Journal of Statistical Planning and Inference 175 (2016) 87–99

Contents lists available at ScienceDirect

Journal of Statistical Planning and Inference

journal homepage: www.elsevier.com/locate/jspi

Sparse principal component analysis with measurement errors

Jianhong Shi [a], Weixing Song [b,*]

[a] School of Mathematics and Computer Science, Shanxi Normal University, Linfen, Shanxi, 041000, China

[b] Department of Statistics, Kansas State University, Manhattan, KS 66503, United States

ARTICLE INFO

Article history:
Received 4 July 2015
Received in revised form 11 March 2016
Accepted 12 March 2016
Available online 18 March 2016

Keywords:
Lasso
Elastic net
Sparse principal component analysis
Measurement error
Bias correction

ABSTRACT

Traditional principal component analysis often produces non-zero loadings, which makes it hard to interpret the principal components. This drawback can be overcome by the sparse principal component analysis procedures developed in the past decade. However, similar work has not been done when the random variables or vectors are contaminated with measurement errors. Simply applying the existing sparse principal component analysis procedure to the error-contaminated data might lead to biased loadings. This paper tries to modify an existing sparse principal component procedure to accommodate the measurement error setup. Similar to error-free cases, we show that the sparse principal component for the latent variables can be formulated as a bias-corrected lasso (elastic net) regression problem based on the observed surrogates, efficient algorithms are also developed to implement the procedure. Numerical simulation studies are conducted to illustrate the finite sample performance of the proposed method.

1. Introduction

As an efficient dimensional reduction technique, principal component analysis (PCA) provides a sequence of orthogonal linear combinations of the random variables, called principal components (PCs), from a multi-dimensional random vector, which can sequentially capture the biggest variability among the data collected from the random vector. Through these PCs, one can extract the commonality contained in the vector, and hopefully, an informative explanation might follow. Many interesting applications of PCA can be found in the areas of engineering, biology, education and other social science, for example, the handwritten zip code classification in Hastie et al. (2001), the human face recognition in Hancock et al. (1996), and the gene expression data analysis in Alter et al. (2000), just name a few. However, the entries in the loading vectors usually are nonzero which makes the interpretation of the PCs difficult. To overcome this drawback, Henry (1958) proposed the famous rotation technique; Gorsuch (1983) recommended rotating with varimax to produce orthogonal PCs or promax to produce oblique PCs. For more information on determining the proper rotations, see Tabachnick and Fidell (2007); By restricting the loading values to be 0, 1, and −1 or other values, Vines (2000) proposed a simple principal component analysis; By imposing a L_1 constraint on the loading vectors directly, Jolliffe and Uddin (2003) proposed the SCoTLASS method. As noted in Zou et al. (2006), the SCoTLASS technique suffers from the high computational cost, and insufficient sparsity of loadings when a high percentage of explained variance is required. Being aware of that the PCA can be reformulated as a ridge regression problem, Zou et al. (2006) proposed a modified Sparse PCA (SPCA) by integrating

* Corresponding author.
E-mail address: weixing@ksu.edu (W. Song).

http://dx.doi.org/10.1016/j.jspi.2016.03.001
0378-3758/

the elastic net approach with the lasso procedure, to produce sparse loadings. In addition to its attractive regression type optimization idea, the popularity of the SPCA is enhanced by its very efficient algorithm. Lasso and elastic net are very popular variable selection procedures in high dimensional modeling, we will not introduce them here for the sake of brevity. More details on these methodology can be found in Tibshirani (1996), Efron et al. (2004), Zou and Hastie (2005), Zou and Trevor (2005) and the references therein.

In the following discussion, we shall use the bold capital letter $\mathbf{X}$ to denote a $n \times p$ data matrix, its ith row is denoted by $\mathbf{x}_i$, $i = 1, 2, \ldots, n$, which will be viewed as a random sample from some population $\mathbf{x}$. The jth column of $\mathbf{X}$ will be denoted by X_j, $j = 1, 2, \ldots, p$. Often times, the quantity of interest $\mathbf{x}$ cannot be observed directly in practice, which is often called the latent variables or vector. Instead, a surrogate $\mathbf{z}$ can be observed, which is related to $\mathbf{x}$ in an additive way $\mathbf{z} = \mathbf{x} + \mathbf{u}$, where $\mathbf{u}$ is called the measurement error, $\mathbf{x}$ and $\mathbf{u}$ are independent. For some introduction on measurement error modeling, see Fuller (1987) and Carroll et al. (2006). Clearly, the PCA based on $\mathbf{x}$ is not feasible in this scenario. If we simply apply the existing SPCA procedure to the error-contaminated data, it might lead to biased loadings, just like the naive estimates in errors-in-variables regression models. Therefore, an interesting question is how to identify the loadings for $\mathbf{x}$ based on the sample from $\mathbf{z}$ and some additional information from $\mathbf{u}$. In the measurement error setup, the covariance matrix $\Sigma_{\mathbf{u}}$ of $\mathbf{u}$ is often assumed to be known. In the case of $\Sigma_{\mathbf{u}}$ being unknown, replicated observations on $\mathbf{x}$ are often used to obtain a consistent estimate of $\Sigma_{\mathbf{u}}$. More discussion on this case can be found in Section 3. In this paper, we will try to extend the SPCA of Zou et al. (2006) to the measurement error setup under the assumption of known $\Sigma_{\mathbf{u}}$. Throughout the paper, for any generic random vector $\mathbf{a}$, $\Sigma_{\mathbf{a}}$ denotes the population covariance of $\mathbf{a}$.

Note that the additive structure and the independence imply $\Sigma_{\mathbf{z}} = \Sigma_{\mathbf{x}} + \Sigma_{\mathbf{u}}$, so to find PCs for $\Sigma_{\mathbf{x}}$, one can directly work on $\Sigma_{\mathbf{z}} - \Sigma_{\mathbf{u}}$, provided the latter is known. However, this is rarely the case in practice. The PCA based on the sample covariance matrix is not as natural as the error-free situation simply because we do not know the sample covariance matrix of the latent vector $\mathbf{x}$. If we denote $S_{\mathbf{ab}} = n^{-1}\sum_{i=1}^{n}(\mathbf{a}_i - \bar{\mathbf{a}})^T(\mathbf{b}_i - \bar{\mathbf{b}})$, where $\bar{\mathbf{a}}$, $\bar{\mathbf{b}}$ are the mean vectors from the corresponding sequences, then simple algebra gives $S_{\mathbf{zz}} = S_{\mathbf{xx}} + S_{\mathbf{xu}} + S_{\mathbf{ux}} + S_{\mathbf{uu}}$. By subtracting $\Sigma_{\mathbf{u}}$ from both sides, we have $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}} = S_{\mathbf{xx}} + S_{\mathbf{xu}} + S_{\mathbf{ux}} + S_{\mathbf{uu}} - \Sigma_{\mathbf{u}}$. Since $S_{\mathbf{xu}} + S_{\mathbf{ux}} + S_{\mathbf{uu}} - \Sigma_{\mathbf{u}}$ converges to 0 at the rate of $1/\sqrt{n}$ under quite general assumptions, so we can expect that the PCs based on $S_{\mathbf{xx}}$ could be well approximated by the PCs based on $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$, but the effect of removing $S_{\mathbf{xu}} + S_{\mathbf{ux}} + S_{\mathbf{uu}} - \Sigma_{\mathbf{u}}$ from analysis on the resulting PCs should be investigated when the sample size is small. To adapt Zou et al. (2006)'s SPCA to our current setup, we have to find a way to transform the problem to a penalized linear regression problem.

The paper is organized as follows. Section 2 discusses the PCA based on population covariance matrices, a simple argument and some numerical examples are presented to show that PCA based on the covariance matrix of the surrogates often leads to biased PCs, but in a particular case, the PCs obtained from the matrices of the surrogates and the latent variables are the same! The direct bias-corrected SPCA approximation is introduced in Section 3, followed by the efficient algorithm developed for the proposed procedure, as well as some remarks on the adjusted total variances, the computational complexity of the algorithm and how to apply the proposed method when $\Sigma_{\mathbf{u}}$ is unknown but replicated observations are available. Numerical studies are conducted in Section 4, and all the theoretical derivations are postponed to Appendix.

2. PCA based on population covariance matrices

Before we work on the sample covariance matrices, it might be more illuminating to investigate the effect of measurement errors on the PCA based on the population covariance matrices. Note that $\Sigma_{\mathbf{z}} = \Sigma_{\mathbf{x}} + \Sigma_{\mathbf{u}}$, the PC directions of $\mathbf{x}$ might not be the same as those of $\mathbf{z}$ because of the perturbation of the measurement error. However, if $\Sigma_{\mathbf{u}}$ is diagonal and all the diagonal entries are equal, then the PC directions of $\mathbf{x}$ and $\mathbf{z}$ are indeed the same. To see this point, assume that $\Sigma_{\mathbf{u}} = \sigma^2 I$, and the spectral decomposition of $\Sigma_{\mathbf{x}}$ is $\mathbf{QMQ}^T$, where $\mathbf{M} = \mathrm{diag}(m_k^2)$. Then we must have

$$\mathbf{Q}^T\Sigma_{\mathbf{z}}\mathbf{Q} = \mathbf{Q}^T\Sigma_{\mathbf{x}}\mathbf{Q} + \sigma^2\mathbf{Q}^T\mathbf{Q} = \mathrm{diag}(m_j^2 + \sigma^2).$$

While maintaining the direction of principal components, the above result also implies the magnitudes along the principal components are inflated by an additive factor σ^2. For the general case, the eigenvalue–eigenvector relationship between $\Sigma_{\mathbf{z}}$ and $\Sigma_{\mathbf{x}}$ becomes complicated, however, $\Sigma_{\mathbf{z}} - \Sigma_{\mathbf{u}} = \Sigma_{\mathbf{x}}$ suggests us to study the PCA of $\Sigma_{\mathbf{x}}$, one can study the PCA of $\Sigma_{\mathbf{z}} - \Sigma_{\mathbf{u}}$. For illustration purpose, we choose

$$\Sigma_{\mathbf{x}} = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}, \qquad \Sigma_{\mathbf{u}}^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \Sigma_{\mathbf{u}}^{(2)} = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}, \qquad \Sigma_{\mathbf{u}}^{(3)} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}.$$

Accordingly, let $\Sigma_{\mathbf{z}}$ be $\Sigma_{\mathbf{x}} + \Sigma_{\mathbf{u}}^{(j)}$ with $j = 1, 2, 3$, that is, the latent vector $\mathbf{x}$ is contaminated with three different measurement errors. Fig. 1 shows the principal components of $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{z}}$ with $\Sigma_{\mathbf{u}}$ defined above. It is easy to see that the principal components of $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{z}}$ are the same for $\Sigma_{\mathbf{u}}^1$, and different for the latter two cases.

Knowing the covariance matrix $\Sigma_{\mathbf{z}}$ is an ideal case. More realistically, the observations for $\mathbf{z}$ are available, therefore the principal component analysis should be based on $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$, where $S_{\mathbf{zz}}$ is the sample covariance matrix of $\mathbf{z}_i$, $i = 1, 2, \ldots, n$. Note that the bias-corrected statistic $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$ is a consistent estimator of $\Sigma_{\mathbf{x}}$, so when the sample size is small, the performance of the principal component analysis based on $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$ may not be very satisfying and the results should be cautiously interpreted. In particular, in the finite sample cases or if the dimension p is larger than the sample size n, $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$

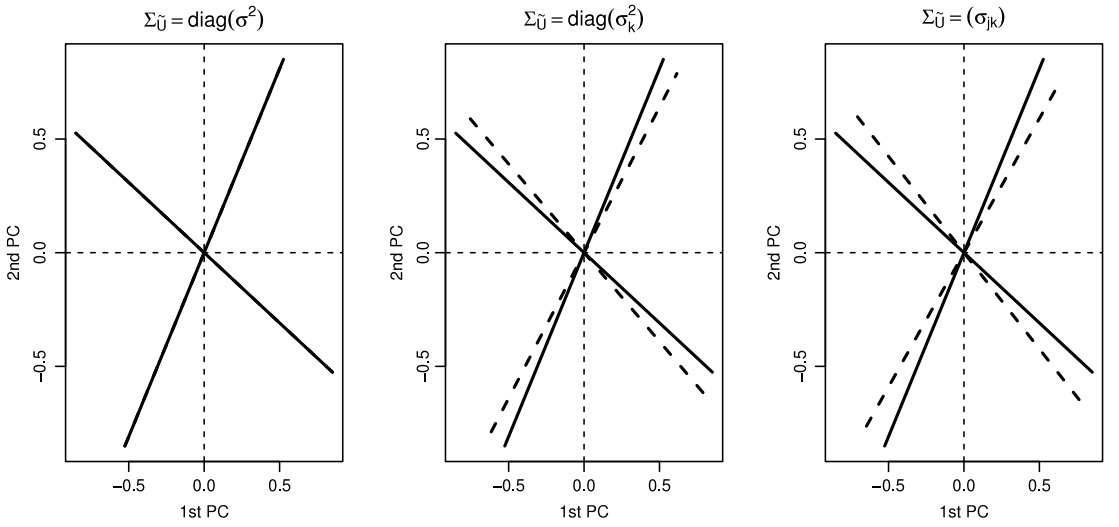

Fig. 1. PCs of $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{z}}$. Solid lines are the PCs of $\Sigma_{\mathbf{x}}$, and dashed lines are the PCs of $\Sigma_{\mathbf{z}}$.

can be negative definite, which presents a serious challenge to the proposed methodology, since the principal component analysis is only discussed among the nonnegative definite matrices. If $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$ is indeed negative definite, it might be worth to try working on its nonnegative definite projection, but the theoretical and practical implication by doing so should be carefully investigated. Due to these concerns, we shall assume that $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$ stays nonnegative definite throughout the paper.

Without loss of generality, each columns in $\mathbf{Z}$ will be centered at 0, so the covariance matrix $S_{\mathbf{zz}} = n^{-1}\mathbf{Z}^T\mathbf{Z}$. One may directly apply the SCoTLASS method proposed by Jolliffe and Uddin (2003) to obtain sparse loadings. However, as Zou et al. (2006) pointed out, the SCoTLASS does not have much guidance in choosing the constraint value, and high computational cost should be paid to find a proper one from a grid of candidate values. Moreover, some numerical studies show that the loadings obtained by SCoTLASS are not sparse enough when a high percentage of explained variance is required. In the following section, we will extend Zou et al. (2006)'s SPCA method to the measurement error setup.

3. Direct sparse approximations

When $\mathbf{x}$ is directly observable, Zou et al. (2006) noted that since each PC is a linear combination of the p variables, thus its loading can be recovered by regressing the PC on the p variables. In fact, for each PC Y, the normalized ridge estimate $\hat{\boldsymbol{\beta}}$ given by

$$\hat{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta}} \left\{ \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2 \right\}$$

for some positive λ reproduces the corresponding loadings. In order to adjust the bias, the argument for measurement error setup is slightly different from Zou et al.'s case. Let the spectral decomposition of $\mathbf{Z}^T\mathbf{Z} - \Sigma_{\mathbf{u}}$ be $\mathbf{V}\mathbf{D}^2\mathbf{V}^T$, where $\mathbf{V}$ is an orthonormal matrix, $\mathbf{D} = \mathrm{diag}(d_j)$, $d_j^2, j = 1, 2, \ldots, p$ are the eigenvalues of $\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}}$, and $\mathbf{V} = [V_1, \ldots, V_p]$. Then we define

$$\hat{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta}} \left\{ \|\mathbf{Z}V_j - \mathbf{Z}\boldsymbol{\beta}\|^2 - n\|V_j - \boldsymbol{\beta}\|^2_{\Sigma_{\mathbf{u}}} + \lambda\|\boldsymbol{\beta}\|^2 \right\}, \tag{1}$$

where $\|\mathbf{a}\|^2_{\mathbf{A}} = \mathbf{a}^T\mathbf{A}\mathbf{a}$ for any proper vector $\mathbf{a}$ and symmetric matrix $\mathbf{A}$. We can show that $\hat{\boldsymbol{\beta}}/\|\hat{\boldsymbol{\beta}}\| = V_j$.

To achieve sparsity, L_1 penalty is added to (1), the optimization problem becomes

$$\hat{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta}} \left\{ \|\mathbf{Z}V_j - \mathbf{Z}\boldsymbol{\beta}\|^2 - n\|V_j - \boldsymbol{\beta}\|^2_{\Sigma_{\mathbf{u}}} + \lambda\|\boldsymbol{\beta}\|^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \right\}. \tag{2}$$

The normalized solution $\hat{V}_j = \hat{\boldsymbol{\beta}}/\|\hat{\boldsymbol{\beta}}\|$ will be used to approximate V_j, the jth loading vector of $\mathbf{x}$.

The criterion based on (2) depends on the results of PCA of $\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}}$ explicitly. Similar to Zou et al. (2006)'s criterion (3.5), this type of criterion is not a genuine alternative to SCoTLASS. We shall present a self-contained regression type criterion to derive the PCs which is the counterpart of Theorem 2 in Zou et al. (2006).

Theorem 1. *For any* $\lambda > 0$, *let*

$$(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) = \mathrm{argmin}_{\boldsymbol{\alpha},\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \|\mathbf{z}_i - \boldsymbol{\alpha}\boldsymbol{\beta}^T\mathbf{z}_i\|^2 - n\boldsymbol{\beta}^T\Sigma_{\mathbf{u}}\boldsymbol{\beta} + 2n\boldsymbol{\beta}^T\Sigma_{\mathbf{u}}\boldsymbol{\alpha} + \lambda\|\boldsymbol{\beta}\|^2 \right\}$$

subject to $\|\boldsymbol{\alpha}\|^2 = 1$. *Then* $\hat{\boldsymbol{\beta}} \propto V_1$.

In the above theorem, $\mathbf{a} \propto \mathbf{b}$ means the vector $\mathbf{a}$ is the same as the vector $\mathbf{b}$ except for a constant. The following theorem is an extension of Theorem 2 which recovers the whole sequence of PCs of $\mathbf{Z}^T\mathbf{Z} - \Sigma_{\mathbf{u}}$.

Theorem 2. *Let* $\mathbf{A}_{p\times k} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ *and* $\mathbf{B}_{p\times k} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k)$. *For any* $\lambda > 0$, *let*

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \text{argmin}_{\mathbf{A},\mathbf{B}} \left\{ \|\mathbf{Z} - \mathbf{ZBA}^T\|^2 - n \cdot Tr\left[(I - \mathbf{AB}^T - \mathbf{BA}^T - \mathbf{BB}^T)\Sigma_{\mathbf{u}}\right] + \lambda\|\mathbf{B}\|^2 \right\}$$

subject to $\mathbf{A}^T\mathbf{A} = I_{k\times k}$. *Then* $\hat{\boldsymbol{\beta}}_j \propto V_j$ *for* $j = 1, \ldots, k$.

The above theorems successfully transform a PCA problem to a ridge regression problem, thus a sparse PC loading can be achieved by adding a L_1 penalty to the target functions defined in Theorem 1 or 2. For example, if we want to have k sparse loading vectors, then we can consider the following optimization problem

$$\text{argmin}_{\mathbf{A},\mathbf{B}} \left\{ \|\mathbf{Z} - \mathbf{ZBA}^T\|^2 - n \cdot \text{Tr}\left[(I - \mathbf{AB}^T - \mathbf{BA}^T - \mathbf{BB}^T)\Sigma_{\mathbf{u}}\right] + \lambda\|\mathbf{B}\|^2 + \|\mathbf{B}\Lambda\| \right\} \tag{3}$$

subject to $\mathbf{A}^T\mathbf{A} = I_{k\times k}$, where $\Lambda = \text{diag}(\lambda_{1,j})_{p\times p}$, and $\lambda_{1,j} > 0$ for all $j = 1, \ldots, p$. Denote $\tilde{\mathbf{B}} = (\tilde{\boldsymbol{\beta}}_1, \ldots, \tilde{\boldsymbol{\beta}}_k)$ as the solution, then $\tilde{V}_j = \tilde{\boldsymbol{\beta}}_j/\|\tilde{\boldsymbol{\beta}}_j\|$ is called an approximation to V_i, and $\mathbf{X}\tilde{V}_j$ is the jth approximated principle component score of $\mathbf{X}$.

In order to develop an efficient algorithm for the optimization problem (2), we need the following theorem.

Theorem 3. *For any fixed* $\mathbf{B}$, *consider the constrained minimization problem*

$$\hat{\mathbf{A}} = \text{argmin}_{\mathbf{A}} \left\{ \|\mathbf{Z} - \mathbf{ZBA}^T\|^2 + n \cdot Tr\left[(\mathbf{AB}^T + \mathbf{BA}^T)\Sigma_{\mathbf{u}}\right] \right\}$$

subject to $\mathbf{A}^T\mathbf{A} = I_{k\times k}$. *Suppose the SVD of* $(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}})\mathbf{B}$ *is* $\mathbf{UDV}^T$, *then* $\hat{\mathbf{A}} = \mathbf{UV}^T$.

From the proof of Theorem 2, we note that the target function in Theorem 2 can be rewritten as a sum of two terms, one term only depends on $\mathbf{A}$, and the other term has the following form

$$\sum_{j=1}^{k} \left[(\boldsymbol{\alpha}_j - \boldsymbol{\beta}_j)^T(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}})(\boldsymbol{\alpha}_j - \boldsymbol{\beta}_j) + \lambda\|\boldsymbol{\beta}_j\|^2 + \lambda_{1,j}\|\boldsymbol{\beta}_j\|_1\right].$$

Denote $\mathbf{M} = (\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}})^{1/2}$, and $\mathbf{Y}_j = \mathbf{M}\boldsymbol{\alpha}_j$, then we can rewrite the above term as

$$\sum_{j=1}^{k} \left[\|\mathbf{Y}_j - \mathbf{M}(\boldsymbol{\beta}_j)^T\|^2 + \lambda\|\boldsymbol{\beta}_j\|^2 + \lambda_{1,j}\|\boldsymbol{\beta}_j\|_1\right].$$

Thus, an alternating algorithm to solve the optimization problem (3) can proceed as follows

(1) Choose an initial value for $\hat{\mathbf{A}}$;
(2) For a given $\hat{\mathbf{A}}$, solve $\hat{\mathbf{B}} = (\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_k)$ through the following elastic net estimation procedure

$$\hat{\boldsymbol{\beta}}_j = \text{argmin}_{\boldsymbol{\beta}_j} \left[\|\mathbf{Y}_j - \mathbf{M}(\boldsymbol{\beta}_j)^T\|^2 + \lambda\|\boldsymbol{\beta}_j\|^2 + \lambda_{1,j}\|\boldsymbol{\beta}_j\|_1\right];$$

(3) For a given $\hat{\mathbf{B}}$, solve $\hat{\mathbf{A}}$ based on Theorem 3. That is, $\hat{\mathbf{A}} = \mathbf{UV}^T$, where $\mathbf{UDV}^T$ is the SVD of $(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}})\hat{\mathbf{B}}$;
(4) Repeat steps (2) and (3) alternately until convergence is achieved
(5) Define $\hat{V}_j = \hat{\boldsymbol{\beta}}_j/\|\hat{\boldsymbol{\beta}}_j\|, j = 1, 2, \ldots, k$ as the PCs.

We end this section with some remarks.

Remark 1 (*Adjusted Total Variance*)**.** Similar to Jolliffe and Uddin (2003) and Zou et al. (2006), the PCs obtained here do not satisfy the uncorrelated property. In the error-free case, let $\mathbf{P} = \mathbf{X}\hat{\mathbf{V}}$ be the modified PCs, Zou et al. (2006) noted that using $\text{Tr}(\mathbf{P}^T\mathbf{P})$ to represent the total variance is too optimistic. To account for the correlations among modified PCs, Zou et al. (2006) proposed a new formula to compute the total variance explained by $\mathbf{P}$. Suppose $\{P_j, j = 1, 2, \ldots, k\}$ are the first k modified PC. Denote $\hat{P}_{j\cdot 1,\ldots,j-1}$ the residual after adjusting P_j for $P_1, \ldots, P_{j-1}$ by linear regression, that is

$$\hat{P}_{j\cdot 1,\ldots,j-1} = P_j - H_{1,\ldots,j-1}P_j \tag{4}$$

where $H_{1,\ldots,j-1}$ is the projection matrix on the linear spaced spanned by $\{P_i : i = 1, \ldots, j-1\}$. Then the adjusted variance of P_j is $\|P_{j\cdot 1,\ldots,j-1}\|^2$, and the total explained variance is defined as $\sum_{j=1}^{k}\|P_{j\cdot 1,\ldots,j-1}\|$.

Since $\mathbf{X}$ is not available in the measurement error setup, so the extension of the above formula to our situation is not straightforward. Notice that the total variance explained by the PCs only depends on the Gram matrix $\mathbf{P}^T\mathbf{P} = \mathbf{X}^T\mathbf{X}$ in the error-free case, and on the matrix $\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}}$ in the measurement error case, so we can define pseudo-PCs of $\mathbf{X}$ as $(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}})^{1/2}\hat{V}_j, j = 1, \ldots, k$. Then in formula (4), replace $P_j, j = 1, 2, \ldots, k$ by the pseudo-PCs $(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}})^{1/2}\hat{V}_j$, and the define the adjusted variance of $\mathbf{X}\hat{V}_j$ and the total explained variance accordingly.

Remark 2 (*Computation Complexity*)**.** Compare the algorithm proposed in this paper and the one from Zou et al. (2006), we can see that the major difference between these two procedure is that in the error-free case, the SPCA only depends on the Gram matrix $\mathbf{X}^T\mathbf{X}$, while in the measurement error case, the SPCA only depends on the bias-attenuated Gram matrix $\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}}$. Therefore, the computation complexity in measurement error setup is the same as in the error-free case.

Remark 3 (*Unknown* $\Sigma_{\mathbf{u}}$)**.** The assumption of known $\Sigma_{\mathbf{u}}$ may not be realistic in practice. However, if replicated measurements can be made at each $\mathbf{x}$ values, then this assumption can be removed. To be specific, assume at each $\mathbf{x}$ we can observe two $\mathbf{z}$ values, that is

$$\mathbf{z}_{ik} = \mathbf{x}_i + \mathbf{u}_{ik}, \quad k = 1, 2;\ i = 1, 2, \ldots, n.$$

Then we have

$$\bar{\mathbf{z}}_i = \frac{\mathbf{z}_{i2} + \mathbf{z}_{i1}}{2} = \mathbf{x}_i + \frac{\mathbf{u}_{i2} + \mathbf{u}_{i1}}{2}.$$

On the other hand, we also have $\tilde{\mathbf{z}}_i = (\mathbf{z}_{i2} - \mathbf{z}_{i1})/2 = (\mathbf{u}_{i2} - \mathbf{u}_{i1})/2$. Note that the covariance matrix of $(\mathbf{u}_{i2} + \mathbf{u}_{i1})/2$ is the same as that of $(\mathbf{u}_{i2} - \mathbf{u}_{i1})/2$, so the covariance matrix of $(\mathbf{u}_{i2} + \mathbf{u}_{i1})/2$ can be estimated by the sample covariance matrix based on $\tilde{\mathbf{z}}_i$'s. Denote this sample covariance matrix as $S_{\tilde{\mathbf{z}}\tilde{\mathbf{z}}}$, and the sample covariance matrix based on $\bar{\mathbf{z}}_i$'s as $S_{\bar{\mathbf{z}}\bar{\mathbf{z}}}$, then the sparse principal component analysis can be made through $S_{\bar{\mathbf{z}}\bar{\mathbf{z}}} - S_{\tilde{\mathbf{z}}\tilde{\mathbf{z}}}$, provided that it is nonnegative definite.

Remark 4 (*Relative Magnitude Between* $\Sigma_{\mathbf{x}}$ *and* $\Sigma_{\mathbf{u}}$)**.** The relative magnitude of $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{u}}$ does affect the performance of the proposed method, especially when the sample size is small. In this case, it is more likely for $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$ to be negative definite. However, as long as the sample size is large and $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$ keeps non-negative definite, the performance of the proposed method works well. The key is that $\Sigma_{\mathbf{u}}$ is known, which implies that no matter how large $\Sigma_{\mathbf{u}}$ is, we can always recover $\Sigma_{\mathbf{x}}$ from $S_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$, although the variability of the solution might be a question of concern.

4. Numerical studies

To evaluate the finite sample performance of the proposed methodology, we conduct two simulation studies in this section based on some existing frameworks in literature. The effects of sample size, the structures of covariance matrix of the measurement errors, the magnitudes of the measurement errors, and the choices of different penalty values on the resulting loading vectors will also be investigated. The first simulation is based on the hidden factor model in Zou et al. (2006). The known hidden factors allow us to better check the performance of the proposed methodology. By adding normal measurement errors with different covariance matrix structures to the 10 variables X_j's, which will be served as latent variables, 10 observed variables Z_j's are created. The second simulation is based on the correlation matrix of 13 variables from the pitprops data set in Jeffers (1967). The variances for these 13 variables are made up hence a pseudo covariance matrix is obtained, say Σ_X, then by assuming different covariance structures of measurement errors, we generate two sets of random observations X and U from normal distributions with specified covariance matrices, the observed contaminated data thus can be constructed according to the additive relationship $Z = X + U$. After replacing $\mathbf{S}_{\mathbf{xx}}$ with $\mathbf{S}_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$, the SPCA algorithm for measurement error model is same as the SPCA algorithm for error-free case, so in the following simulation, all the computations are done using R-package `elasticnet` developed by Zou and Hastie (2012). In the following, we shall call the SPCA based on $\mathbf{S}_{\mathbf{zz}}$ the Naive method, the SPCA based on $\mathbf{S}_{\mathbf{zz}} - \Sigma_{\mathbf{u}}$ the bias-corrected (BC) method, and the SPCA based on $\mathbf{S}_{\mathbf{xx}}$ the Oracle method.

The performance of the proposed BC SPCA will be compared with the Naive SPCA, and the SPCA based on the $\mathbf{S}_{\mathbf{xx}}$ of the latent variables which, although infeasible in real application, may serve as a benchmark for the purpose of comparison. The following three aspects will be used as the criteria to judge the goodness of procedure:

- The agreement of the sparsity. This agreement will be evaluated by checking the number of nonzero loading values for leading sparse PCs being selected.
- The direction of the loading vectors. This is evaluated by the correlation coefficients or the L_2-distances between the loading vectors from Naive SPCA or BC SPCA procedures and the SPCA procedure based on $\mathbf{S}_{\mathbf{xx}}$.
- The percentages of expressed variances from leading sparse PCs being selected.

4.1. A synthetic example

According to Zou et al. (2006), the three hidden factors are defined as

$$V_{1i} \sim N(0, 290), \qquad V_{2i} \sim N(0, 300), \qquad V_{3i} = -0.3V_{1i} + 0.925V_{2i} + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $\varepsilon_i \sim N(0, 1), i = 1, 2, \ldots, n$, and V_{1i}'s, V_{2i}'s and ε_i's are independent. Then 10 latent variables X are defined as follows

$$\begin{aligned} X_{ji} &= V_{1i} + \varepsilon_{1i}, \qquad \varepsilon_{1i} \sim N(0, 1), \quad j = 1, 2, 3, 4, \\ X_{ji} &= V_{2i} + \varepsilon_{2i}, \qquad \varepsilon_{2i} \sim N(0, 1), \quad j = 5, 6, 7, 8, \\ X_{ji} &= V_{3i} + \varepsilon_{3i}, \qquad \varepsilon_{3i} \sim N(0, 1), \quad j = 9, 10, \end{aligned}$$

Table 1
Covariance matrix of **X**.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_1	291									
X_2	290	291								
X_3	290	290	291							
X_4	290	290	290	291						
X_5	0	0	0	0	301					
X_6	0	0	0	0	300	301				
X_7	0	0	0	0	300	300	301			
X_8	0	0	0	0	300	300	300	301		
X_9	−87	−87	−87	−87	277.5	277.5	277.5	277.5	283.79	
X_{10}	−87	−87	−87	−87	277.5	277.5	277.5	277.5	282.79	283.79

Table 2
Number of non-zero loading values in the first 2 PCs.

	Naive		BC		BC-O		ORACLE	
	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2
$n = 100$	6	4	4	4	5	4	6	4
$n = 200$	6	4	4	3	6	4	6	4
$n = 500$	6	4	5	4	6	4	6	4
$n = 1000$	6	4	6	4	6	4	6	4

and ε_{ji}'s are independent. Let $\mathbf{U} = (U_1, \ldots, U_{10})'$ be a 10-dimensional multivariate random vectors from $N_{10}(0, \Sigma_{\mathbf{U}})$, we construct the observed variables according to $Z_{ji} = X_{ji} + U_{ji}$, where $(U_{j1}, \ldots, U_{j,n})$ are n observations from $U_j, j = 1, \ldots, 10$. In the simulation, $\Sigma_{\mathbf{u}}$ is chosen to be $\Sigma_{\mathbf{u},1} = 20I_{10\times 10}$, $\Sigma_{\mathbf{u},2}$ a diagonal matrix with randomly generated values from uniform distribution between 5 and 50 in its diagonal direction, and $\Sigma_{\mathbf{u},3}$ whose off-diagonal elements are 2's, the first five diagonal values are 30's and the rest five diagonal values are 15's. To see the effect of sample sizes, we choose $n = 100, 200, 500, 1000$.

For the sake of completeness, the lower triangular part of the population covariance matrix of **X** is given in Table 1.

As analyzed by Zou et al. (2006), in the measurement error free case, that is, the observations on **X** are available, then we only need to consider two derived variables with right sparse representations. Furthermore, the first PC should recover the factor V_2 only using X_5, X_6, X_7, X_8, and the second derived variables should recover the factor V_1 only using X_1, X_2, X_3 and X_4. The numerical study in Zou et al. (2006) also shows that by restricting the numbers of nonzero loadings to four, the first PC uniformly assigns nonzero loadings on X_5, X_6, X_7, X_8, while the second PC uniformly assigns nonzero on X_1, X_2, X_3 and X_4. However, in real application, the exact population covariance matrix of **X** is unknown, so we shall use the sample covariance in the simulation. Also, we will not restrict the number of nonzero loadings to four for the PCs, instead, the sparsity of the loading vectors is only controlled by the penalty parameter. In all the simulations, the parameter λ in quadratic penalty is chosen to be 0, and the parameters $\lambda_j, j = 1, \ldots, 10$ are chosen to be 20. Some other choices for the penalty parameters are also tested, and the simulation results show the performance of the procedures are pretty stable when the penalty parameters do not vary too much. For the sake of brevity, the simulation results of other choices are omitted.

As we mentioned in Section 1, also witnessed in Section 4 and Appendix, the proposed algorithms approximate $\mathbf{S_{xx}}$ with $\mathbf{S_{zz}} - \Sigma_{\mathbf{u}}$. In fact, a better approximation should be $\mathbf{S_{zz}} - \mathbf{S_{xu}} - \mathbf{S_{ux}} - \Sigma_{\mathbf{u}}$. Lack of information on $\mathbf{S_{xu}}$ and $\mathbf{S_{ux}}$ prevents us from making a more precise analysis based on the better approximation. To see the effect of the knowledge of $\mathbf{S_{xu}}$ and $\mathbf{S_{ux}}$ on the final SPCA, we also conduct a simulation study pretending we know the oracle information of these two covariance matrices, that is, we will use the simulated observations from **x** and **u** to calculate $\mathbf{S_{xu}}$ and $\mathbf{S_{ux}}$. The corresponding methodology is labeled as BC-O.

Tables 2–4 are the simulation results when $\Sigma_{\mathbf{u}} = 20I_{10\times 10}$. Table 2 reports the number of non-zero loading values in the 2 two leading PCs. It is not surprising to see that for such a measurement error covariance structure, all procedures picked out nearly same number of variables, in particular, when the sample size is large. The percentages of expressed variance from all ten PCs are presented in Table 3. For moderated large sample sizes, the proposed bias-corrected SPCA clearly is much superior to the naive algorithm, and as expected, the performance of BC-O is nearly the same as the SPCA based on the true covariance matrix of **X**. The superiority of the proposed BC over the naive method is more clearly seen from Table 4, where the correlations of the first selected PCs between Naive and Oracle, BC and Oracle, BC-O and Oracle are reported. The biggest correlations are between BC-O and Oracle, followed by BC and Oracle, and Naive and Oracle. The high correlation between BC and Oracle indicates that after bias correction, the biases in loading vectors are reduced, in particular for large sample sizes. Also, we can see, knowing the "oracle" information on $\mathbf{S_{xu}}$ and $\mathbf{S_{ux}}$ can greatly improve the SPCA performance.

Tables 5–7 are the simulation results for the second covariance structure, and Tables 8–10 are for the third covariance structure. From Tables 5 and 8, we can see the number of nonzero loading values are more variable for the Naive method when the sample size is small, and the superiority of the proposed method over the naive algorithm is evident. Another striking phenomenon is that under the selected penalty parameters, all 10 PCs contribute, no matter how small it is, to the

Table 3
Percentage of expressed variance.

		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
	Naive	0.150	0.157	0.015	0.013	0.010	0.009	0.012	0.008	0.007	0.007
$n = 100$	BC	0.301	0.365	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.339	0.440	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.521	0.447	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.238	0.113	0.011	0.014	0.010	0.011	0.008	0.008	0.008	0.008
$n = 200$	BC	0.295	0.167	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.562	0.393	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.580	0.399	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.275	0.161	0.010	0.008	0.007	0.008	0.006	0.006	0.008	0.006
$n = 500$	BC	0.461	0.387	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.515	0.382	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.573	0.403	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.226	0.139	0.010	0.009	0.009	0.011	0.008	0.008	0.008	0.007
$n = 1000$	BC	0.599	0.324	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.602	0.358	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.604	0.377	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 4
Correlation matrix.

	PC1			PC2		
	(Naive, Oracle)	(BC, Oracle)	(BC-O, Oracle)	(Naive, Oracle)	(BC, Oracle)	(BC-O, Oracle)
$n = 100$	0.273	0.518	0.746	0.378	0.749	0.950
$n = 200$	0.484	0.511	0.974	0.509	0.581	0.987
$n = 500$	0.532	0.770	0.899	0.496	0.960	0.957
$n = 1000$	0.470	0.993	0.998	0.523	0.892	0.959

Table 5
Number of non-zero loading values in the first 2 PCs.

	Naive		BC		BC-O		ORACLE	
	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2
$n = 100$	1	2	6	4	6	3	6	4
$n = 200$	6	1	5	4	6	4	6	4
$n = 500$	6	4	5	3	5	4	6	4
$n = 1000$	1	4	5	4	6	4	6	4

Table 6
Percentage of expressed variance.

		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
	Naive	0.088	0.130	0.029	0.019	0.024	0.018	0.018	0.014	0.009	0.006
$n = 100$	BC	0.327	0.422	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.513	0.316	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.521	0.447	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.187	0.113	0.032	0.021	0.018	0.016	0.012	0.009	0.007	0.005
$n = 200$	BC	0.309	0.391	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.558	0.403	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.569	0.408	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.149	0.162	0.022	0.014	0.015	0.010	0.011	0.007	0.004	0.004
$n = 500$	BC	0.369	0.298	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.404	0.388	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.578	0.398	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.108	0.142	0.027	0.021	0.017	0.014	0.010	0.009	0.007	0.006
$n = 1000$	BC	0.449	0.356	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.604	0.361	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.609	0.372	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

total variances, and the bias-corrected methods successfully put all their variance contributions into the first two PCs, which coincides perfectly with the SPCA conducted directly on the covariance matrix of the latent variables.

We also tried other covariance matrices for the measurement error with different magnitude, all simulation results show that the proposed method can greatly reduce the bias, and can even almost eliminate the bias when sample size is large. Those results will not be presented here for the sake of brevity.

Table 7
Correlation matrix.

	PC1			PC2		
	(Naive, Oracle)	(BC, Oracle)	(BC-O, Oracle)	(Naive, Oracle)	(BC, Oracle)	(BC-O, Oracle)
$n = 100$	0.322	0.669	0.969	0.286	0.940	0.838
$n = 200$	0.453	0.587	0.988	0.448	0.976	0.994
$n = 500$	0.382	0.640	0.694	0.518	0.760	0.966
$n = 1000$	0.245	0.756	0.995	0.542	0.967	0.972

Table 8
Number of non-zero loading values in the first 2 PCs.

	Naive		BC		BC-O		ORACLE	
	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2
$n = 100$	6	4	4	3	5	4	6	4
$n = 200$	6	1	4	3	6	4	6	4
$n = 500$	6	4	5	4	6	4	6	4
$n = 1000$	6	4	6	4	6	4	6	4

Table 9
Percentage of expressed variance.

		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
	Naive	0.321	0.130	0.022	0.015	0.010	0.010	0.007	0.005	0.005	0.000
$n = 100$	BC	0.294	0.325	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.344	0.438	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.521	0.447	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.387	0.102	0.022	0.016	0.012	0.009	0.004	0.005	0.005	0.006
$n = 200$	BC	0.290	0.175	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.568	0.393	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.580	0.399	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.487	0.122	0.015	0.012	0.011	0.008	0.004	0.004	0.003	0.000
$n = 500$	BC	0.458	0.382	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.563	0.383	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.573	0.403	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.433	0.105	0.018	0.013	0.009	0.011	0.004	0.004	0.004	0.005
$n = 1000$	BC	0.600	0.282	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.604	0.350	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.604	0.377	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 10
Correlation matrix.

	PC1			PC2		
	(Naive, Oracle)	(BC, Oracle)	(BC-O, Oracle)	(Naive, Oracle)	(BC, Oracle)	(BC-O, Oracle)
$n = 100$	0.630	0.518	0.754	0.303	0.671	0.944
$n = 200$	0.714	0.501	0.986	0.318	0.595	0.989
$n = 500$	0.861	0.765	0.987	0.408	0.957	0.960
$n = 1000$	0.738	0.995	0.999	0.440	0.810	0.944

To check the performance of the proposed method when p is larger, we also conduct a simulation study when $p = 50$ and $n = 100, 200, 500$ and 1000. In the simulation, the first ten latent variables are the same as in the above simulation study, the other 40 latent variables are generated from $X_{ji} = 0.1V_{1i} + \varepsilon_{ji}, j = 11, 12, \ldots, 50$, and $i = 1, 2, \ldots, n$, $\varepsilon_{ji} \sim N(0, 1)$ for all i, j. The $\Sigma_{\mathbf{u}}$ is also chosen as the ones we used in the above simulation, except now the dimension becomes 50. For $\Sigma_{\mathbf{u},3}$, we set the first 25 diagonal values to be 30 and the rest 25 diagonal values to be 15, the off-diagonal elements are still 2's. The simulation results clearly show that the proposed method is superior to the naive method. Tables 11–13 are the simulation results for $\Sigma_{\mathbf{u},3}$. The simulation results for $\Sigma_{\mathbf{u},1}$ and $\Sigma_{\mathbf{u},2}$ are omitted for the sake of brevity.

We also conduct a simulation study when $\Sigma_{\mathbf{u}}$ is unknown but replicated observations on $\mathbf{x}$ are available, see Remark 3 in Section 3 for the details of implementation. The simulation results show that the naive method sometimes performs better than the proposed bias-corrected method. Since the average of replicates $\mathbf{z}$ approximates the true $\mathbf{x}$ better than only single observation, so the improvement is not a surprise. However, as the measurement error variances become larger, the proposed method becomes superior to the naive method.

Table 11
Number of non-zero loading values in the first 2 PCs.

	Naive		BC		BC-O		ORACLE	
	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2
$n = 100$	3	2	5	4	5	4	6	4
$n = 200$	1	4	6	4	6	4	6	4
$n = 500$	1	4	6	4	6	4	6	4
$n = 1000$	6	4	6	4	6	4	6	4

Table 12
Percentage of expressed variance.

		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
	Naive	0.134	0.089	0.008	0.007	0.007	0.008	0.016	0.007	0.007	0.008
$n = 100$	BC	0.344	0.381	0.006	0.003	0.003	0.002	0.000	0.000	0.000	0.000
	BC-O	0.361	0.378	0.004	0.004	0.003	0.006	0.000	0.000	0.000	0.000
	Oracle	0.492	0.422	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.080	0.076	0.008	0.009	0.007	0.008	0.008	0.016	0.014	0.011
$n = 200$	BC	0.534	0.323	0.003	0.003	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.545	0.327	0.003	0.003	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.575	0.350	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.082	0.084	0.007	0.015	0.008	0.007	0.006	0.010	0.010	0.007
$n = 500$	BC	0.565	0.322	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.570	0.330	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.586	0.343	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Naive	0.117	0.091	0.007	0.011	0.010	0.007	0.007	0.006	0.007	0.007
$n = 1000$	BC	0.564	0.337	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	BC-O	0.577	0.337	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Oracle	0.587	0.348	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 13
Correlation matrix.

	PC1			PC2		
	(Naive, Oracle)	(BC, Oracle)	(BC-O, Oracle)	(Naive, Oracle)	(BC, Oracle)	(BC-O, Oracle)
$n = 100$	0.585	0.853	0.877	0.415	0.967	0.975
$n = 200$	0.394	0.988	0.996	0.500	0.991	1.000
$n = 500$	0.337	0.989	0.998	0.525	0.989	0.998
$n = 1000$	0.487	0.988	1.000	0.542	0.996	0.995

Table 14
Pitprops correlation matrix.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}
X_1	1.000												
X_2	0.954	1.000											
X_3	0.364	0.297	1.000										
X_4	0.342	0.284	0.882	1.000									
X_5	−0.129	−0.118	−0.148	0.220	1.000								
X_6	0.313	0.291	0.153	0.381	0.364	1.000							
X_7	0.496	0.503	−0.029	0.174	0.296	0.813	1.000						
X_8	0.424	0.419	−0.054	−0.059	0.004	0.090	0.372	1.000					
X_9	0.592	0.648	0.125	0.137	−0.039	0.211	0.465	0.482	1.000				
X_{10}	0.545	0.569	−0.081	−0.014	0.037	0.274	0.679	0.557	0.526	1.000			
X_{11}	0.084	0.076	0.162	0.097	−0.091	−0.036	−0.113	0.061	0.085	−0.319	1.000		
X_{12}	−0.019	−0.036	0.220	0.169	−0.145	0.024	−0.232	−0.357	−0.127	−0.368	0.029	1.000	
X_{13}	0.134	0.144	0.126	0.015	−0.208	−0.329	−0.424	−0.202	−0.076	−0.291	0.007	0.184	1.000

4.2. A sensitivity study based on pitprops data

The pitprops data set of Jeffers (1967) consists of a correlation matrix based on 13 variables and 180 observations. Those 13 variables were originally used as explanatory variables in a regression problem from a study on the strength of pitprops cut from home-grown timber. See Jeffers (1967) for a detailed description of the data. It is a famous classical example illustrating the difficulty of interpreting the PCs. The pitprops correlation matrix, denoted by **R**, is given in Table 14.

To generate the observed data, we define a diagonal matrix $\mathbf{V} = \text{diag}(\sigma_1, \ldots, \sigma_{13})$, and $\mathbf{VRV}$ will be treated as the covariance matrix of $\mathbf{X} = (X_1, \ldots, X_{13})$; then we define a covariance matrix $\Sigma_{\mathbf{u}}$ which will be treated as the covariance

Table 15
Number of non-zero loading values in the first 6 PCs.

	Naive				BC				ORACLE			
	PC1	PC2	PC3	PC4-6	PC1	PC2	PC3	PC4-6	PC1	PC2	PC3	PC4-6
$n = 180$	6	2	3	1	5	3	3	1	6	4	4	1
$n = 360$	5	1	2	1	9	3	2	1	9	4	4	1
$n = 720$	6	1	2	1	6	4	4	1	6	3	4	1
$n = 1440$	7	1	3	1	7	2	5	1	7	3	4	1

Table 16
Percentage of expressed variance.

		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
	Naive	0.092	0.092	0.072	0.078	0.074	0.065	0.065	0.067	0.057	0.047	0.045	0.045	0.043
$n = 180$	BC	0.255	0.173	0.079	0.074	0.068	0.044	0.034	0.035	0.000	0.000	0.000	0.000	0.000
	Oracle	0.206	0.137	0.103	0.065	0.057	0.066	0.045	0.038	0.040	0.018	0.000	0.000	0.000
	Naive	0.089	0.082	0.081	0.078	0.076	0.069	0.069	0.065	0.056	0.054	0.052	0.051	0.052
$n = 360$	BC	0.194	0.177	0.123	0.078	0.070	0.053	0.052	0.031	0.019	0.000	0.000	0.000	0.000
	Oracle	0.238	0.110	0.138	0.064	0.049	0.065	0.051	0.032	0.015	0.030	0.000	0.000	0.000
	Naive	0.085	0.084	0.079	0.079	0.071	0.071	0.071	0.066	0.061	0.060	0.051	0.051	0.053
$n = 720$	BC	0.189	0.139	0.121	0.080	0.069	0.050	0.058	0.036	0.030	0.015	0.000	0.000	0.000
	Oracle	0.185	0.127	0.125	0.077	0.071	0.058	0.052	0.046	0.039	0.020	0.000	0.000	0.000
	Naive	0.084	0.076	0.068	0.075	0.073	0.069	0.070	0.069	0.061	0.071	0.057	0.058	0.049
$n = 1440$	BC	0.160	0.133	0.126	0.073	0.067	0.048	0.049	0.045	0.028	0.035	0.000	0.000	0.000
	Oracle	0.189	0.131	0.115	0.076	0.075	0.053	0.056	0.038	0.044	0.019	0.000	0.000	0.000

Table 17
Distances of loading vectors between Naive, BC and Oracle.

	PC1		PC2		PC3		PC4		PC5		PC6	
	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O
$n = 180$	1.732	0.364	0.395	0.003	0.687	0.044	0	0	0	0	2	2
$n = 360$	0.674	0.114	0.108	0.343	0.694	0.104	2	2	2	2	0	0
$n = 720$	0.579	0.013	2.000	0.002	2.000	0.002	0	0	2	2	2	2
$n = 1440$	1.822	0.080	0.554	0.003	1.880	0.006	0	0	0	0	2	2

matrix of measurement errors; finally, a sample of size n from the multivariate normal distribution with mean 0, covariance matrix **VRV**, and a sample of size n from the multivariate normal distribution with mean 0, covariance matrix $\Sigma_{\mathbf{u}}$ will be generated, denote them as $\mathbf{X}$ and $\mathbf{U}$, and the observed data set will be $\mathbf{Z} = \mathbf{X} + \mathbf{U}$. With different choices of $\Sigma_{\mathbf{u}}$, we shall apply the proposed SPCA algorithm to find SPCAs for $\mathbf{X}$. Similar to the first simulation study, we will investigate the effect of sample sizes, structure of covariance matrix of the measurement error on the sparsity of the resulting PCAs. The sample sizes are taken to be $n = 180, 360, 720$ and 1440.

Three covariance structures for the measurement errors are considered in the simulation: $\Sigma_{\mathbf{u}} = 2I_{13\times 13}$; $\Sigma_{\mathbf{u}} = \text{diag}(2, 2, 2, 2, 4, 4, 4, 4, 1, 1, 1, 1, 6)$; and $\Sigma_{\mathbf{u}}$ with the first 7 values being 10, and the last 6 values being 1 in its diagonal, and 0.25 being off-diagonal entries.

As in Zou et al. (2006), the first 6 PCs are reported here. For the first covariance structure, Table 12 reports the nonzero loading values from the first 6 PCs for three procedures, the naive, BC and the Oracle procedures; Table 13 is the percentage of expressed variances. Different from the first simulation study, here the closeness of the PCs between any two different methods is evaluated by the L_2 distances of PC loading vectors, because of all zero values in some PCs. In this simulation, we choose $\lambda = 0$ for the quadratic penalty, and for L_1-penalty, $\lambda_1 = 0.06, 0.16, 0.1$ for the first three PCs and 0.5 for other 10 PCs, the first 6 λ_1 values are the same as in Zou et al. (2006), we also used some different λ_1 values in the simulation near the values we are chosen, and the simulation results vary only slightly. The SPCA result shows that under the error-free setup, that is the SPCA based on the covariance matrix given in Table 14, the number of nonzero loading values for the first 6 PCs are 7, 4, 4, 1, 1, 1, also see Zou et al. (2006). As seen from Table 15, the analysis based on $\mathbf{S_{xx}}$ gives almost identical result for large sample sizes. The slim difference clearly is due to the simulation randomness. Compared to the Naive method, the BC method produces closer results to the Oracle. The percentages of expressed variances reported in Table 16 show the proposed BC method is more similar to the Oracle than the Naive method to the Oracle, and the distances of loading vectors from Naive and BC to the Oracle reported in Table 17 clearly indicate that the loading vectors from BC are pretty well aligned with the loading vectors from Oracle except for very few cases. The distance value 2 for PC4 through PC6 is the results from the SPCA procedures assigning value 1 to exactly one variable. For brevity, in Table 17, BC/O means the distance of loading vectors between BC and Oracle, and Naive/O between Naive and Oracle.

Tables 18–20 are the simulation results for the second covariance structure, and Tables 21–23 are for the third covariance structure. Similar patterns to the above simulation for the first type covariance matrix can be seen. We can also see that under

Table 18
Number of non-zero loading values in the first 6 PCs.

	Naive				BC				ORACLE			
	PC1	PC2	PC3	PC4-6	PC1	PC2	PC3	PC4-6	PC1	PC2	PC3	PC4-6
$n = 180$	7	3	2	1	8	3	7	1	6	4	4	1
$n = 360$	5	1	1	1	8	3	2	1	9	4	4	1
$n = 720$	6	1	1	1	9	4	3	1	6	3	4	1
$n = 1440$	5	1	1	1	6	2	6	1	7	3	4	1

Table 19
Percentage of expressed variance.

		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
	Naive	0.107	0.080	0.138	0.092	0.092	0.056	0.068	0.049	0.042	0.041	0.041	0.037	0.028
$n = 180$	BC	0.190	0.178	0.071	0.072	0.069	0.045	0.026	0.018	0.015	0.000	0.000	0.000	0.000
	Oracle	0.206	0.137	0.103	0.065	0.057	0.066	0.045	0.038	0.040	0.018	0.000	0.000	0.000
	Naive	0.078	0.140	0.096	0.093	0.070	0.089	0.076	0.047	0.047	0.046	0.047	0.038	0.030
$n = 360$	BC	0.202	0.190	0.123	0.070	0.069	0.052	0.052	0.030	0.019	0.000	0.000	0.000	0.000
	Oracle	0.238	0.110	0.138	0.064	0.049	0.065	0.051	0.032	0.015	0.030	0.000	0.000	0.000
	Naive	0.069	0.142	0.100	0.065	0.096	0.088	0.083	0.051	0.045	0.049	0.037	0.046	0.032
$n = 720$	BC	0.195	0.086	0.103	0.077	0.079	0.059	0.047	0.037	0.022	0.016	0.000	0.000	0.000
	Oracle	0.185	0.127	0.125	0.077	0.071	0.058	0.052	0.046	0.039	0.020	0.000	0.000	0.000
	Naive	0.066	0.135	0.099	0.094	0.065	0.093	0.085	0.048	0.045	0.043	0.050	0.045	0.034
$n = 1440$	BC	0.157	0.133	0.139	0.073	0.068	0.053	0.050	0.043	0.025	0.000	0.000	0.000	0.000
	Oracle	0.189	0.131	0.115	0.076	0.075	0.053	0.056	0.038	0.044	0.019	0.000	0.000	0.000

Table 20
Distances of loading vectors between Naive, BC and Oracle.

	PC1		PC2		PC3		PC4		PC5		PC6	
	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O
$n = 180$	1.968	1.365	0.407	0.003	1.959	1.907	2	2	2	2	2	2
$n = 360$	0.674	0.071	0.108	0.291	2.000	0.113	2	2	2	2	2	0
$n = 720$	0.535	0.144	2.000	0.680	0.480	0.936	2	0	2	2	2	2
$n = 1440$	0.573	0.641	2.000	0.001	2.000	0.713	2	0	2	0	2	2

Table 21
Number of non-zero loading values in the first 6 PCs.

	Naive				BC				ORACLE			
	PC1	PC2	PC3	PC4-6	PC1	PC2	PC3	PC4-6	PC1	PC2	PC3	PC4-6
$n = 180$	7	2	1	1	6	2	6	1	6	4	4	1
$n = 360$	5	1	2	1	8	3	4	1	9	4	4	1
$n = 720$	6	1	2	1	8	3	8	1	6	3	4	1
$n = 1440$	7	2	3	1	7	2	6	1	7	3	4	1

Table 22
Percentage of expressed variance.

		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
	Naive	0.133	0.127	0.109	0.118	0.109	0.085	0.099	0.028	0.027	0.027	0.022	0.023	0.015
$n = 180$	BC	0.195	0.217	0.109	0.059	0.067	0.050	0.041	0.019	0.015	0.000	0.000	0.000	0.000
	Oracle	0.206	0.137	0.103	0.065	0.057	0.066	0.045	0.038	0.040	0.018	0.000	0.000	0.000
	Naive	0.129	0.134	0.108	0.115	0.111	0.098	0.107	0.026	0.029	0.027	0.027	0.022	0.017
$n = 360$	BC	0.139	0.157	0.130	0.066	0.065	0.044	0.050	0.028	0.021	0.021	0.000	0.000	0.000
	Oracle	0.238	0.110	0.138	0.064	0.049	0.065	0.051	0.032	0.015	0.030	0.000	0.000	0.000
	Naive	0.128	0.118	0.105	0.122	0.109	0.109	0.109	0.024	0.030	0.029	0.025	0.020	0.018
$n = 720$	BC	0.233	0.108	0.072	0.069	0.058	0.061	0.052	0.037	0.030	0.014	0.000	0.000	0.000
	Oracle	0.185	0.127	0.125	0.077	0.071	0.058	0.052	0.046	0.039	0.020	0.000	0.000	0.000
	Naive	0.138	0.109	0.115	0.111	0.113	0.112	0.104	0.029	0.029	0.027	0.026	0.023	0.019
$n = 1440$	BC	0.182	0.141	0.134	0.074	0.068	0.053	0.046	0.044	0.026	0.000	0.000	0.000	0.000
	Oracle	0.189	0.131	0.115	0.076	0.075	0.053	0.056	0.038	0.044	0.019	0.000	0.000	0.000

the selected penalty parameters, all 13 PCs contribute to the total variances for Naive method, while the BC algorithm can produce some PCs with 0 contribution of variances.

Table 23
Distances of loading vectors between Naive, BC and Oracle.

	PC1		PC2		PC3		PC4		PC5		PC6	
	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O	Naive/O	BC/O
$n = 180$	1.998	0.045	0.978	0.003	1.949	1.020	2	0	2	0	2	2
$n = 360$	0.729	0.609	2.000	0.217	2.000	0.678	2	2	2	2	2	0
$n = 720$	1.998	0.380	2.000	2.000	0.978	1.745	2	0	2	0	2	0
$n = 1440$	1.561	0.255	0.969	0.016	1.646	0.343	2	0	2	0	2	0

Acknowledgments

The authors would like to thank the associate editor and two referees for their constructive comments and suggestions, which lead to great improvement on the presentation of the paper.

Jianhong Shi's research is supported by the Natural Science Foundation of Shanxi Province, China (2013011002-1), Weixing Song's research is partly supported by the grant NSF DMS 1205276.

Appendix. Proof of main results

Proof of direct approximation. The target function can be rewritten as

$$(V_j - \boldsymbol{\beta})^T(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}})(V_j - \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|^2.$$

The minimizer $\hat{\boldsymbol{\beta}}$ therefore has the form of

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}} + \lambda I)^{-1}(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}})V_j.$$

Plugging the spectral decomposition of $\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}}$ into the above expression, we obtain

$$\hat{\boldsymbol{\beta}} = \mathbf{V}(\mathrm{diag}(d_k^2) + \lambda I)^{-1}\mathrm{diag}(d_k^2)\mathbf{V}^T V_j = \frac{d_j^2}{d_j^2 + \lambda}V_j$$

which implies the desired result. □

Proof of Theorem 1. Note that the target function can be rewritten as

$$\sum_{i=1}^{n}\|\mathbf{z}_i - \boldsymbol{\alpha}\boldsymbol{\beta}^T\mathbf{z}_i\|^2 = \|\mathbf{Z} - \mathbf{Z}\boldsymbol{\beta}\boldsymbol{\alpha}^T\|^2 = \mathrm{Tr}(\mathbf{Z}^T\mathbf{Z}) + \boldsymbol{\beta}^T\mathbf{Z}^T\mathbf{Z}\boldsymbol{\beta} - 2\boldsymbol{\alpha}^T\mathbf{Z}^T\mathbf{Z}\boldsymbol{\beta}.$$

Therefore, the optimization problem is equivalent to minimizing

$$\boldsymbol{\beta}^T\mathbf{Z}^T\mathbf{Z}\boldsymbol{\beta} - 2\boldsymbol{\alpha}^T\mathbf{Z}^T\mathbf{Z}\boldsymbol{\beta} - n\boldsymbol{\beta}^T\Sigma_{\mathbf{u}}\boldsymbol{\beta} + 2n\boldsymbol{\beta}^T\Sigma_{\mathbf{u}}\boldsymbol{\alpha} + \lambda\|\boldsymbol{\beta}\|^2 \tag{5}$$

with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ subject to $\boldsymbol{\alpha}^T\boldsymbol{\alpha} = 1$. For any fixed α, taking derivative with respect to $\boldsymbol{\beta}$ of the above function, and setting the derivative to be 0, one can get

$$\left(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}} + \lambda I\right)\boldsymbol{\beta} - \left(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}}\right)\boldsymbol{\alpha} = 0$$

or $\boldsymbol{\beta}(\boldsymbol{\alpha}) = \left(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}} + \lambda I\right)^{-1}\left(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}}\right)\boldsymbol{\alpha}$. Plugging $\boldsymbol{\beta}(\boldsymbol{\alpha})$ into (6), then we have to maximize the following with respect to $\boldsymbol{\alpha}$,

$$\boldsymbol{\alpha}^T\left(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}}\right)\left(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}} + \lambda I\right)^{-1}\left(\mathbf{Z}^T\mathbf{Z} - n\Sigma_{\mathbf{u}}\right)\boldsymbol{\alpha}.$$

By the spectral decomposition $\mathbf{Z}^T\mathbf{Z} - \Sigma_{\mathbf{u}} = \mathbf{V}\mathbf{D}^2\mathbf{V}^T$, we can easily obtain that $\hat{\boldsymbol{\alpha}} = V_1$. Hence

$$\hat{\boldsymbol{\beta}} = \frac{d_1^2}{d_1^2 + \lambda}V_1$$

which implies the desired result. □

Proof of Theorem 2. Note that $\mathbf{A}$ is a $p \times k$ matrix such that $\mathbf{A}^T\mathbf{A} = I$, so we can find another $p \times (p-k)$ matrix $\mathbf{A}_\perp$ such that $[\mathbf{A}; \mathbf{A}_\perp]$ is a $p \times p$ orthonormal matrix. Then we can write $\|\mathbf{Z} - \mathbf{Z}\mathbf{B}\mathbf{A}^T\|^2 = \|\mathbf{Z}\mathbf{A}_\perp\|^2 + \|\mathbf{Z}\mathbf{A} - \mathbf{Z}\mathbf{B}\|^2$. Note that $\mathbf{A}\mathbf{A}^T + \mathbf{A}_\perp\mathbf{A}_\perp^T = I$, the target function in Theorem 2 thus can be rewritten as

$$\|\mathbf{Z}\mathbf{A}_\perp\|^2 + \|\mathbf{Z}\mathbf{A} - \mathbf{Z}\mathbf{B}\|^2 - n\cdot\mathrm{Tr}(\mathbf{A}_\perp^T\Sigma_{\mathbf{u}}\mathbf{A}_\perp) - n\cdot\mathrm{Tr}\left[(\mathbf{A} - \mathbf{B})^T\Sigma_{\mathbf{u}}(\mathbf{A} - \mathbf{B})\right] + \lambda\|\mathbf{B}\|^2. \tag{6}$$

Because

$$\|\mathbf{ZA}-\mathbf{ZB}\|^2=\sum_{j=1}^{k}\|\mathbf{Z}\boldsymbol{\alpha}_j-\mathbf{Z}\boldsymbol{\beta}_j\|^2,\qquad \mathrm{Tr}\left[(\mathbf{A}-\mathbf{B})^T\Sigma_{\mathbf{u}}(\mathbf{A}-\mathbf{B})\right]=\sum_{j=1}^{k}\|\Sigma_{\mathbf{u}}^{1/2}\boldsymbol{\alpha}_j-\Sigma_{\mathbf{u}}^{1/2}\boldsymbol{\beta}_j\|,$$

so for any fixed $\mathbf{A}$, for each $j=1,\ldots,k$, we should minimize the following quantity with respect to $\boldsymbol{\beta}_j$,

$$\|\mathbf{Z}\boldsymbol{\alpha}_j-\mathbf{Z}\boldsymbol{\beta}_j\|^2-n\|\Sigma_{\mathbf{u}}^{1/2}\boldsymbol{\alpha}_j-\Sigma_{\mathbf{u}}^{1/2}\boldsymbol{\beta}_j\|+\lambda\|\boldsymbol{\beta}_j\|^2=(\boldsymbol{\alpha}_j-\boldsymbol{\beta}_j)^T(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})(\boldsymbol{\alpha}_j-\boldsymbol{\beta}_j)+\lambda\|\boldsymbol{\beta}_j\|^2.$$

Easy to see the minimizer of the above function is $\boldsymbol{\beta}_j(\boldsymbol{\alpha}_j)=(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}}+\lambda I)^{-1}(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})\boldsymbol{\alpha}_j$, this implies

$$\mathbf{B}(\mathbf{A})=(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}}+\lambda I)^{-1}(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})\mathbf{A}. \tag{7}$$

Plugging $\mathbf{B}(\mathbf{A})$ into (6), and note that $(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})(\mathbf{A}\mathbf{A}^T+\mathbf{A}_\perp\mathbf{A}_\perp^T)=(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})$, we see that the target function (6) can be written as

$$\mathrm{Tr}(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})-\mathrm{Tr}\left[\mathbf{A}^T((\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}}))(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}}+\lambda I)^{-1}(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})\mathbf{A}\right].$$

Recall the spectral decomposition $\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}}=\mathbf{V}\mathbf{D}^2\mathbf{V}^T$, the target function can be further written as

$$\mathrm{Tr}(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})-\mathrm{Tr}\left[\mathbf{A}^T\mathbf{V}\mathbf{D}^2(\mathbf{D}^2+\lambda I)^{-1}\mathbf{D}^2\mathbf{V}^T\mathbf{A}\right].$$

To minimize the target function now amounts to maximize $\mathrm{Tr}\left[\mathbf{A}^T\mathbf{V}\mathbf{D}^2(\mathbf{D}^2+\lambda I)^{-1}\mathbf{D}^2\mathbf{V}^T\mathbf{A}\right]$ with respect to $\mathbf{A}$. It is easy to see that the maximizer is $\hat{\mathbf{A}}=\mathbf{V}[;1:k]$, the first k columns of $\mathbf{V}$. Thus, from (7),

$$\hat{\mathbf{B}}=(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}}+\lambda I)^{-1}(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})\mathbf{V}[;1:k]=\mathbf{V}(\mathbf{D}^2+\lambda I)^{-1}\mathbf{D}^2\mathbf{V}^T\mathbf{V}[;1:k].$$

That is, for any $j=1,\ldots,k$, $\hat{\boldsymbol{\beta}}_j=[d_j^2/(d_j^2+\lambda)]V_j\propto V_j$. This completes the proof. □

Proof of Theorem 3. Note that $\|\mathbf{Z}-\mathbf{ZBA}^T\|^2=\mathrm{Tr}\left[(I-\mathbf{AB}^T)\mathbf{Z}^T\mathbf{Z}(I-\mathbf{BA}^T)\right]$ and $n\cdot\mathrm{Tr}\left[(\mathbf{AB}^T+\mathbf{BA}^T)\Sigma_{\mathbf{u}}\right]=2n\mathrm{Tr}(\Sigma_{\mathbf{u}}\mathbf{BA}^T)$, so the target function in Theorem 3 can be written as

$$\mathrm{Tr}(\mathbf{Z}^T\mathbf{Z})-2\mathrm{Tr}(\mathbf{Z}^T\mathbf{ZBA}^T)+\mathrm{Tr}(\mathbf{AB}^T\mathbf{Z}^T\mathbf{ZBA}^T)+2n\cdot\mathrm{Tr}(\Sigma_{\mathbf{u}}\mathbf{BA}^T).$$

Using the constraint $\mathbf{A}^T\mathbf{A}=I$, the above target function can be further simplified to

$$\mathrm{Tr}(\mathbf{Z}^T\mathbf{Z})+\mathrm{Tr}(\mathbf{B}^T\mathbf{Z}^T\mathbf{ZB})-2\mathrm{Tr}([\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}}]\mathbf{BA}^T).$$

Therefore, we only have to maximize the term $\mathrm{Tr}([\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}}]\mathbf{BA}^T)$, which has the form of $\mathrm{Tr}(\mathbf{M}^T\mathbf{NA}^T)$ as in the Theorem 4 of Zou et al. (2006). With the SVD of $(\mathbf{Z}^T\mathbf{Z}-n\Sigma_{\mathbf{u}})\mathbf{B}$, the theorem follows the same proof thread as in Zou et al. (2006). □

References

Alter, O., Brown, P., Botstein, D., 2000. Singular value decomposition for genome-wide expression data processing and modeling. Proc. Natl. Acad. Sci. 97, 10101–10106.

Carroll, R.J., Ruppert, D., Stefanski, L.A., Crainiceanu, C.M., 2006. Measurement Error in Nonlinear Models: A Modern Perspective, second ed. Chapman and Hall/CRC.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Ann. Statist. 32, 407–499.

Fuller, W.A., 1987. Measurement Error Models. John Wiley and Sons, Inc..

Gorsuch, R.L., 1983. Factor Analysis, second ed.. Lawrence Erlbaum Associates, Hillsdale, NJ.

Hancock, P., Burton, A., Bruce, V., 1996. Face processing: Human perception and principal components analysis. Mem. Cogn. 24, 26–40.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning: Data mining, Inference and Prediction. Springer Verlag, New York.

Henry, F.K., 1958. The varimax criterion for analytic rotation in factor analysis. Psychometrika 23 (3), 187–200.

Jeffers, J., 1967. Two case studies in the application of principal component. Appl. Stat. 16, 225–236.

Jolliffe, I.T., Uddin, M., 2003. A modified principal component technique based on the lasso. J. Comput. Graph. Statist. 12, 531–547.

Tabachnick, B.G., Fidell, L.S., 2007. Using Multivariate Statistics, fifth ed.. Pearson Allyn & Bacon, Upper Saddle River, NJ.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B 58, 267–288.

Vines, S., 2000. Simple principal components. Appl. Stat. 49, 441–451.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B 67, 301–320.

Zou, H., Hastie, T., 2012. elasticnet: Elastic-net for sparse estimation and sparse PCA. R package version 1.1. http://CRAN.R-project.org/package=elasticnet.

Zou, H., Hastie, T., Tibshirani, R., 2006. Sparse principal component analysis. J. Comput. Graph. Statist. 15 (2), 265–286.

Zou, H., Trevor, T., 2005. Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B 67 (2), 301–320.