Advances in Data Analysis and Classification (2021) 15:795–823
https://doi.org/10.1007/s11634-020-00435-2

REGULAR ARTICLE

Sparse principal component regression via singular value decomposition approach

Shuichi Kawano[1]

Received: 17 March 2020 / Revised: 30 November 2020 / Accepted: 11 December 2020 /
Published online: 8 February 2021

Abstract
Principal component regression (PCR) is a two-stage procedure: the first stage performs principal component analysis (PCA) and the second stage builds a regression model whose explanatory variables are the principal components obtained in the first stage. Since PCA is performed using only explanatory variables, the principal components have no information about the response variable. To address this problem, we present a one-stage procedure for PCR based on a singular value decomposition approach. Our approach is based upon two loss functions, which are a regression loss and a PCA loss from the singular value decomposition, with sparse regularization. The proposed method enables us to obtain principal component loadings that include information about both explanatory variables and a response variable. An estimation algorithm is developed by using the alternating direction method of multipliers. We conduct numerical studies to show the effectiveness of the proposed method.

Keywords ADMM · Lasso · One-stage procedure · Singular value decomposition · Principal component analysis

Mathematics Subject Classification 62H25 · 62J07 · 62J05

1 Introduction

Principal component regression (PCR), invented by Jolliffe (1982) and Massy (1965), is widely used in various fields of research, including chemometrics, bioinformatics, and psychology, and has been extensively studied (Chang and Yang 2012; Dicker

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11634-020-00435-2.

✉ Shuichi Kawano
skawano@ai.lab.uec.ac.jp

[1] Department of Computer and Network Engineering, Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan

et al. 2017; Febrero-Bande et al. 2017; Frank and Friedman 1993; Hartnett et al. 1998; Reiss and Ogden 2007; Rosipal et al. 2001; Wang and Abbott 2008). PCR is a two-stage procedure: one first performs principal component analysis (PCA) (Jolliffe 2002; Pearson 1901), and then performs regression in which the explanatory variables are the selected principal components. However, the principal components have no information on the response variable. Because of this, the prediction accuracy of the PCR could be low, if the response variable is related to principal components having small eigenvalues.

To address this problem, a one-stage procedure for PCR was proposed in Kawano et al. (2015). This one-stage procedure was developed by combining a regression squared loss function with the sparse PCA (SPCA) loss function in Zou et al. (2006). The estimate of the regression parameter and loading matrix in the PCA is obtained as the minimizer of the combination of two loss functions with sparse regularization. By virtue of sparse regularization, sparse estimates of the parameters can be obtained. Kawano et al. (2015) referred to the one-stage procedure as sparse principal component regression (SPCR). Kawano et al. (2018) also extended SPCR within the framework of generalized linear models. However, it is unclear whether the PCA loss function in Zou et al. (2006) is the best choice for building SPCR, as there exist several formulae for PCA.

This paper proposes a novel formulation for SPCR. As a PCA loss for SPCR, we adopt a loss function based on a singular value decomposition approach (Shen and Huang 2008). Using the basic loss function, a combination of the PCA loss and the regression squared loss, with sparse regularization, we derive an alternative formulation for SPCR. We call the proposed method as sparse principal component regression based on a singular value decomposition approach (SPCRsvd). An estimation algorithm of SPCRsvd is developed using an alternating direction method of multipliers (Boyd et al. 2011) and a linearized alternating direction method of multipliers (Li et al. 2014; Wang and Yuan 2012). We show the effectiveness of SPCRsvd through numerical studies. Specifically, the performance of SPCRsvd is shown to be competitive with or better than that of SPCR.

As an alternative approach, partial least squares (PLS) (Frank and Friedman 1993; Wold 1975) is a widely used statistical method that regresses a response variable on composite variables built by combining a response variable and explanatory variables. In Chun and Keleş (2010), sparse partial least squares (SPLS) was proposed, which enables the removal of irrelevant explanatory variables when constructing the composite variables. PLS and SPLS are similar to SPCR and SPCRsvd in terms of using new explanatory variables with information relating the response variable to the original explanatory variables. Herein, these methods are compared using simulated data and real data.

The remainder of the paper is organized as follows. In Sect. 2, we review SPCA in Zou et al. (2006) and Shen and Huang (2008), and SPCR in Kawano et al. (2015). We present SPCRsvd in Sect. 3. Section 4 derives two computational algorithms for SPCRsvd and discusses the selection of tuning parameters. Monte Carlo simulations and real data analyses are presented in Sect. 5. Conclusions are given in Sect. 6.

2 Preliminaries

2.1 Sparse principal component analysis

PCA finds a loading matrix that induces a low-dimensional structure in the data. As an easy way to interpret the principal component loading matrix, SPCA has been proposed. To date, several formulae for SPCA have been proposed (Bresler et al. 2018; Chen et al. 2020; d'Aspremont et al. 2007; Erichson et al. 2020; Shen and Huang 2008; Vu et al. 2013; Witten et al. 2009; Zou et al. 2006). For an overview of SPCA, we refer the reader to Zou and Xue (2018) and the references therein. In this subsection, we review the two formulae for SPCA in Zou et al. (2006) and Shen and Huang (2008).

Let $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ denote an $n \times p$ data matrix, where n and p are the number of observations and the number of variables, respectively. Without loss of generality, we assume that the columns of the matrix X are centered. In Zou et al. (2006), SPCA was proposed as

$$\min_{A,B} \left\{ \sum_{i=1}^{n} \|\boldsymbol{x}_i - AB^\top \boldsymbol{x}_i\|_2^2 + \lambda \sum_{j=1}^{k} \|\boldsymbol{\beta}_j\|_2^2 + \sum_{j=1}^{k} \lambda_{1,j} \|\boldsymbol{\beta}_j\|_1 \right\} \quad \text{subject to } A^\top A = I_k, \tag{1}$$

where $A = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ and $B = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k)$ are $p \times k$ principal component (PC) loading matrices, k denotes the number of principal components, I_k is the $k \times k$ identity matrix, $\lambda, \lambda_{1,1}, \ldots, \lambda_{1,k}$ are non-negative regularization parameters, and $\|\cdot\|_q$ is the L_q norm for an arbitrary finite vectors. This SPCA formulation can be regarded as a least squares approach. The first term represents performing PCA by least squares. The second and third terms represent sparse regularization similar to elastic net regularization (Zou and Hastie 2005). These terms enable us to set some of the estimates of B to zero. If $\lambda = 0$, then the regularization terms reduce to the adaptive lasso (Zou 2006).

A simple calculation gives

$$\min_{A,B} \sum_{j=1}^{k} \left\{ \|X\boldsymbol{\alpha}_j - X\boldsymbol{\beta}_j\|_2^2 + \lambda \|\boldsymbol{\beta}_j\|_2^2 + \lambda_{1,j} \|\boldsymbol{\beta}_j\|_1 \right\} \quad \text{subject to } A^\top A = I_k. \tag{2}$$

Optimizing the parameters A and B for this minimization problem is straightforward. Given a fixed A, the SPCA problem (2) turns out to be a simple elastic net problem. Thus, the estimate of B can be obtained by the least angle regression algorithm (Efron et al. 2004) or the coordinate descent algorithm (Friedman et al. 2007; Wu and Lange 2008). Given a fixed B, an estimate of A can be obtained by solving the reduced rank Procrustes rotation problem (Zou et al. 2006). By alternating procedures, we can obtain the final estimates $\hat{A}$ and $\hat{B}$ of A and B, respectively. Note that only $\hat{B}$ is used as the principal component loading matrix.

Alternately, Shen and Huang (2008) proposed another formulation of SPCA, which can be regarded as a singular value decomposition (SVD) approach. Consider a low-rank approximation of the data matrix X obtained by SVD in the form

$$UDV^\top = \sum_{k=1}^{r} d_k \boldsymbol{u}_k \boldsymbol{v}_k^\top, \tag{3}$$

where $U = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r)$ is an $n \times r$ matrix with $U^\top U = I_r$, $V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r)$ is an $r \times r$ orthogonal matrix, $D = \mathrm{diag}(d_1, \ldots, d_r)$, and $r < \min(n, p)$. The singular values are assumed to be ordered such that $d_r \geq \cdots \geq d_p \geq 0$. Using the connection between PCA and SVD, Shen and Huang (2008) obtained the sparse PC loading by estimating V with sparse regularization.

To achieve sparseness of V, Shen and Huang (2008) adopted the rank-one approximation procedure. First, the first PC loading vector $\tilde{\boldsymbol{v}}_1$ is obtained by solving the minimization problem

$$\min_{\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{v}}_1} \left\{ \|X - \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^\top\|_F^2 + \lambda P(\tilde{\boldsymbol{v}}_1) \right\} \quad \text{subject to} \quad \|\tilde{\boldsymbol{u}}_1\|_2 = 1. \tag{4}$$

Here $\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{v}}_1$ are defined as rescaled vectors such that $\tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^\top = d_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top$, $P(\cdot)$ is a penalty function that induces the sparsity of $\tilde{\boldsymbol{v}}_1$, and $\|\cdot\|_F$ is the Frobenius norm defined by $\|A\|_F = \sqrt{\mathrm{tr}(A^\top A)}$ for an arbitrary matrix A. As the penalty function, Shen and Huang (2008) used the lasso penalty (Tibshirani 1996), the hard-thresholding penalty (Donoho and Johnstone 1994), or the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li 2001). The rank-one approximation problem is easy to solve (4); see Algorithm 1 in Shen and Huang (2008). The remaining PC loading vectors are obtained by performing rank-one approximations of the corresponding residual matrices. For example, to derive the second PC loading vector $\tilde{\boldsymbol{v}}_2$, we solve the minimization problem

$$\min_{\tilde{\boldsymbol{u}}_2, \tilde{\boldsymbol{v}}_2} \left\{ \|X^\dagger - \tilde{\boldsymbol{u}}_2 \tilde{\boldsymbol{v}}_2^\top\|_F^2 + \lambda P(\tilde{\boldsymbol{v}}_2) \right\} \quad \text{subject to} \quad \|\tilde{\boldsymbol{u}}_2\|_2 = 1,$$

where $X^\dagger = X - \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^\top$. The regularization parameter λ is selected by cross-validation.

2.2 Sparse principal component regression

For a one-dimensional continuous response variable Y and a p-dimensional explanatory variable $\boldsymbol{x}$, suppose we have obtained a dataset $\{(y_i, \boldsymbol{x}_i); i = 1, \ldots, n\}$. We assume that the response variable is explained by variables composed by PCA of $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$. Traditional PCR uses a regression model with a few PC scores corresponding to large eigenvalues. Note that these PC scores are derived by PCA prior to the regression. This two-stage procedure might then fail to predict the response if the response variable is related to PCs corresponding to small eigenvalues.

To attain a one-stage procedure for PCR, the SPCR proposed in Kawano et al. (2015) was formulated as the following minimization problem:

$$\min_{A,B,\gamma_0,\boldsymbol{\gamma}} \left\{ \sum_{i=1}^{n} \left(y_i - \gamma_0 - \boldsymbol{\gamma}^\top B^\top \boldsymbol{x}_i \right)^2 + w \sum_{i=1}^{n} \|\boldsymbol{x}_i - AB^\top \boldsymbol{x}_i\|_2^2 \right.$$
$$\left. + \lambda_\beta \xi \sum_{j=1}^{k} \|\boldsymbol{\beta}_j\|_2^2 + \lambda_\beta (1-\xi) \sum_{j=1}^{k} \|\boldsymbol{\beta}_j\|_1 + \lambda_\gamma \|\boldsymbol{\gamma}\|_1 \right\} \quad (5)$$
$$\text{subject to} \quad A^\top A = I_k,$$

where γ_0 is an intercept, $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_k)^\top$ comprises coefficients for regression, λ_β and λ_γ are non-negative regularization parameters, w is a positive tuning parameter, and ξ in [0, 1] is a tuning parameter. The first term in Formula (5) is the regression squared loss function including the PCs $B^\top \boldsymbol{x}$ as explanatory variables, while the second term is the PCA loss function used in SPCA in Zou et al. (2006). Sparse regularization in SPCR has two roles: sparseness and identifiability of parameters. For the identifiability by sparse regularization, we refer the reader to Choi et al. (2010), Jennrich (2006), Kawano et al. (2015). Kawano et al. (2018) also extended SPCR from the viewpoint of generalized linear models, which can deal with binary, count, and multi-categorical data for the response variable.

3 SVD-based sparse principal component regression

SPCR uses two basic loss functions: the regression squared loss function and the PCA loss function in Zou et al. (2006). However, it is unclear whether the PCA loss is the best choice for building SPCR. To investigate this issue, we propose another formulation for SPCR using the SVD approach in Shen and Huang (2008).

We consider the following minimization problem:

$$\min_{\beta_0,\boldsymbol{\beta},Z,V} \left\{ \frac{1}{n} \|\boldsymbol{y} - \beta_0 \mathbf{1}_n - XV\boldsymbol{\beta}\|_2^2 + \frac{w}{n} \|X - ZV^\top\|_F^2 + \lambda_V \|V\|_1 + \lambda_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 \right\}$$
$$\text{subject to} \quad V^\top V = I_k, \quad (6)$$

where β_0 is an intercept, k is the number of PCs, $\boldsymbol{\beta}$ is a k-dimensional coefficient vector, Z is an $n \times k$ matrix of PCs, V is a $p \times k$ PC loading matrix, and $\mathbf{1}_n$ is an n-dimensional vector of ones. In addition, w is a positive tuning parameter and λ_V, $\lambda_{\boldsymbol{\beta}}$ are non-negative regularization parameters.

The first term is the regression squared loss function relating the response and the PCs XV. The second term is the PCA loss function in the SVD approach in Shen and Huang (2008). Although the formula is seemingly different from the first term in Formula (4), they are essentially equivalent: we estimate the k PCs simultaneously, while Shen and Huang (2008) estimates them sequentially. The third and fourth terms constitute the lasso penalty that induces zero estimates of the parameters V and $\boldsymbol{\beta}$,

respectively. The tuning parameter w controls the degree of the second term. A smaller value for w is used when our aim is to obtain better prediction accuracies, while a larger value for w is used when we want to obtain exact expressions of the PC loadings. The minimization problem (6) allows us to perform regression analysis and PCA simultaneously. We call this method SPCRsvd. In Sect. 5, we will observe that SPCRsvd is competitive with or better than SPCR through numerical studies.

We remark on two points here. First, it is possible to use Z in the first term of (6) instead of XV, since Z is also the PCs. However, the formulation with Z instead of XV did not perform well in numerical studies, so we adopt the formulation with XV here. Second, SPCR imposes a ridge penalty for the PC loading but SPCRsvd does not. The ridge penalty basically comes from SPCA in Zou et al. (2006). Because SPCRsvd is not based on SPCA in Zou et al. (2006), a ridge penalty does not appear in Formula (6). It is possible to add a ridge penalty and replace the lasso penalty with other penalties that induce sparsity, e.g., the adaptive lasso penalty, the SCAD penalty, or minimax concave penalty (Zhang 2010), but the our aim of this paper is to establish the basic procedure of Formula (6).

4 Implementation

4.1 Computational algorithm

To obtain the estimates of the parameters $\boldsymbol{\beta}, Z, V$ in Formula (6), we employ the alternating direction method of multipliers (ADMM) and the linearized alternating direction method of multipliers (LADMM). ADMM and LADMM have recently been used in various models with sparse regularization; see, for example, Boyd et al. (2011); Danaher et al. (2014); Li et al. (2014); Ma and Huang (2017); Price et al. (2019); Tan et al. (2014); Wang et al. (2018); Yan and Bien (2020) and Ye and Xie (2011).

To solve the minimization problem (6) by using ADMM, we rewrite the problem as

$$\begin{aligned}&\min_{\beta_0,\boldsymbol{\beta},\boldsymbol{\beta}_0,Z,V,V_0,V_1}\left\{\frac{1}{n}\|\boldsymbol{y}-\beta_0\mathbf{1}_n-XV_1\boldsymbol{\beta}\|_2^2+\frac{w}{n}\|X\right.\\&\left.-ZV^\top\|_F^2+\lambda_V\|V_0\|_1+\lambda_{\boldsymbol{\beta}}\|\boldsymbol{\beta}_0\|_1\right\}\\&\text{subject to}\quad V^\top V=I_k,\quad V=V_0=V_1,\quad \boldsymbol{\beta}=\boldsymbol{\beta}_0.\end{aligned}\tag{7}$$

The scaled augmented Lagrangian for the problem (7) is then given by

$$\begin{aligned}&\frac{1}{n}\|\boldsymbol{y}-\beta_0\mathbf{1}_n-XV_1\boldsymbol{\beta}\|_2^2+\frac{w}{n}\|X-ZV^\top\|_F^2+\lambda_V\|V_0\|_1+\lambda_{\boldsymbol{\beta}}\|\boldsymbol{\beta}_0\|_1\\&+\frac{\rho_1}{2}\|V-V_0+\Lambda_1\|_F^2+\frac{\rho_2}{2}\|V_1-V_0+\Lambda_2\|_F^2+\frac{\rho_3}{2}\|\boldsymbol{\beta}-\boldsymbol{\beta}_0+\boldsymbol{\lambda}_3\|_2^2\\&\text{subject to}\quad V^\top V=I_k,\end{aligned}$$

where Λ_1, Λ_2, $\boldsymbol{\lambda}_3$ are dual variables and ρ_1, ρ_2, ρ_3 (> 0) are penalty parameters. This gives rise to the following ADMM algorithm:

Step 1 Set the values of the tuning parameter w, the regularization parameters λ_V, $\lambda_{\boldsymbol{\beta}}$, and the penalty parameters ρ_1, ρ_2, ρ_3.

Step 2 Initialize all the parameters as $\beta_0^{(0)}$, $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{\beta}_0^{(0)}$, $Z^{(0)}$, $V^{(0)}$, $V_0^{(0)}$, $V_1^{(0)}$, $\Lambda_1^{(0)}$, $\Lambda_2^{(0)}$, $\boldsymbol{\lambda}_3^{(0)}$.

Step 3 For $m = 0, 1, 2, \ldots$, repeat from Steps 4 to 11 until convergence.

Step 4 Update V_1 as follows:

$$\begin{aligned}\mathrm{vec}(V_1^{(m+1)}) = \left(\frac{1}{n}\boldsymbol{\beta}^{(m)}\boldsymbol{\beta}^{(m)\top} \otimes X^\top X + \frac{\rho_2}{2} I_k \otimes I_p\right)^{-1} \\ \mathrm{vec}\left\{\frac{1}{n}X^\top(\boldsymbol{y} - \beta_0^{(m)}\mathbf{1}_n)\boldsymbol{\beta}^{(m)\top} \right. \\ \left. + \frac{\rho_2}{2}(V_0^{(m)} - \Lambda_2^{(m)})\right\},\end{aligned}$$

where $\otimes$ represents the Kronecker product.

Step 5 Update V as follows:

$$V^{(m+1)} = PQ^\top,$$

where P and Q are the matrices given by the SVD

$$\frac{w}{n}X^\top Z^{(m)} + \frac{\rho_1}{2}\left(V_0^{(m)} - \Lambda_1^{(m)}\right) = P\Omega Q^\top.$$

Step 6 Update V_0 as follows:

$$\begin{aligned}&v_{0ij}^{(m+1)} = \mathcal{S}\left(\frac{\rho_1(v_{ij}^{(m+1)} + \lambda_{1ij}^{(m)}) + \rho_2(v_{ij}^{(m+1)} + \lambda_{2ij}^{(m)})}{\rho_1 + \rho_2}, \frac{\lambda_V}{\rho_1 + \rho_2}\right), \\ &i = 1, \ldots, p, \ \ j = 1, \ldots, k,\end{aligned}$$

where $v_{0ij}^{(m)} = (V_0^{(m)})_{ij}$, $v_{ij}^{(m)} = (V^{(m)})_{ij}$, $\lambda_{\ell ij}$ ($\ell = 1, 2$) is the (i, j)-th element of the matrix Λ_ℓ ($\ell = 1, 2$), and $\mathcal{S}(\cdot, \cdot)$ is the soft-thresholding operator defined by $\mathcal{S}(x, \lambda) = \mathrm{sign}(x)(|x| - \lambda)_+$.

Step 7 Update Z by $Z^{(m+1)} = XV^{(m+1)}$.

Step 8 Update $\boldsymbol{\beta}$ as follows:

$$\begin{aligned}\boldsymbol{\beta}^{(m+1)} = \left(\frac{1}{n}V_1^{(m+1)\top}X^\top X V_1^{(m+1)} + \frac{\rho_3}{2}I_k\right)^{-1}\left\{\frac{1}{n}V_1^{(m+1)T}X^\top(\boldsymbol{y} - \beta_0^{(m)}\mathbf{1}_n)\right. \\ \left. + \frac{\rho_3}{2}(\boldsymbol{\beta}_0^{(m)} - \boldsymbol{\lambda}_3^{(m)})\right\}.\end{aligned}$$

Step 9 Update $\boldsymbol{\beta}_0$ as follows:

$$\beta_{0j}^{(m+1)} = \mathcal{S}\left(\beta_j^{(m+1)} + \lambda_{3j}^{(m)}, \frac{\lambda_\beta}{\rho_3}\right), \quad j = 1, \ldots, k,$$

where $\lambda_{3j}^{(m)}$ and $\beta_j^{(m)}$ are the j-th elements of the vectors $\boldsymbol{\lambda}_3^{(m)}$ and $\boldsymbol{\beta}^{(m)}$, respectively.

Step 10 Update β_0 as follows:

$$\beta_0^{(m+1)} = \frac{1}{n}\mathbf{1}_n^\top(\boldsymbol{y} - XV_1^{(m+1)}\boldsymbol{\beta}^{(m+1)}).$$

Step 11 Update $\Lambda_1, \Lambda_2, \boldsymbol{\lambda}_3$ as follows:

$$\begin{aligned}
\Lambda_1^{(m+1)} &= \Lambda_1^{(m)} + V^{(m+1)} - V_0^{(m+1)},\\
\Lambda_2^{(m+1)} &= \Lambda_2^{(m)} + V_1^{(m+1)} - V_0^{(m+1)},\\
\boldsymbol{\lambda}_3^{(m+1)} &= \boldsymbol{\lambda}_3^{(m)} + \boldsymbol{\beta}^{(m+1)} - \boldsymbol{\beta}_0^{(m+1)}.
\end{aligned}$$

The derivations of the updates are given in "Appendix A".

To apply LADMM to the minimization problem (6), we consider the following problem:

$$\begin{aligned}
&\min_{\beta_0, \boldsymbol{\beta}, \boldsymbol{\beta}_0, Z, V, V_0} \left\{ \frac{1}{n}\|\boldsymbol{y} - \beta_0\mathbf{1}_n - XV_0\boldsymbol{\beta}\|_2^2 + \frac{w}{n}\|X - ZV^\top\|_F^2 + \lambda_V\|V_0\|_1 + \lambda_{\boldsymbol{\beta}}\|\boldsymbol{\beta}_0\|_1 \right\}\\
&\text{subject to} \quad V^\top V = I_k, \quad V = V_0, \quad \boldsymbol{\beta} = \boldsymbol{\beta}_0. \qquad (8)
\end{aligned}$$

The augmented Lagrangian for this problem is given by

$$\begin{aligned}
&\frac{1}{n}\|\boldsymbol{y} - \beta_0\mathbf{1}_n - XV_0\boldsymbol{\beta}\|_2^2 + \frac{w}{n}\|X - ZV^\top\|_F^2 + \lambda_V\|V_0\|_1 + \lambda_{\boldsymbol{\beta}}\|\boldsymbol{\beta}_0\|_1\\
&+ \frac{\rho_1}{2}\|V_0 - V + \Lambda\|_F^2 + \frac{\rho_2}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}_0 + \boldsymbol{\lambda}\|_2^2\\
&\text{subject to} \quad V^\top V = I_k,
\end{aligned}$$

where $\Lambda, \boldsymbol{\lambda}$ are dual variables and ρ_1, ρ_2 (> 0) are penalty parameters.

The updates of the LADMM algorithm are almost the same as those of the ADMM algorithm. We summarize the updates and the derivations in "Appendix B".

Here we remark on the main differences between ADMM and LADMM. LADMM has two penalty parameters (ρ_1, ρ_2), while ADMM has three penalty parameters (ρ_1, ρ_2, ρ_3). This means that the total number of tuning parameters in LADMM is only one less than that in ADMM. This is an advantage of LADMM regardless of whether the user tunes the penalty parameters subjectively or objectively. On the other hand, approximation by Taylor expansion is used in LADMM. If this approximation is inappropriate, LADMM may fail to estimate parameters. In terms of running times,

ADMM seems to be faster than LADMM, based on several numerical studies. These results will be presented in Sect. 6 when discussing the limitations of the current study.

4.2 Determination of tuning parameters

We have the six tuning parameters: $w, \lambda_V, \lambda_{\boldsymbol{\beta}}, \rho_1, \rho_2, \rho_3$. The penalty parameters ρ_1, ρ_2, ρ_3 are fixed as $\rho_1 = \rho_2 = \rho_3 = 1$ in accordance with Boyd et al. (2011). The tuning parameter w is set according to the purpose of the analysis. A small value is allocated to w when the user considers the regression loss to be more important than the PCA loss. This idea follows Kawano et al. (2015, 2018).

The two regularization parameters $\lambda_V, \lambda_{\boldsymbol{\beta}}$ are objectively selected by K-fold cross-validation. For the original dataset divided into the K datasets $(\boldsymbol{y}^{(1)}, X^{(1)}), \ldots, (\boldsymbol{y}^{(K)}, X^{(K)})$, the criterion for the K-fold cross-validation in ADMM is given by

$$\mathrm{CV} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\left\|\boldsymbol{y}^{(k)} - \hat{\beta}_0^{(-k)}\mathbf{1}_{(k)} - X^{(k)}\hat{V}_1^{(-k)}\hat{\boldsymbol{\beta}}^{(-k)}\right\|_2^2, \tag{9}$$

where $\hat{\beta}_0^{(-k)}$, $\hat{V}_1^{(-k)}$, $\hat{\boldsymbol{\beta}}^{(-k)}$ are the estimates of β_0, V_1, $\boldsymbol{\beta}$, respectively, computed with the data excluding the k-th dataset. We omit the CV criterion for LADMM, since we only replace $\hat{V}_1^{(-k)}$ in (9) with $\hat{V}_0^{(-k)}$.

We choose the values of the regularization parameters $\lambda_V, \lambda_{\boldsymbol{\beta}}$ from the minimizers of CV in (9).

5 Numerical study

5.1 Monte Carlo simulations

We conducted Monte Carlo simulations to investigate the effectiveness of SPCRsvd. The simulations had six cases, which were the same as those in Kawano et al. (2015) except for Case 6. These six cases are given as follows.

Case 1 The 10-dimensional covariate vector $\boldsymbol{x} = (x_1, \ldots, x_{10})$ follows a multivariate normal distribution having a zero mean vector and variance-covariance matrix Σ. The response was obtained by

$$y_i = \zeta_1 \boldsymbol{e}_1^\top \boldsymbol{x}_i + \zeta_2 \boldsymbol{e}_2^\top \boldsymbol{x}_i + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $\boldsymbol{e}_1 = (1, \underbrace{0, \ldots, 0}_{9})^\top$, $\boldsymbol{e}_2 = (0, 1, \underbrace{0, \ldots, 0}_{8})^\top$, and ε_i are independently distributed as a normal distribution with mean zero and variance σ^2. We used $\zeta_1 = 2, \zeta_2 = 1, \Sigma = I_{10}$. Then we note that $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are eigenvectors of Σ.

Case 2 This case is the same as Case 1 except with $\zeta_1 = 8, \zeta_2 = 1, \Sigma = \mathrm{diag}(1, 3^2, \underbrace{1, \ldots, 1}_{8})$. Then $\boldsymbol{e}_2$ becomes the first eigenvector. In addition,

$\mathrm{Cov}(y, x_1) = 8$ and $\mathrm{Cov}(y, x_2) = 9$. For more details of this setting, we refer to p. 196 in Kawano et al. (2015).

Case 3 The 20-dimensional covariate vector $\boldsymbol{x} = (x_1, \ldots, x_{20})$ has multivariate normal distribution $N_{20}(\boldsymbol{0}, \Sigma)$. The response was obtained as

$$y_i = 4\boldsymbol{\zeta}^\top \boldsymbol{x}_i + \varepsilon_i, \quad i = 1, \ldots, n,$$

where ε_i are independently distributed as $N(0, \sigma^2)$. We used $\boldsymbol{\zeta} = (\boldsymbol{v}, \underbrace{0, \ldots, 0}_{11})^\top$ and $\Sigma = \text{block diag}(\Sigma_1, I_{11})$, where $\boldsymbol{v} = (-1, 0, 1, 1, 0, -1, -1, 0, 1)$ and $(\Sigma_1)_{ij} = 0.9^{|i-j|}$ $(i, j, = 1, \ldots, 9)$. Note that $\boldsymbol{v}$ is a sparse approximation of the fourth eigenvector of Σ_1. This case deals with the situation where the response is associated with the fourth principal component.

Case 4 The 30-dimensional covariate vector $\boldsymbol{x} = (x_1, \ldots, x_{30})$ has multivariate normal distribution $N_{30}(\boldsymbol{0}, \Sigma)$. The response was obtained as

$$y_i = 4\boldsymbol{\zeta}_1^\top \boldsymbol{x}_i + 4\boldsymbol{\zeta}_2^\top \boldsymbol{x}_i + \varepsilon_i, \quad i = 1, \ldots, n,$$

where ε_i are independently distributed as $N(0, \sigma^2)$. We used $\boldsymbol{\zeta}_1 = (\boldsymbol{v}_1, \underbrace{0, \ldots, 0}_{21})^\top$, $\boldsymbol{\zeta}_2 = (\underbrace{0, \ldots, 0}_{9}, \boldsymbol{v}_2, \underbrace{0, \ldots, 0}_{15})^\top$, $\Sigma = \text{block diag}(\Sigma_1, \Sigma_2, I_{15})$. Here $\boldsymbol{v}_1 = (-1, 0, 1, 1, 0, -1, -1, 0, 1)$, $\boldsymbol{v}_2 = (\underbrace{1, \ldots, 1}_{6})$, and $(\Sigma_2)_{ij} = 0.9^{|i-j|}$ $(i, j, = 1, \ldots, 6)$. Note that $\boldsymbol{v}_1$ is a sparse approximation of the third eigenvector of Σ_1 and $\boldsymbol{v}_2$ is the first eigenvector of Σ_2. This case deals with the situation where the response is associated with the third principal component from Σ_1 and the first principal component from Σ_2.

Case 5 This case is the same as Case 4 except with $\boldsymbol{v}_2 = (1, 0, -1, -1, 0, 1)$. Note that $\boldsymbol{v}_2$ is a sparse approximation of the third eigenvector of Σ_2. This case deals with the situation where the response is associated with the third principal components from Σ_1 and Σ_2.

Case 6 This case is the same as Case 2 except with $\boldsymbol{x} = (x_1, \ldots, x_{100})$. This is a high-dimensional case of Case 2.

The sample size was set to $n = 50, 200$. The standard deviation was set to $\sigma = 1, 2$. We considered the two algorithms given in Sect. 4.1: ADMM for SPCRsvd (SPCRsvd-ADMM) and LADMM for SPCRsvd (SPCRsvd-LADMM). SPCRsvd was fitted to the simulated data with one or five components ($k = 1, 5$) except for Case 6 and one or two components ($k = 1, 2$) for Case 6. We set the value of the tuning parameter w to 0.1 and employed five-fold cross-validation for selecting the regularization parameters $\lambda_V, \lambda_{\boldsymbol{\beta}}$. We used a two-dimensional grid and evaluated the CV in (9) on the grid, as illustrated in Fig. 1. The cross-validation surface was obtained by SPCRsvd-ADMM with $k = 1$ and was estimated by data generated from Case 1 with $n = 50, \sigma = 1$. The minimum is achieved for the combination of the first candidate of λ_V and the seventh candidate of $\lambda_{\boldsymbol{\beta}}$.

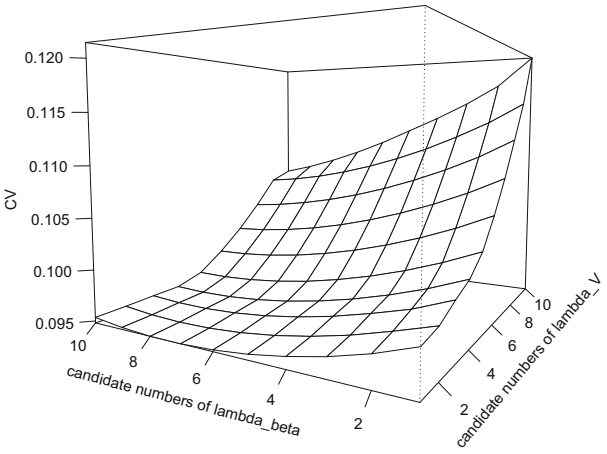

Fig. 1 Cross-validation surface in SPCRsvd-ADMM estimated by data generated from Case 1

SPCRsvd was compared with SPCR, PCR, SPLS, and PLS. SPCR was computed by the package **spcr**, SPLS by **spls**, and PLS and PCR by **pls**. These packages are included in the software R (R Core Team 2020). We used the default settings of the packages when determining the values of tuning parameters in SPCR, PCR, SPLS, and PLS. The values of the tuning parameters w and ξ in SPCR were set to 0.1 and 0.01, respectively, and then the regularization parameters were selected by five-fold cross-validation. The value of the regularization parameter in SPLS was selected by 10-fold cross-validation. The number of components in SPLS, PLS, and PCR was also selected by 10-fold cross-validation from ranges from one to five when SPCRsvd-ADMM, SPCRsvd-LADMM, and SPCR employ $k = 5$ and from one to two when $k = 2$. The performance was evaluated in terms of MSE $= E[(y-\hat{y})^2]$. The simulation was conducted 100 times. MSE was estimated from 1,000 random samples.

We summarize the means and standard deviations of MSEs in Tables 1, 2, 3, 4, 5 and 6. The results for $\sigma = 1, 2$ had similar tendencies. PCR and PLS were worst in almost all cases, so we will focus on comparing the other methods. SPCRsvd-LADMM and SPCRsvd-ADMM were competitive with SPCR. In particular, SPCRsvd-LADMM and SPCRsvd-ADMM provided smaller MSEs than SPCR in almost all cases when $k = 1$. Compared to SPLS, SPCRsvd-LADMM and SPCRsvd-ADMM were slightly inferior in many cases when $k = 5$. However, SPLS produced so large values of MSEs in many cases when $k = 1$.

The true positive rate (TPR), the true negative rate (TNR), and the Matthews correlation coefficient (MCC) (Matthews 1975) were also computed for SPCRsvd-LADMM, SPCRsvd-ADMM, SPCR, and SPLS. TPR and TNR are respectively defined by

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\left|\left\{j : \zeta_j^* \neq 0\right\}\right|} = \frac{1}{100}\sum_{k=1}^{100} \frac{\left|\left\{j : \hat{\zeta}_j^{(k)} \neq 0 \wedge \zeta_j^* \neq 0\right\}\right|}{\left|\left\{j : \zeta_j^* \neq 0\right\}\right|},$$

Table 1 Mean (standard deviation) of MSE for Case 1

σ	n	k	SPCRsvd-LADMM	SPCRsvd-ADMM	SPCR	SPLS	PLS	PCR
1	50	1	1.584	**1.162**	2.130	1.455	1.999	5.663
			(1.322)	(0.119)	(1.932)	(0.472)	(0.433)	(0.646)
		5	1.324	**1.174**	1.291	1.124	1.284	3.791
			(0.878)	(0.114)	(0.712)	(0.134)	(0.139)	(1.083)
	200	1	1.698	**1.033**	3.863	1.030	1.256	5.598
			(1.715)	(0.061)	(2.474)	(0.052)	(0.122)	(0.559))
		5	1.240	**1.039**	1.761	1.021	1.054	3.568
			(1.001)	(0.059)	(1.736)	(0.050)	(0.052)	(0.978)
2	50	1	5.288	**4.736**	5.469	5.004	5.426	8.765
			(1.394)	(0.448)	(1.470)	(0.647)	(0.613)	(0.746)
		5	4.795	**4.733**	4.936	4.692	5.129	7.091
			(0.848)	(0.427)	(0.661)	(0.556)	(0.550)	(1.118)
	200	1	4.856	**4.127**	7.216	4.107	4.386	8.606
			(1.782)	(0.221)	(2.347)	(0.238)	(0.245)	(0.633)
		5	4.376	**4.139**	4.530	4.086	4.217	6.631
			(0.995)	(0.214)	(1.285)	(0.200)	(0.208)	(0.995)

The bold values correspond to the smallest means among SPCRsvd-LADMM, SPCRsvd-ADMM, and SPCR

Table 2 Mean (standard deviation) of MSE for Case 2

σ	n	k	SPCRsvd-LADMM	SPCRsvd-ADMM	SPCR	SPLS	PLS	PCR
1	50	1	1.957	**1.223**	1.278	41.436	47.118	67.487
			(7.338)	(0.161)	(0.141)	(18.707)	(11.370)	(4.646)
		5	2.046	**1.230**	1.887	1.109	1.306	39.232
			(7.332)	(0.145)	(7.363)	(0.120)	(0.151)	(13.636)
	200	1	**1.037**	1.041	1.054	40.502	46.295	65.608
			(0.063)	(0.061)	(0.051)	(15.652)	(5.246)	(3.078)
		5	1.055	1.039	**1.030**	1.023	1.054	35.782
			(0.098)	(0.055)	(0.052)	(0.049)	(0.052)	(13.074)
2	50	1	6.414	5.664	**5.098**	43.830	50.346	70.566
			(10.200)	(7.298)	(0.546)	(19.342)	(11.393)	(4.921)
		5	5.848	**5.018**	5.380	4.424	5.148	42.498
			(7.273)	(0.582)	(7.302)	(0.470)	(0.553)	(13.706)
	200	1	**4.178**	4.189	4.217	42.876	49.259	68.664
			(0.2383)	(0.233)	(0.207)	(16.265)	(5.550)	(3.317)
		5	4.228	4.189	**4.114**	4.091	4.216	38.880
			(0.241)	(0.227)	(0.205)	(0.202)	(0.208)	(13.069)

The bold values correspond to the smallest means among SPCRsvd-LADMM, SPCRsvd-ADMM, and SPCR

Table 3 Mean (standard deviation) of MSE for Case 3

σ	n	k	SPCRsvd-LADMM	SPCRsvd-ADMM	SPCR	SPLS	PLS	PCR
1	50	1	**1.564**	1.581	1.793	20.625	20.847	21.404
			(0.314)	(0.331)	(2.160)	(1.924)	(2.012)	(1.295)
		5	1.663	1.933	**1.563**	1.998	3.398	22.244
			(0.437)	(0.602)	(0.316)	(1.192)	(1.442)	(1.475)
	200	1	**1.085**	1.098	1.096	15.259	16.817	20.642
			(0.068)	(0.072)	(0.069)	(4.717)	(2.886)	(0.863)
		5	1.114	1.144	**1.096**	1.089	1.158	20.759
			(0.083)	(0.105)	(0.070)	(0.240)	(0.080)	(0.917)
2	50	1	6.412	**6.408**	6.562	24.353	24.423	24.520
			(1.279)	(1.247)	(2.057)	(2.389)	(2.342)	(1.441)
		5	6.615	6.829	**6.349**	6.525	8.000	25.519
			(1.591)	(1.832)	(1.258)	(2.178)	(2.183)	(1.730)
	200	1	**4.579**	4.610	4.766	19.078	20.220	23.627
			(1.963)	(1.961)	(2.632)	(4.390)	(2.733)	(1.002)
		5	4.654	**4.451**	4.763	4.272	4.430	23.776
			(1.963)	(0.300)	(2.632)	(0.361)	(0.272)	(1.063)

The bold values correspond to the smallest means among SPCRsvd-LADMM, SPCRsvd-ADMM, and SPCR

Table 4 Mean (standard deviation) of MSE for Case 4

σ	n	k	SPCRsvd-LADMM	SPCRsvd-ADMM	SPCR	SPLS	PLS	PCR
1	50	1	2.595	**2.302**	2.307	21.540	47.460	433.826
			(1.542)	(0.593)	(0.619)	(1.389)	(23.355)	(114.041)
		5	2.720	2.646	**2.249**	6.0157	11.939	33.604
			(1.903)	(0.819)	(0.558)	(5.308)	(3.919)	(7.875)
	200	1	1.160	1.176	**1.158**	21.018	24.899	477.828
			(0.075)	(0.077)	(0.076)	(0.991)	(5.165)	(37.972)
		5	1.165	1.201	**1.158**	1.183	1.701	23.414
			(0.079)	(0.103)	(0.077)	(0.106)	(0.261)	(1.582)
2	50	1	9.695	**9.511**	9.667	24.983	50.747	437.040
			(3.176)	(2.290)	(2.413)	(1.946)	(23.481)	(114.199)
		5	10.734	9.552	**9.511**	12.712	17.237	36.904
			(4.010)	(2.480)	(2.320)	(6.581)	(4.258)	(7.903)
	200	1	4.705	4.695	**4.662**	24.103	27.978	480.882
			(0.304)	(0.303)	(0.310)	(1.213)	(5.196)	(37.853)
		5	4.764	4.744	**4.660**	4.458	5.219	26.522
			(0.390)	(0.319)	(0.312)	(0.305)	(0.462)	(1.730)

The bold values correspond to the smallest means among SPCRsvd-LADMM, SPCRsvd-ADMM, and SPCR

Table 5 Mean (standard deviation) of MSE for Case 5

σ	n	k	SPCRsvd-LADMM	SPCRsvd-ADMM	SPCR	SPLS	PLS	PCR
1	50	1	2.155	2.207	**2.144**	35.283	35.094	34.654
			(0.501)	(0.577)	(0.524)	(3.264)	(2.726)	(1.806)
		5	2.574	3.171	**2.113**	10.190	16.033	35.537
			(1.142)	(1.654)	(0.512)	(6.852)	(5.439)	(2.125)
	200	1	**1.151**	1.493	1.506	30.629	30.876	34.297
			(0.076)	(3.318)	(3.491)	(2.614)	(2.497)	(1.602)
		5	1.208	1.220	**1.156**	1.236	1.814	34.208
			(0.112)	(0.105)	(0.076)	(0.167)	(0.304)	(1.580)
2	50	1	**8.949**	8.985	9.236	38.659	38.612	37.800
			(2.151)	(2.133)	(3.420)	(3.369)	(3.104)	(2.004)
		5	9.671	9.405	**8.848**	17.133	21.495	38.780
			(2.993)	(2.447)	(2.170)	(8.875)	(6.056)	(2.355)
	200	1	**4.654**	4.675	4.999	34.093	34.301	37.374
			(0.300)	(0.306)	(3.575)	(2.778)	(2.681)	(1.812)
		5	4.805	4.719	**4.635**	4.555	5.306	37.343
			(0.376)	(0.322)	(0.307)	(0.458)	(0.471)	(1.790)

The bold values correspond to the smallest means among SPCRsvd-LADMM, SPCRsvd-ADMM, and SPCR

Table 6 Mean (standard deviation) of MSE for Case 6

σ	n	k	SPCRsvd-LADMM	SPCRsvd-ADMM	SPCR	SPLS	PLS	PCR
1	50	1	4.231	**1.796**	2.388	41.871	53.047	67.579
			(8.020)	(0.336)	(0.445)	(18.629)	(4.839)	(4.285)
		2	3.985	3.714	**1.455**	1.062	44.022	66.887
			(7.576)	(5.460)	(0.451)	(0.070)	(4.491)	(4.119)
	200	1	**1.132**	1.161	1.949	40.597	47.641	65.833
			(0.087)	(0.103)	(0.191)	(15.581)	(4.672)	(2.973)
		2	1.138	1.162	**1.058**	1.018	21.014	65.233
			(0.121)	(0.121)	(0.053)	(0.045)	(2.891)	(3.132)
2	50	1	12.009	**8.359**	9.313	45.062	56.410	70.652
			(8.929)	(1.560)	(1.723)	(18.692)	(4.828)	(4.361)
		2	12.183	12.967	**6.391**	4.250	47.681	69.978
			(8.387)	(6.057)	(1.666)	(0.280)	(4.499)	(4.233)
	200	1	**5.188**	5.241	7.784	42.878	50.619	68.876
			(0.369)	(0.393)	(0.783)	(16.130)	(4.790)	(3.053)
		2	5.193	5.261	**4.271**	4.073	24.591	68.280
			(0.400)	(0.355)	(0.216)	(0.182)	(2.938)	(3.193)

The bold values correspond to the smallest means among SPCRsvd-LADMM, SPCRsvd-ADMM, and SPCR

Table 7 Mean (standard deviation) of TPR, TNR, and MCC for Case 1

σ	n	k		SPCRsvd-LADMM	SPCRsvd-ADMM	SPCR	SPLS
1	50	1	TPR	0.980	**1**	0.810	0.930
				(0.272)	(0)	(0.394)	(0.174)
			TNR	0.461	0.553	0.387	**0.951**
				(0.339)	(0.271)	(0.325)	(0.130)
			MCC	0.327	0.470	0.188	**0.881**
				(0.295)	(0.232)	(0.144)	(0.184)
		5	TPR	0.970	**1**	0.980	**1**
				(0.171)	(0)	(0.140)	(0)
			TNR	0.512	0.532	0.273	**0.905**
				(0.288)	(0.249)	(0.201)	(0.220)
			MCC	0.412	0.460	0.238	**0.879**
				(0.253)	(0.222)	(0.137)	(0.235)
	200	1	TPR	0.870	**1**	0.430	**1**
				(0.337)	(0)	(0.497)	(0)
			TNR	0.480	0.700	0.711	**1**
				(0.381)	(0.311)	(0.357)	(0)
			MCC	0.306	0.637	0.123	**1**
				(0.347)	(0.305)	(0.183)	(0)
		5	TPR	0.960	**1**	0.850	**1**
				(0.196)	(0)	(0.358)	(0)
			TNR	0.577	0.630	0.441	**0.916**
				(0.333)	(0.298)	(0.300)	(0.152)
			MCC	0.474	0.557	0.255	**0.880**
				(0.323)	(0.292)	(0.176)	(0.196)
2	50	1	TPR	0.890	**1**	0.870	0.795
				(0.314)	(0)	(0.337)	(0.247)
			TNR	0.356	0.347	0.238	**0.942**
				(0.288)	(0.219)	(0.313)	(0.126)
			MCC	0.214	0.301	0.122	**0.775**
				(0.203)	(0.169)	(0.109)	(0.212)
		5	TPR	0.970	**1**	0.990	0.940
				(0.171)	(0)	(0.100)	(0.163)
			TNR	0.387	0.412	0.142	**0.878**
				(0.249)	(0.224)	(0.151)	(0.215)
			MCC	0.309	0.354	0.146	**0.795**
				(0.189)	(0.172)	(0.112)	(0.239)
	200	1	TPR	0.860	**1**	0.370	**1**
				(0.348)	(0)	(0.485)	(0)

Table 7 continued

σ	n	k		SPCRsvd-LADMM	SPCRsvd-ADMM	SPCR	SPLS
			TNR	0.412	0.501	0.703	**0.966**
				(0.345)	(0.269)	(0.399)	(0.106)
			MCC	0.227	0.417	0.070	**0.951**
				(0.261)	(0.219)	(0.122)	(0.136)
		5	TPR	0.950	**1**	0.920	**1**
				(0.219)	(0)	(0.272)	(0)
			TNR	0.487	0.545	0.276	**0.920**
				(0.304)	(0.262)	(0.264)	(0.150)
			MCC	0.375	0.469	0.190	**0.886**
				(0.255)	(0.223)	(0.138)	(0.194)

The bold values correspond to the largest means

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\left|\left\{j : \zeta_j^* = 0\right\}\right|} = \frac{1}{100}\sum_{k=1}^{100}\frac{\left|\left\{j : \hat{\zeta}_j^{(k)} = 0 \wedge \zeta_j^* = 0\right\}\right|}{\left|\left\{j : \zeta_j^* = 0\right\}\right|},$$

where $\mathrm{TP} = \sum_{k=1}^{100} |\{j : \hat{\zeta}_j^{(k)} \neq 0 \wedge \zeta_j^* \neq 0\}|/100$, $\mathrm{TN} = \sum_{k=1}^{100} |\{j : \hat{\zeta}_j^{(k)} = 0 \wedge \zeta_j^* = 0\}|/100$, ζ_j^* is the true j-th coefficient, $\hat{\zeta}_j^{(k)}$ is the estimated j-th coefficient for the k-th simulation, and $|\{*\}|$ is the number of elements included in set $\{*\}$. MCC is defined by

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}},$$

where $\mathrm{FP} = \sum_{k=1}^{100} |\{j : \hat{\zeta}_j^{(k)} \neq 0 \wedge \zeta_j^* = 0\}|/100$ and $\mathrm{FN} = \sum_{k=1}^{100} |\{j : \hat{\zeta}_j^{(k)} = 0 \wedge \zeta_j^* \neq 0\}|/100$.

Table 7 represents the means and standard deviations of TPR, TNR, and MCC for Case 1. Many methods provided higher TPR, whereas SPCR sometimes did not. SPLS provided the highest TNR and MCC among the methods in all situations. For all cases, these tendencies for TPR and TNR were essentially unchanged, while SPCRsvd-ADMM sometimes provided the highest ratios of MCC. The results from Cases 2 to 6 are shown in the supplementary material.

We also investigated the sensitivity of the tuning parameter w and the penalty parameters. Table 8 shows MSEs for SPCRsvd with $w = 1, 0.5, 0.01$. Note that we could not compute MSEs for $w = 0.01$ in Case 6. Table 9 shows MSEs for SPCRsvd with $\rho = 1.5, 0.5$, where $\rho = \rho_1 = \rho_2$ for SPCRsvd-LADMM and $\rho = \rho_1 = \rho_2 = \rho_3$ for SPCRsvd-ADMM. Note that the number of iterations of the simulation was 10 times and we set $n = 50$, $\sigma = 1$, and $k = 1$ in both settings. From the results, we observe that varying w has little influence on MSEs in SPCRsvd-ADMM, whereas it has a small influence in SPCRsvd-LADMM. For the penalty parameters, we observe that varying ρ has a small influence on MSEs (in particular, Case 6 seems to be affected

Table 8 Mean (standard deviation) of MSE for $w = 1, 0.5, 0.01$

Case	SPCRsvd-LADMM			SPCRsvd-ADMM		
	$w = 1$	$w = 0.5$	$w = 0.01$	$w = 1$	$w = 0.5$	$w = 0.01$
1	1.496	1.625	1.775	1.160	1.162	1.163
	(1.208)	(1.433)	(1.567)	(0.121)	(0.122)	(0.122)
2	1.926	1.933	1.962	1.220	1.221	1.229
	(7.338)	(7.337)	(7.337)	(0.159)	(0.161)	(0.158)
3	3.202	2.310	2.688	1.600	1.589	1.581
	(5.153)	(1.549)	(4.494)	(0.352)	(0.339)	(0.334)
4	2.913	2.311	2.404	2.298	2.301	2.313
	(1.067)	(0.631)	(0.761)	(0.603)	(0.590)	(0.582)
5	2.221	2.172	3.838	2.207	2.203	2.189
	(0.627)	(0.586)	(7.136)	(0.576)	(0.571)	(0.561)
6	3.285	4.906	–	1.858	1.820	–
	(3.187)	(4.027)	–	(0.483)	(0.346)	–

Table 9 Mean (standard deviation) of MSE for $\rho = 1.5, 0.5$, where we set $\rho = \rho_1 = \rho_2$ for SPCRsvd-LADMM and $\rho = \rho_1 = \rho_2 = \rho_3$ for SPCRsvd-ADMM

Case	SPCRsvd-LADMM		SPCRsvd-ADMM	
	$\rho = 1.5$	$\rho = 0.5$	$\rho = 1.5$	$\rho = 0.5$
1	1.624	1.475	1.163	1.180
	(1.357)	(1.145)	(0.131)	(0.116)
2	2.672	1.979	1.211	1.237
	(10.275)	(7.333)	(0.174)	(0.131)
3	1.529	1.620	1.574	1.612
	(0.331)	(0.319)	(0.354)	(0.325)
4	2.237	2.387	2.280	2.429
	(0.581)	(0.623)	(0.647)	(0.591)
5	2.395	2.296	2.172	2.298
	(3.261)	(0.527)	(0.610)	(0.564)
6	4.246	10.715	1.592	2.152
	(10.685)	(7.624)	(0.306)	(0.416)

by ρ.). However, we note that the influences do not essentially change the conclusions derived from Tables 1, 2, 3, 4, 5 and 6 in almost all cases. This means that MSEs of SPCRsvd may be relatively insensitive to w and ρ.

5.2 Real data analyses

We applied SPCRsvd to real datasets. Specifically, we applied it to eight real datasets (housing, communities, concrete, diabetes, parkinsons, triazines, winequality-red, and winequality-white), which are available from the UCI database (http://archive.ics.

Table 10 Sample size and number of covariates in real datasets

	Sample size	# of covariates
Housing	506	13
Communities	1993	101
Concrete	1030	8
Diabetes	442	10
Parkinsons	5875	19
Triazines	186	36
Winequality-red	1599	11
Winequality-white	4898	11

uci.edu/ml/index.html). The sample sizes and the numbers of covariates are listed in Table 10. If the sample size was larger than 1100, we randomly extracted 1,100 observations from the dataset. For each dataset, we randomly selected 100 observations as training data and used the remaining as test data to estimate MSEs. We standardized the covariates for each dataset. We applied two algorithms: SPCRsvd-LADMM and SPCRsvd-ADMM. The procedure was repeated 50 times.

We compared SPCRsvd with the four methods used in Sect. 5.1. The number of components was set as $k = 1$. The value of the tuning parameter w in SPCRsvd was set to 0.01, and then λ_V and λ_β were selected by five-fold cross-validation. The tuning parameters in the other methods were selected in similar manners to in Sect. 5.1.

Table 11 lists the means and standard deviations of MSEs. PLS and PCR were competitive but did not provide the smallest MSEs for any dataset. SPCR was slightly better than PLS and PCR. SPCRsvd-LADMM and SPCRsvd-ADMM provided smaller MSEs than the other methods in many cases. Although SPLS sometimes provided smaller MSEs than other methods, SPLS also had the worst MSEs in some cases. From the result, we may conclude that SPCRsvd-LADMM and SPCRsvd-ADMM are superior to the other methods in terms of giving smaller MSEs, which is consistent with the results in Sect. 5.1.

6 Conclusions

In this paper, we proposed SPCRsvd, a one-stage procedure for PCR with a loss function for regression loss and PCA loss of SVD. To obtain the estimates of the parameters in SPCRsvd, we developed two computational algorithms based on ADMM and LADMM. From our numerical studies, we observed that our one-stage method is competitive with or better than competing approaches.

A major limitation of SPCRsvd is the computational cost. Figure 2 shows common logarithm of the run-times for the simulation presented in Sect. 5.1. Note that the number of iterations of the simulation was 10 times and we set $n = 50$, $\sigma = 1$, and $k = 1$. In these results, we observe that SPCRsvd-ADMM was faster than SPCRsvd-LADMM, and that the SPCRsvd-based methods required more computation time than the other four methods in almost cases. This high computational cost causes

Table 11 Mean (standard deviation) of MSE for real datasets

	SPCRsvd-LADMM	SPCRsvd-ADMM	SPCR	SPLS	PLS	PCR
Housing	**28.51**	28.64	28.85	33.26	29.16	29.23
	(2.85)	(3.15)	(3.06)	(4.67)	(3.24)	(3.23)
Communities	3.467×10^{-2}	2.802×10^{-2}	3.465×10^{-2}	$\mathbf{2.500 \times 10^{-2}}$	7.368×10^{-2}	6.929×10^{-2}
	(0.403×10^{-2})	(0.627×10^{-2})	(0.220×10^{-2})	(0.133×10^{-2})	(6.501×10^{-2})	(4.623×10^{-2})
Concrete	124.4	**123.7**	124.7	142.0	125.0	125.0
	(13.7)	(13.8)	(14.3)	(11.1)	(14.5)	(14.4)
Diabetes	**3221**	3280	3280	3429	3281	3282
	(140)	(163)	(154)	(286)	(156)	(156)
Parkinsons	**113.6**	147.5	146.2	115.9	169.6	171.6
	(19.0)	(52.4)	(58.0)	(6.8)	(81.1)	(79.1)
Triazines	2.510×10^{-2}	2.516×10^{-2}	2.497×10^{-2}	$\mathbf{2.417 \times 10^{-2}}$	2.827×10^{-2}	2.798×10^{-2}
	(0.370×10^{-2})	(0.379×10^{-2})	(0.383×10^{-2})	(0.332×10^{-2})	(0.546×10^{-2})	(0.537×10^{-2})
Winequality-red	5.132×10^{-1}	4.875×10^{-1}	4.927×10^{-1}	$\mathbf{4.841 \times 10^{-1}}$	4.947×10^{-1}	4.947×10^{-1}
	(0.701×10^{-1})	(0.516×10^{-1})	(0.480×10^{-1})	(0.266×10^{-1})	(0.451×10^{-1})	(0.460×10^{-1})
Winequality-white	6.820×10^{-1}	$\mathbf{6.811 \times 10^{-1}}$	6.906×10^{-1}	7.065×10^{-1}	7.012×10^{-1}	7.010×10^{-1}
	(0.478×10^{-1})	(0.521×10^{-1})	(0.325×10^{-1})	y (0.362×10^{-1})	(0.467×10^{-1})	(0.472×10^{-1})

The bold values correspond to the smallest means

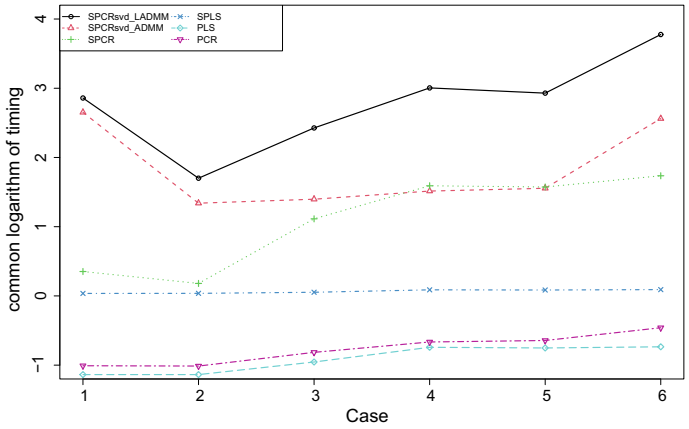

Fig. 2 Common logarithm of run-times (seconds) for the simulation in Sect. 5.1

some problems. For example, SPCRsvd provides relatively low TNR, based on Table 7. To address this issue, one could apply the adaptive lasso to the regularization term in SPCRsvd. However, owing to the computational cost, it may be difficult to perform SPCRsvd with the adaptive lasso because the adaptive lasso generally requires more computation time than lasso.

SPCRsvd cannot handle binary data for the explanatory variables. To perform PCA for binary data, Lee et al. (2010) introduced the logistic PCA with sparse regularization. It would be interesting to extend SPCRsvd in the context of the method in Lee et al. (2010). We leave them as future research.

Acknowledgements The author thanks the reviewers for their helpful comments and constructive suggestions. This work was supported by JSPS KAKENHI Grant Numbers JP19K11854 and JP20H02227, and MEXT KAKENHI Grant Numbers JP16H06429, JP16K21723, and JP16H06430.

Appendix

A Derivation of the updates in the ADMM algorithm

By simple calculations, we can easily obtain the solutions for β_0, Λ_1, Λ_2, $\boldsymbol{\lambda}_3$. Hence we show only the derivations for V_1, V, V_0, Z, $\boldsymbol{\beta}$, $\boldsymbol{\beta}_0$. For simplicity, we omit iteration index m.

Update of V_1.

$$V_1 := \arg\min_{V_1} \left\{ \frac{1}{n} \| \boldsymbol{y} - \beta_0 \mathbf{1}_n - X V_1 \boldsymbol{\beta} \|_2^2 + \frac{\rho_2}{2} \| V_1 - V_0 + \Lambda_2 \|_F^2 \right\}.$$

Set $\boldsymbol{y}^* = \boldsymbol{y} - \beta_0 \mathbf{1}_n$. The terms on the right-hand side are calculated by

$$\begin{aligned} \| \boldsymbol{y}^* - X V_1 \boldsymbol{\beta} \|_2^2 &= \boldsymbol{y}^{*\top} \boldsymbol{y}^* - 2\mathrm{tr}(\boldsymbol{\beta} \boldsymbol{y}^{*\top} X V_1) + \boldsymbol{\beta}^\top V_1^\top X^\top X V_1 \boldsymbol{\beta}, \\ \| V_1 - V_0 + \Lambda_2 \|_F^2 &= \mathrm{tr}(V_1^\top V_1) - 2\mathrm{tr}\{(V_0 - \Lambda_2)^\top V_1\} + \mathrm{tr}\{(V_0 - \Lambda_2)^\top (V_0 - \Lambda_2)\}. \end{aligned}$$

Using these, we obtain

$$\begin{aligned} \mathcal{F} :=& \frac{1}{n} \| \boldsymbol{y}^* - X V_1 \boldsymbol{\beta} \|_2^2 + \frac{\rho_2}{2} \| V_1 - V_0 + \Lambda_2 \|_F^2 \\ =& \frac{1}{n} \boldsymbol{\beta}^\top V_1^\top X^\top X V_1 \boldsymbol{\beta} - \frac{2}{n} \mathrm{tr}(\boldsymbol{\beta} \boldsymbol{y}^{*\top} X V_1) \\ &+ \frac{\rho_2}{2} \mathrm{tr}(V_1^\top V_1) - \rho_2 \mathrm{tr}\{(V_0 - \Lambda_2)^\top V_1\} + C, \end{aligned}$$

where C is a constant. Setting $\partial \mathcal{F} / \partial V_1 = \boldsymbol{O}$, we have

$$\frac{2}{n} X^\top X V_1 \boldsymbol{\beta} \boldsymbol{\beta}^\top - \frac{2}{n} X^\top \boldsymbol{y}^* \boldsymbol{\beta}^\top + \rho_2 V_1 - \rho_2 (V_0 - \Lambda_2) = \boldsymbol{O}.$$

This leads to the update for V_1.
Update of V.

$$V := \arg\min_{V} \left\{ \frac{w}{n} \| X - Z V^\top \|_F^2 + \frac{\rho_1}{2} \| V - V_0 + \Lambda_1 \|_F^2 \right\} \quad \text{subject to} \quad V^\top V = I_k.$$

The terms on the right-hand side are calculated by

$$\begin{aligned} \| X - Z V^\top \|_F^2 &= \mathrm{tr}(X^\top X) - 2\mathrm{tr}(V Z^\top X) + \mathrm{tr}(Z^\top Z), \\ \| V - V_0 + \Lambda_1 \|_F^2 &= -2\mathrm{tr}\{(V_0 - \Lambda_1)^\top V\} + \mathrm{tr}\{(V_0 - \Lambda_1)^\top (V_0 - \Lambda_1)\} + k. \end{aligned}$$

Adding the equality constraint $V^\top V = I_k$, we obatin

$$\begin{aligned} &\arg\min_{V} \left\{ \frac{w}{n} \| X - Z V^\top \|_F^2 + \frac{\rho_1}{2} \| V - V_0 + \Lambda_1 \|_F^2 \right\} \\ &= \arg\min_{V} \left\{ \left\| V - \left\{ \frac{w}{n} X^\top Z + \frac{\rho_1}{2} (V_0 - \Lambda_1) \right\} \right\|_F^2 \right\}. \end{aligned}$$

From the SVD $w X^\top Z / n + \rho_1 (V_0 - \Lambda_1) / 2 = P \Omega Q^\top$, we obtain the solution $V = P Q^\top$. This follows from the Procrustes rotation by Zou et al. (2006).

Update of V_0.

$$V_0 := \arg\min_{V_0} \left\{ \frac{\rho_1}{2} \| V - V_0 + \Lambda_1 \|_F^2 + \frac{\rho_2}{2} \| V_1 - V_0 + \Lambda_2 \|_F^2 + \lambda_V \| V_0 \|_1 \right\}. \tag{A.1}$$

A simple calculation shows that the first two terms on the right-hand side are given by

$$\frac{\rho_1 + \rho_2}{2} \left\| V_0 - \frac{1}{\rho_1 + \rho_2} \{ \rho_1 (V + \Lambda_1) + \rho_2 (V_1 + \Lambda_2) \} \right\|_F^2.$$

Formula (A.1) can be rewritten as

$$V_0 := \arg\min_{V_0} \left\{ \frac{1}{2} \left\| V_0 - \frac{1}{\rho_1 + \rho_2} \{ \rho_1 (V + \Lambda_1) + \rho_2 (V_1 + \Lambda_2) \} \right\|_F^2 + \frac{\lambda_V}{\rho_1 + \rho_2} \| V_0 \|_1 \right\}.$$

Thus we obtain the update of V_0.
Update of Z.

$$Z := \arg\min_{Z} \left\{ \frac{w}{n} \| X - Z V^\top \|_F^2 \right\}.$$

We have the solution $Z = XV$ from the first-order optimality condition.
Update of $\boldsymbol{\beta}$.

$$\boldsymbol{\beta} := \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{n} \| \boldsymbol{y} - \beta_0 \mathbf{1}_n - X V_1 \boldsymbol{\beta} \|_2^2 + \frac{\rho_2}{2} \| \boldsymbol{\beta} - \boldsymbol{\beta}_0 + \boldsymbol{\lambda} \|_2^2 \right\}.$$

The first-order optimality condition is

$$-\frac{2}{n} V_1^\top X^\top (\boldsymbol{y} - \beta_0 \mathbf{1}_n - X V_1 \boldsymbol{\beta}) + \rho_2 (\boldsymbol{\beta} - \boldsymbol{\beta}_0 + \boldsymbol{\lambda}) = \mathbf{0}.$$

This leads to the update of $\boldsymbol{\beta}$.
Update of $\boldsymbol{\beta}_0$.

$$\boldsymbol{\beta}_0 := \arg\min_{\boldsymbol{\beta}_0} \left\{ \frac{\rho_2}{2} \| \boldsymbol{\beta} - \boldsymbol{\beta}_0 + \boldsymbol{\lambda} \|_2^2 + \lambda_\beta \| \boldsymbol{\beta}_0 \|_1 \right\}.$$

It is clear that the update of $\boldsymbol{\beta}_0$ can be simply obtained by an element-wise soft-threshold operator.

B LADMM algorithm for SPCRsvd

The LADMM algorithm for SPCRsvd is as follows:

Step 1 Set the values of the tuning parameter w, regularization parameters λ_V, $\lambda_{\boldsymbol{\beta}}$, and penalty parameters ρ_1, ρ_2.
Step 2 Initialize all the parameters as $\beta_0^{(0)}$, $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{\beta}_0^{(0)}$, $Z^{(0)}$, $V^{(0)}$, $V_0^{(0)}$, $\Lambda^{(0)}$, $\boldsymbol{\lambda}^{(0)}$.
Step 3 For $m = 0, 1, 2, \ldots$, repeat Steps 4 to 10 until convergence.
Step 4 Update V as follows:

$$V^{(m+1)} = PQ^\top,$$

where P and Q are the matrices given by SVD

$$\frac{w}{n} X^\top Z^{(m)} + \frac{\rho_1}{2}\left(V_0^{(m)} + \Lambda^{(m)}\right) = P\Omega Q^\top.$$

Step 5 Update V_0 as follows:

$$v_{0ij}^{(m+1)} = \mathcal{S}\left(s_{ij}, \lambda_V / \left(\frac{2\nu + n\rho_1}{n}\right)\right), \quad i = 1, \ldots, p, \ j = 1, \ldots, k, \tag{B.1}$$

where $v_{0ij}^{(m)} = (V_0^{(m)})_{ij}$, ν is the maximum eigenvalue of $\boldsymbol{\beta}^{(m)}\boldsymbol{\beta}^{(m)\top} \otimes X^\top X$, and s_{ij} is the (i, j)-th element of the matrix

$$\frac{2n}{2\nu + n\rho_1}\left\{\frac{1}{n}\left(X^\top(\boldsymbol{y} - \beta_0^{(m)}\mathbf{1}_n)\boldsymbol{\beta}^{(m)\top} - X^\top X V_0^{(m)}\boldsymbol{\beta}^{(m)}\boldsymbol{\beta}^{(m)\top}\right) + \frac{\nu}{n}V_0^{(m)} - \frac{\rho_1}{2}(\Lambda^{(m)} - V^{(m+1)})\right)\right\}.$$

Step 6 Update Z by $Z^{(m+1)} = XV^{(m+1)}$.
Step 7 Update $\boldsymbol{\beta}$ as follows:

$$\boldsymbol{\beta}^{(m+1)} = \left(\frac{1}{n}V_0^{(m+1)\top}X^\top X V_0^{(m+1)} + \frac{\rho_2}{2}I_k\right)^{-1}\left\{\frac{1}{n}V_0^{(m+1)\top}X^\top(\boldsymbol{y} - \beta_0^{(m)}\mathbf{1}_n) + \frac{\rho_2}{2}(\boldsymbol{\beta}_0^{(m)} - \boldsymbol{\lambda}^{(m)})\right\}.$$

Step 8 Update $\boldsymbol{\beta}_0$ as follows:

$$\beta_{0j}^{(m+1)} = \mathcal{S}\left(\beta_j^{(m+1)} + \lambda_j^{(m)}, \frac{\lambda_\beta}{\rho_2}\right), \quad j = 1, \ldots, k,$$

where $\lambda_j^{(m)}$ and $\beta_j^{(m)}$ are the j-th element of the vector $\boldsymbol{\lambda}^{(m)}$ and $\boldsymbol{\beta}^{(m)}$, respectively.

Step 9 Update β_0 as follows:

$$\beta_0^{(m+1)} = \frac{1}{n}\mathbf{1}_n^\top(\boldsymbol{y} - XV_0^{(m+1)}\boldsymbol{\beta}^{(m+1)}).$$

Step 10 Update $\Lambda, \boldsymbol{\lambda}$ as follows:

$$\Lambda^{(m+1)} = \Lambda^{(m)} + V_0^{(m+1)} - V^{(m+1)},$$
$$\boldsymbol{\lambda}^{(m+1)} = \boldsymbol{\lambda}^{(m)} + \boldsymbol{\beta}^{(m+1)} - \boldsymbol{\beta}_0^{(m+1)}.$$

Next, we describe only the update of only V_0 because the derivations of other updates are the same as in Appendix A. As in Appendix A, we omit iteration index m.

We consider

$$V_0 := \arg\min_{V_0}\left\{\frac{1}{n}\|\boldsymbol{y} - \beta_0\mathbf{1}_n - XV_0\boldsymbol{\beta}\|_2^2 + \frac{\rho_1}{2}\|V_0 - V + \Lambda\|_F^2 + \lambda_V\|V_0\|_1\right\}. \tag{B.2}$$

Set $\boldsymbol{y}^* = \boldsymbol{y} - \beta_0\mathbf{1}_n$. By Taylor expansion, the term $\|\boldsymbol{y}^* - XV_0\boldsymbol{\beta}\|_2^2$ is approximated as

$$\begin{aligned}\|\boldsymbol{y}^* - XV_0\boldsymbol{\beta}\|_2^2 &= \boldsymbol{y}^{*\top}\boldsymbol{y}^* - 2\mathrm{tr}(\boldsymbol{\beta}\boldsymbol{y}^{*\top}XV_0) + \boldsymbol{\beta}^\top V_0^\top X^\top XV_0\boldsymbol{\beta} \\ &\approx \boldsymbol{y}^{*\top}\boldsymbol{y}^* - 2\mathrm{tr}(\boldsymbol{\beta}\boldsymbol{y}^{*\top}XV_0) + 2\mathrm{tr}(\boldsymbol{\beta}\boldsymbol{\beta}^\top\tilde{V}_0X^\top XV_0) + \nu\|V_0 - \tilde{V}_0\|_F^2,\end{aligned}$$

where $\tilde{V}_0$ is the current estimate of V_0 and ν is a constant. Following Li et al. (2014), we use the maximum eigenvalue of $\boldsymbol{\beta}\boldsymbol{\beta}^\top \otimes X^\top X$ as ν. Using the approximation, the problem (B.2) can be replaced with

$$V_0 := \arg\min_{V_0}\left\{\underbrace{-\frac{2}{n}\mathrm{tr}(\boldsymbol{\beta}\boldsymbol{y}^{*\top}XV_0) + \frac{2}{n}\mathrm{tr}(\boldsymbol{\beta}\boldsymbol{\beta}^\top\tilde{V}_0X^\top XV_0) + \frac{\nu}{n}\|V_0 - \tilde{V}_0\|_F^2 + \frac{\rho_1}{2}\|V_0 - V + \Lambda\|_F^2}_{\text{(A)}} + \lambda_V\|V_0\|_1\right\}.$$

Formula (A) is calculated as

$$\frac{2\nu + n\rho_1}{2n}\left\|V_0 - \frac{2n}{2\nu + n\rho_1}\left\{\frac{1}{n}(X^\top\boldsymbol{y}^*\boldsymbol{\beta}^\top - X^\top X\tilde{V}_0\boldsymbol{\beta}\boldsymbol{\beta}^\top) + \frac{\nu}{n}\tilde{V}_0 - \frac{\rho_1}{2}(\Lambda - V)\right\}\right\|_F^2.$$

This leads to the update of V_0 given in Formula (B.1).

References

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J et al (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends® Mach Learn 3(1):1–122

Bresler G, Park SM, Persu M (2018) Sparse PCA from sparse linear regression. In: Advances in Neural Information Processing Systems, pp. 10942–10952

Chang X, Yang H (2012) Combining two-parameter and principal component regression estimators. Stat Pap 53(3):549–562
Chen S, Ma S, Xue L, Zou H (2020) An alternating manifold proximal gradient method for sparse principal component analysis and sparse canonical correlation analysis. Inf J Optim 2(3):192–208
Choi J, Zou H, Oehlert G (2010) A penalized maximum likelihood approach to sparse factor analysis. Stat Interf 3(4):429–436
Chun H, Keleş S (2010) Sparse partial least squares regression for simultaneous dimension reduction and variable selection. J R Stat Soc Ser B 72(1):3–25
Danaher P, Wang P, Witten DM (2014) The joint graphical lasso for inverse covariance estimation across multiple classes. J R Stat Soc Ser B 76(2):373–397
d'Aspremont A, El Ghaoui L, Jordan MI, Lanckriet GR (2007) A direct formulation for sparse PCA using semidefinite programming. SIAM Rev 49(3):434–448
Dicker LH, Foster DP, Hsu D et al (2017) Kernel ridge versus principal component regression: minimax bounds and the qualification of regularization operators. Electron J Stat 11(1):1022–1047
Donoho DL, Johnstone JM (1994) Ideal spatial adaptation by wavelet shrinkage. Biometrika 81(3):425–455
Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32(2):407–499
Erichson NB, Zheng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY (2020) Sparse principal component analysis via variable projection. SIAM J Appl Math 80(2):977–1002
Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. J Am Stat Assoc 96(456):1348–1360
Febrero-Bande M, Galeano P, González-Manteiga W (2017) Functional principal component regression and functional partial least-squares regression: an overview and a comparative study. Int Stat Rev 85(1):61–83
Frank LE, Friedman JH (1993) A statistical view of some chemometrics regression tools. Technometrics 35(2):109–135
Friedman J, Hastie T, Höfling H, Tibshirani R (2007) Pathwise coordinate optimization. Ann Appl Stat 1(2):302–332
Hartnett M, Lightbody G, Irwin G (1998) Dynamic inferential estimation using principal components regression (PCR). Chemom Intell Lab Syst 40(2):215–224
Jennrich RI (2006) Rotation to simple loadings using component loss functions: the oblique case. Psychometrika 71(1):173–191
Jolliffe IT (1982) A note on the use of principal components in regression. Appl Stat 31(3):300–303
Jolliffe IT (2002) Principal component analysis. Wiley Online Library, New York
Kawano S, Fujisawa H, Takada T, Shiroishi T (2015) Sparse principal component regression with adaptive loading. Comput Stat Data Anal 89:192–203
Kawano S, Fujisawa H, Takada T, Shiroishi T (2018) Sparse principal component regression for generalized linear models. Comput Stat Data Anal 124:180–196
Lee S, Huang JZ, Hu J (2010) Sparse logistic principal components analysis for binary data. Ann Appl Stat 4(3):1579–1601
Li X, Mo L, Yuan X, Zhang J (2014) Linearized alternating direction method of multipliers for sparse group and fused lasso models. Comput Stat Data Anal 79:203–221
Ma S, Huang J (2017) A concave pairwise fusion approach to subgroup analysis. J Am Stat Assoc 112(517):410–423
Massy WF (1965) Principal components regression in exploratory statistical research. J Am Stat Assoc 60(309):234–256
Matthews BW (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure 405(2):442–451
Pearson K (1901) On lines and planes of closest fit to systems of point in space. Philos Mag 2:559–572
Price BS, Geyer CJ, Rothman AJ (2019) Automatic response category combination in multinomial logistic regression. J Comput Graph Stat 28(3):758–766
R Core Team (2020) R : A language and environment for statistical computing. R foundation for statistical computing, Vienna, Austria (2020). https://www.R-project.org/
Reiss PT, Ogden RT (2007) Functional principal component regression and functional partial least squares. J Am Stat Assoc 102(479):984–996
Rosipal R, Girolami M, Trejo LJ, Cichocki A (2001) Kernel PCA for feature extraction and de-noising in nonlinear regression. Neural Comput Appl 10(3):231–243

Shen H, Huang JZ (2008) Sparse principal component analysis via regularized low rank matrix approximation. J Multivar Anal 99(6):1015–1034
Tan K, London P, Mohan K, Lee S, Fazel M, Witten D (2014) Learning graphical models with hubs. J Mach Learn Res 15:3297–3331
Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc Ser B 58(1):267–288
Vu VQ, Cho J, Lei J, Rohe K (2013) Fantope projection and selection: A near-optimal convex relaxation of sparse PCA. In: Advances in neural information processing systems, pp. 2670–2678
Wang B, Zhang Y, Sun WW, Fang Y (2018) Sparse convex clustering. J Comput Graph Stat 27(2):393–403
Wang K, Abbott D (2008) A principal components regression approach to multilocus genetic association studies. Genet Epidemiol 32(2):108–118
Wang X, Yuan X (2012) The linearized alternating direction method for dantzig selector. SIAM J Sci Comput 34(5):A2792–A2811
Witten DM, Tibshirani R, Hastie T (2009) A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics 10(3):515–534
Wold H (1975) Soft modeling by latent variables: the nonlinear iterative partial least squares approach. Perspectives in probability and statistics, papers in honour of MS Bartlett pp. 520–540
Wu TT, Lange K (2008) Coordinate descent algorithms for lasso penalized regression. Ann Appl Stat 2(1):224–244
Yan X, Bien J (2020) Rare feature selection in high dimensions. J Am Stat Assoc (accepted) pp 1–30
Ye GB, Xie X (2011) Split bregman method for large scale fused lasso. Comput Stat Data Anal 55(4):1552–1569
Zhang CH et al (2010) Nearly unbiased variable selection under minimax concave penalty. Ann Stat 38(2):894–942
Zou H (2006) The adaptive lasso and its oracle properties. J Am Stat Assoc 101(476):1418–1429
Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B 67(2):301–320
Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. J Comput Graph Stat 15(2):265–286
Zou H, Xue L (2018) A selective overview of sparse principal component analysis. Proce IEEE 106(8):1311–1320