K)$.

The convergence of the algorithm essentially follows from standard results. Note that the loss function is strictly convex in both A and B, and the penalties are convex in B, and thus the objective function is strictly convex and has a unique global minimum. The integrations between Step 2 and Step 3 of the Algorithm amount to block coordinate descent, which is guaranteed to converge for differentiable convex functions (see, e.g., Bazaraa et al. (1993)). The original objective function has singularities, but the objective function (10) obtained from the local quadratic approximation that we are actually optimizing is differentiable everywhere, and thus the convergence of coordinate descent is guaranteed. Thus, we only need to make sure that each step of the coordinate descent is guaranteed to converge. In Step 3, we are optimizing the objective function (9) exactly and obtain the solution in closed form. In Step 2, the optimization is iterative, but convergence follows easily by adapting the arguments of Hunter and Li (2005) for local quadratic approximation obtained from general results for minorization-maximization algorithms.

3 Numerical illustration of SFPCA

First, we illustrate the performance of the proposed SFPCA method on a number of synthetic datasets described next.

Simulation 1

This simulation scenario is adopted from Zou et al. (2006). Three latent variables are generated as follows:

$$\begin{aligned} V_1 &\sim N(0, 290), \\ V_2 &\sim N(0, 300), \\ V_3 &= -0.3V_1 + 0.6V_2 + \epsilon, \end{aligned}$$

where V_1, V_2 and ϵ are independent, and $\epsilon \sim N(0, 1)$. Next, ten observable variables are constructed as follows:

$$X_j = \begin{cases} V_1 + e_j, & \text{if } 1 \le j \le 4; \\ V_2 + e_j, & \text{if } 5 \le j \le 8; \\ V_3 + e_j, & \text{if } j = 9, 10; \end{cases}$$

where $\epsilon_j, 1 \le j \le 10$ are i.i.d. $N(0, 1)$. The variance of the three latent variables are 290, 300 and 38, respectively. Notice that by construction, variables X_1 through X_4 form a block with a constant within-block pairwise correlation of .997 ("block 1"), while variables X_5 through X_8 and X_9, X_{10} form another two blocks ("block 2" and "block 3", respectively). Ideally, a sparse first PC should pick up block 2 variables with equal loadings, while a sparse second PC should consist of block 1 variables with equal loadings, since the variance of V_2 is larger than that of V_1.

Zou et al. (2006) compared sparse PCA with ordinary PCA and SCoTLASS using the true covariance matrix. In our simulation, we opted for the more realistic procedure of generating 20 samples according to the above description and repeated the simulation 50 times. PC vectors from ordinary PCA, sparse PCA and SFPCA were computed from these simulated datasets and the results are shown in Table 1, along with the ordinary PC vectors computed from the true covariance matrix. The table entries correspond to the median and the median absolute deviation (in parentheses) of the loadings over 50 replications. To measure the variation of the loadings within block 1 and 2, we also calculated the standard deviation among the loadings within these blocks and record their medians and median absolute deviations in rows "Block 1" and "Block 2", respectively. The proportions of adjusted variance and adjusted cumulative variance are reported as "AV (%)" and "ACV (%)". Adjusted variance was defined by Zou et al. (2006) as follows: let $\widehat{B}$ be the first K modified PC vectors. Using the QR decomposition,

we have $X\widehat{B} = QR$, where Q is orthonormal and R is upper triangular. Then the adjusted variance of the k-th PC equals $R_{k,k}^2$.

The tuning parameters were selected by minimizing the Bayesian information criterion (BIC) defined in Section 2.5, using a grid search over $\{2^{-10}, 2^{-9}, \ldots, 2^{10}\}$ for λ_1 and $\{10^{-3}, \ldots, 10^3\}$ for λ_2, respectively.

Table 1 shows that both SFPCA and sparse PCA recover the correct sparse structure of the loadings in the first two PC vectors. The median standard deviations within block 2 in PC 1 and block 1 in PC 2 equal to zero, which implies that SFPCA accurately recovers the loadings within the block. In contrast, the median standard deviations within block 2 in PC 1 and within block 1 in PC 2 reveal that the loadings estimated by sparse PCA exhibit significant variation.

Simulation 2

This example is a high-dimensional version ($p > n$) of simulation 1. We define

$$X_j = \begin{cases} V_1 + e_j, & \text{if } 1 \le j \le 20; \\ V_2 + e_j, & \text{if } 21 \le j \le 40; \\ V_3 + e_j, & \text{if } 41 \le j \le 50; \end{cases}$$

where $\epsilon_j, 1 \le j \le 50$ are i.i.d. $N(0,1)$. Then 20 samples were generated in each of the 50 repetitions. The factor loadings estimated from this simulation are illustrated in Figure 1. Sparse PCA and SFPCA produce similar sparse structures in the loadings. However, compared with the "jumpy" loadings from sparse PCA, the loadings estimated by SFPCA are smooth and easier for interpretation.

4 Application of SFPCA to Real Datasets

Drivers Dataset

This dataset provides information about the physical size and age of 38 drivers along with a response variable, seat position in a car. (Faraway, 2004). For the purposes of PCA, the response variable was excluded from the analysis. The eight available variables on driver characteristics are age, weight, height in shoes, height in bare feet, seated height, lower arm length, thigh length, and lower leg length. All height/length variables are highly correlated

(average correlation among these variables is about 0.8) and form a natural block; hence, we expect them to have similar loadings. SFPCA was applied to this dataset and compared its results with those obtained from ordinary PCA and sparse PCA (Table 2).

It can be seen that ordinary PCA captures the block structure in the first PC, but the factor loadings exhibit significant variation. Interestingly, the factor loadings from sparse PCA exhibit even greater variability, while the percentage of total variance explained by the first PC is only 55%, as opposed to 70% by ordinary PCA. On the other hand, SPFCA exhibits good performance in terms of goodness of fit (68.7%) and clearly reveals a single block structure in the "size" variables.

Pitprops Dataset

The pitprops dataset, introduced in Jeffers (1967), has become a classic example of the difficulties in interpretation of principal components. In this dataset, the sizes and properties of 180 pitprops (lumbers used to support the roofs of tunnels in coal mines) are recorded. The available variables are: the top diameter of the prop (topdiam), the length of the prop (length), the moisture content of the prop (moist), the specific gravity of the timber at the time of the test (testsg), the oven-dry specific gravity of the timber (ovensg), the number of annual rings at the top of the prop (ringtop), the number of annual rings at the base of the prop (ringbut), the maximum bow (bowmax), the distance of the point of maximum bow from the top of the prop (bowdist), the number of knot whorls (whorls), the length of clear prop from the top of the prop (clear), the average number of knots per whorl (knots) and the average diameter of the knots (diaknot). The first six PCs from regular PCA account for 87% of the total variability (measured by cumulative proportion of total variance explained).

We applied SPFCA and sparse PCA to the dataset and the results are given in Table 3. The loadings from SFPCA show a sparse structure similar to that of sparse PCA, but the first three PCs from SFPCA involve fewer variables than those of SPCA. The equal loadings within blocks assigned by SFPCA produce a clear picture for interpretation purposes. Referring to the interpretation in Jeffers (1967), the first PC gives the same loadings to "topdiam", "length", "ringbut", "bowmax", "bowdist" and "whorls" and provides a general measure of size; the second PC assigns equal loadings to "moist" and "testsg" and measures the degree of seasoning; the third PC, giving equal loadings to "ovensg" and "ringtop", accounts for the rate of the growth and the strength of the timber; the following three PCs represent "clear",

"knots" and "diaknot", respectively.

Meat Spectrum Data

In this section, we apply SFPCA to a dataset involving spectra obtained from meat analysis (Borggaard and Thodberg, 1992; Thodberg, 1996). In recent decades, spectrometry techniques have been widely used to identify the fat content in pork, because it has proved significantly cheaper and more efficient than traditional analytical chemistry methods. In this dataset, 215 samples were analyzed by a Tecator near-infrared spectrometer which measured the spectrum of light transmitted through a sample of minced pork meat. The spectrum gives the absorbance at 100 wavelength channels in the range of 850 to 1050 nm.

The adjusted cumulative total variances explained by the first two PCs from ordinary PCA, sparse PCA and SFPCA are 99.6%, 98.9% and 98.4%, respectively. Since wavelengths are naturally ordered, a natural way to display the loadings is to plot them against the wavelength. The plot of the first two PCs for the 100 wavelength channels is shown in Figure 3.

SFPCA smoothes the ordinary PC vectors producing piece-wise linear curves which are easier to interpret. The SFPCA results show clearly that the first PC represents the overall mean over different wavelengths while the second PC represents a contrast between the low and high frequencies. On the other hand, the high variability in the loadings produces by sparse PCA makes the PC curves difficult to interpret.

USPS Handwritten Digit Data

In this example, the three PCA methods are compared on the USPS handwritten digit data set (Hull, 1994). This data set was collected by the US Postal Service (USPS) and contains 11,000 gray scale digital images of the ten digits at 16×16 pixel resolution. We focused on the digit "3" and sampled 20 images at random, thus operating in a large p, small n setting. While BIC gave good results for most data sets we examined, for the USPS data it tended to under shrink the coefficient estimates. However, we found that cross-validation produced good results and was computationally feasible, so we used five- fold cross-validation to select the optimal tuning parameters for SPCA and SFPCA. The optimal tuning parameter for SPCA turned out to be equal to zero, so here SPCA coincides with ordinary PCA. The reconstructed images by the first and second principal components ("eigen-images") arranged in the original spatial order are shown in Figure 4. It can be seen that SFPCA achieves a fairly strong fusing

effect for the background pixels, thus producing a smoother, cleaner background image. This is confirmed by the results in Table 4 that give the proportion of distinct elements in the first two principal components for PCA and SFPCA. Notice that since PCA does not impose any sparsity or fusion, the resulting proportion is 100%, compared to those for SFPCA (35.5% and 22.7% for the first and second PCs, respectively).

5 Concluding Remarks

In this paper, a method is developed to estimate principal components that capture block structures in the variables, which aids in the interpretation of the data analysis results. To achieve this goal, the orthogonality requirement is relaxed and an ℓ_1 penalty is imposed on the norm of the PC vectors, as well as a "fusion" penalty driven by variable correlations. Application of the method to both synthetic and real data sets illustrates its advantages when it comes to interpretation.

The idea of sparse fused loadings is also applicable in a number of other unsupervised learning techniques, including canonical correlation and factor analysis, as well as regression analysis, classification techniques (e.g., LDA and SVM) and survival analysis (e.g., Cox model and Buckley-James model). We note that Daye and Jeng (2009) proposed a weighted fusion penalty for variable selection in a regression model. Unlike the generalized fusion penalty which penalizes the pairwise Manhattan distances between the variables, their method penalizes the pairwise Euclidean distances, and thus would not necessarily shrink the coefficients of highly correlated variables to identical values. Similarly, Tutz and Ulbricht (2009) proposed a BlockBoost method, whose penalty also tends to fuse the pairwise difference between the regression coefficients. In particular, when these pairwise correlations are close to ± 1, the solution of BlockBoost is closed to that of Daye and Jeng (2009).

Acknowledgments: The authors would like to thank the Editor, the Associate Editor and two referees for helpful comments and suggestions.

References

Bazaraa, M., Sherali, H., and Shetty, C. (1993), *Nonlinear programming: theory and algorithms*, John Wiley & Sons, New York.

Borggaard, C. and Thodberg, H. (1992), "Optimal minimal neural interpretation of spectra," *Analytic Chemistry*, 64, 545–551.

Daye, Z. and Jeng, X. (2009), "Shrinkage and model selection with correlated variables via weighted fusion," *Computational Statistics and Data Analysis*, 53, 1284–1298.

Fan, J. and Li, R. (2001), "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Asscociation*, 96, 1348–1360.

Faraway, J. (2004), *Linear model in R*, CRC Press.

Hull, J. (1994), "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 550–554.

Hunter, D. and Li, R. (2005), "Variable selection using MM algorithms," *Annals of Statistics*, 33, 1617–1642.

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

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

Jollife, I., Trendafilov, N., and Uddin, M. (2003), "A modified principal component technique based on the LASSO," *Journal of Computational and Graphical Statistics*, 12, 531–547.

Land, S. and Friedman, J. (1996), "Variable fusion: a new method of adaptive signal regression," Tech. rep., Department of Statistics, Stanford University, Stanford.

Li, C. and Li, H. (2008), "Network-constraint regularization and variable selection for analysis of genomic data," *Bioinformatics*, 24, 1175–1182.

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

Thodberg, H. (1996), "A review of Bayesian neural networks with an application to nearinfrared spectroscopy," *IEEE Transactions on Neural Networks*, 7, 56–72.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005), "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society, Series B*, 67, 91–108.

Tutz, G. and Ulbricht, J. (2009), "Penalized regression with correlation-based penalty," *Statistics and Computing*, To appear.

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.

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

— (2007), "On the degrees of freedom of the LASSO," *Annals of Statistics*, 35, 2173–2192.

Table 1: Results for simulation 1. "PCA-T" corresponds to the ordinary PCA estimation from the true covariance matrix. "PCA-S" corresponds to the ordinary PCA estimation from the sample covariance matrix. "SPCA" represents the sparse PCA, and "SFPCA" represents the sparse fused PCA. "AV" is the adjusted variance, and "ACV" is the adjusted cumulative variance. The row"Block 1" shows the standard deviation of the loadings of variables 1 to 4, and "Block 2" shows the same for variables 5 to 8. In each row, the top entry is the median and the bottom entry in parentheses is the median absolute deviation over 50 replications.

Loadings	PC 1				PC 2			
	PCA-T	PCA-S	SPCA	SFPCA	PCA-T	PCA-S	SPCA	SFPCA
1	0.055	-0.123	0	0	0.488	0.447	0.506	0.500
	(—)	(0.162)	(0)	(0)	(—)	(0.032)	(0.072)	(0)
2	0.055	-0.127	0	0	0.488	0.444	0.492	0.500
	(—)	(0.161)	(0)	(0)	(—)	(0.031)	(0.085)	(0)
3	0.055	-0.129	0	0	0.488	0.448	0.491	0.500
	(—)	(0.161)	(0)	(0)	(—)	(0.033)	(0.085)	(0)
4	0.055	-0.125	0	0	0.488	0.442	0.493	0.500
	(—)	(0.159)	(0)	(0)	(—)	(0.032)	(0.089)	(0)
5	-0.453	0.376	0.422	0.487	0.089	0.164	0	0
	(—)	(0.040)	(0.021)	(0.015)	(—)	(0.131)	(0)	(0)
6	-0.453	0.374	0.415	0.487	0.089	0.165	0	0
	(—)	(0.038)	(0.021)	(0.016)	(—)	(0.133)	(0)	(0)
7	-0.453	0.375	0.417	0.487	0.089	0.161	0	0
	(—)	(0.040)	(0.019)	(0.015)	(—)	(0.133)	(0)	(0)
8	-0.453	0.376	0.417	0.487	0.089	0.159	0	0
	(—)	(0.038)	(0.020)	(0.015)	(—)	(0.127)	(0)	(0)
9	-0.289	0.389	0.382	0.155	-0.093	-0.015	0	0
	(—)	(0.025)	(0.021)	(0.122)	(—)	(0.132)	(0)	(0)
10	-0.289	0.389	0.388	0.155	-0.093	-0.009	0	0
	(—)	(0.026)	(0.027)	(0.119)	(—)	(0.127)	(0)	(0)
Block 1	0	0.003	0	0	0	0.002	0.064	0
	(—)	(0.003)	(0)	(0)	(—)	(0.002)	(0.050)	(0)
Block 2	0	0.001	0.014	0	0	0.004	0	0
	(—)	(0.001)	(0.014)	(0)	(—)	(0.003)	(0)	(0)
AV (%)	42.7	61.9	57.6	47.3	40.3	37.7	37.1	36.7
	(—)	(4.4)	(1.0)	(6.3)	(—)	(4.2)	(2.2)	(1.5)
ACV (%)	42.7	61.9	57.6	47.3	83.0	99.5	95.1	83.7
	(—)	(4.4)	(1.0)	(6.3)	(—)	(0.1)	(2.7)	(6.1)

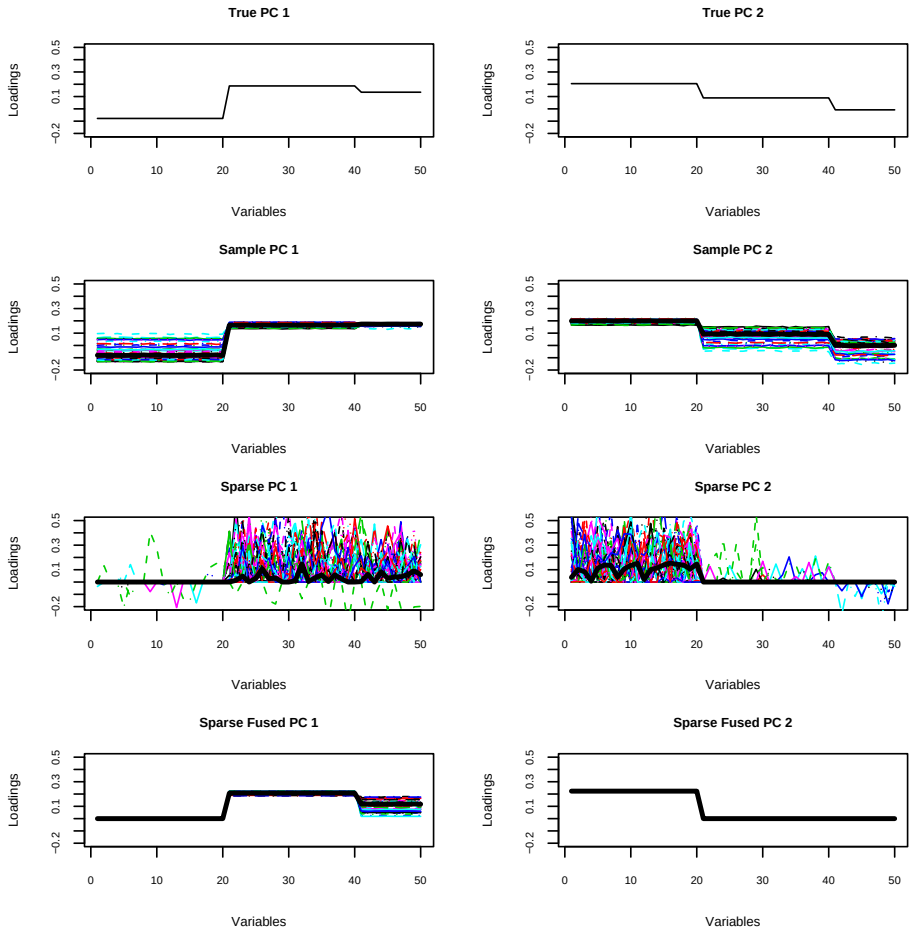

Figure 1: Factor loadings of the first (left column) and second (right column) PC vectors estimated by ordinary PCA from the true covariance (first row), ordinary PCA from the sample covariance (second row), sparse PCA (third row) and SFPCA (fourth row). Each colored curve represents the PC vector in one replication. The median loadings over 50 repetitions are represented by the black bold lines.

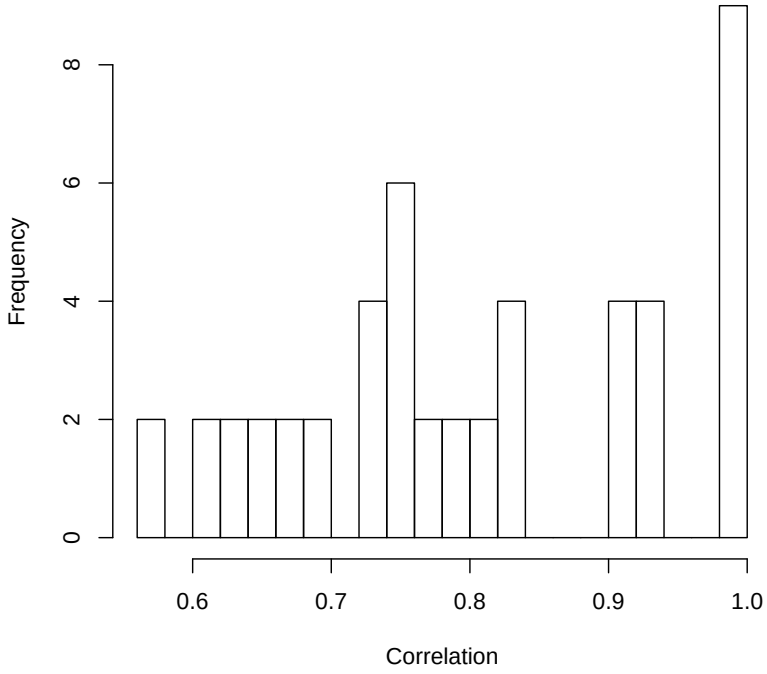

Figure 2: The histogram of the pairwise correlations between the height/length variables: weight, height in shoes, height in bare feet, seated height, lower arm length, thigh length, and lower leg length.

Table 2: Numerical results for the drivers example.

Variables	PC 1			PC 2		
	PCA	SPCA	SFPCA	PCA	SPCA	SFPCA
Age	0.007			0.876	0.970	1.000
Weight	0.367	0.284	0.378	0.045		
HtShoes	0.411	0.139	0.378	-0.106		
Ht	0.412	0.764	0.378	-0.112		
Seated	0.381	0.313	0.378	-0.218		
Arm	0.349	0.208	0.378	0.374	0.242	
Thigh	0.328	0.247	0.378	0.125		
Leg	0.390	0.341	0.378	-0.056		
AV (%)	70.9	55.0	68.7	15.5	14.2	12.2
ACV (%)	70.9	55.0	68.7	86.4	69.2	80.8

Table 3: Numerical results for the pitprops example.

Variables	PC 1			PC 2			PC 3		
	PCA	SPCA	SFPCA	PCA	SPCA	SFPCA	PCA	SPCA	SFPCA
topdiam	0.404	0.477	0.408	0.218			-0.207		
length	0.406	0.476	0.408	0.186			-0.235		
moist	0.124			0.541	0.785	0.707	0.141		
testsg	0.173			0.456	0.620	0.707	0.352		
ovensg	0.057	-0.177		-0.170			0.481	0.640	0.707
ringtop	0.284		0.052	-0.014			0.475	0.589	0.707
ringbut	0.400	0.250	0.408	-0.190			0.253	0.492	
bowmax	0.294	0.344	0.408	-0.189	-0.021		-0.243		
bowdist	0.357	0.416	0.408	0.017			-0.208		
whorls	0.379	0.400	0.408	-0.248			-0.119		
clear	-0.011			0.205			-0.070		
knots	-0.115			0.343	0.013		0.092	-0.015	
diaknot	-0.113			0.309			-0.326	-0.308519	
AV (%)	32.4	28.0	31.5	18.3	14.4	15.1	14.4	13.3	10.1
ACV (%)	32.4	28.0	31.5	50.7	42.0	46.6	65.1	55.3	56.7

Variables	PC 4			PC 5			PC 6		
	PCA	SPCA	SFPCA	PCA	SPCA	SFPCA	PCA	SPCA	SFPCA
topdiam	-0.091			0.083			0.120		
length	-0.103			0.113			0.163		
moist	0.078			-0.350			-0.276		
testsg	0.055			-0.356			-0.054		
ovensg	0.049			-0.176			0.626		
ringtop	-0.063			0.316			0.052		
ringbut	-0.065			0.215			0.003		
bowmax	0.286			-0.185			-0.055		
bowdist	0.097			0.106			0.034		
whorls	-0.205			-0.156			-0.173		
clear	0.804	1.000	1.000	0.343			0.175		
knots	-0.301			0.600	1.000	1.000	-0.170		
diaknot	-0.303			-0.08			0.626	1.000	1.000
AV (%)	8.5	7.4	8.0	7.0	6.8	7.3	6.3	6.2	7.0
ACV (%)	73.6	62.7	64.7	80.6	69.5	72.0	86.9	75.8	79.0

Table 4: The proportion of distinct elements in the eigen-images of digit "3" estimated by PCA and SFPCA, respectively.

PC	PCA (%)	SFPCA (%)
1	100	35.5
2	100	22.7

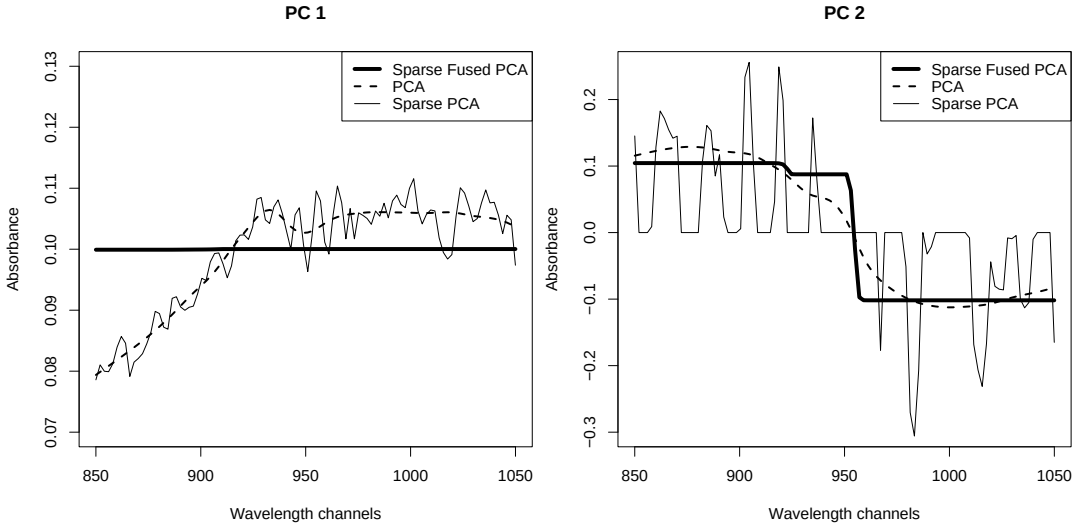

Figure 3: Comparison of the first (left panel) and second (right panel) PC vectors from ordinary PCA (dashed line), sparse PCA (dotted line) and SFPCA (solid line).

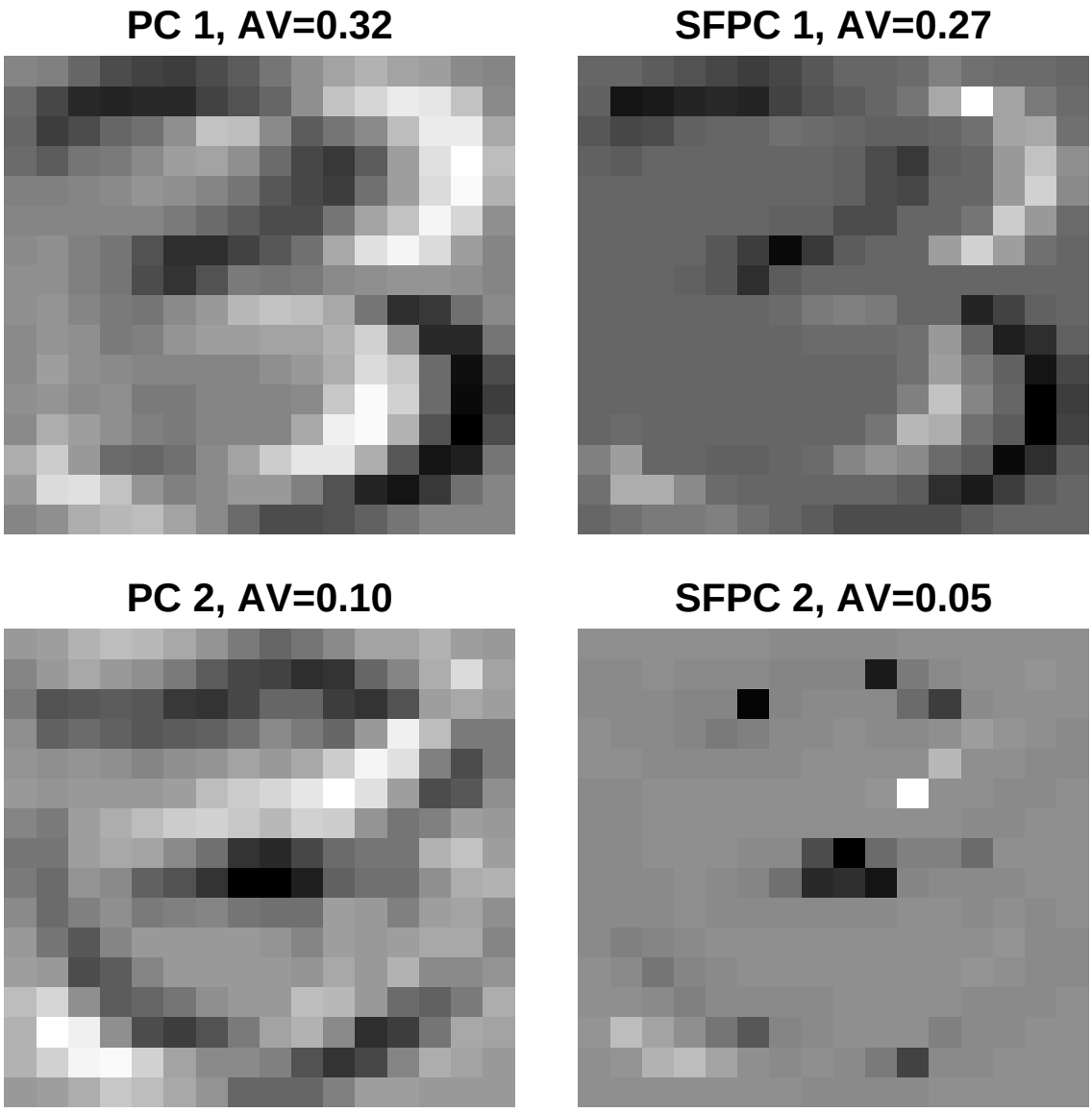

Figure 4: The first two eigen-images of digit "3" estimated by PCA and SFPCA, respectively.