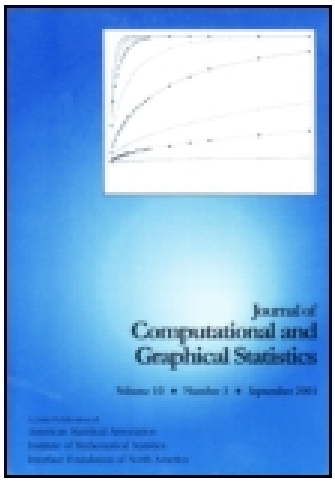

Journal of Computational and Graphical Statistics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ucgs20

Sparse Regression by Projection and Sparse Discriminant Analysis

Xin Qi[a], Ruiyan Luo[a], Raymond J. Carroll[b] & Hongyu Zhao[c]

[a] Department of Mathematics and Statistics, Georgia State University, 30 Pryor Street, Atlanta, GA 30303-3083 and

[b] Department of Statistics, Texas A&M University, 3143 TAMU, College Station, TX 77843-3143

[c] Department of Biostatistics, Yale School of Public Health, 60 College Street, New Haven CT 06520

Accepted author version posted online: 16 May 2014.

To cite this article: Xin Qi, Ruiyan Luo, Raymond J. Carroll & Hongyu Zhao (2014): Sparse Regression by Projection and Sparse Discriminant Analysis, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2014.907094

To link to this article: http://dx.doi.org/10.1080/10618600.2014.907094

Sparse Regression by Projection and Sparse Discriminant Analysis

Xin Qi , Ruiyan Luo
Department of Mathematics and Statistics, Georgia State University, 30 Pryor Street, Atlanta, GA 30303-3083
xqi3@gsu.edu and rluo@gsu.edu

Raymond J. Carroll
Department of Statistics, Texas A&M University, 3143 TAMU, College Station, TX 77843-3143
carroll@stat.tamu.edu

Hongyu Zhao
Department of Biostatistics, Yale School of Public Health, 60 College Street, New Haven CT 06520
hongyu.zhao@yale.edu

Abstract

Recent years have seen active developments of various penalized regression methods, such as LASSO and elastic net, to analyze high dimensional data. In these approaches, the direction and length of the regression coefficients are determined simultaneously. Due to the introduction of penalties, the length of the estimates can be far from being optimal for accurate predictions. We introduce a new framework, regression by projection, and its sparse version to analyze high dimensional data. The unique nature of this framework is that the directions of the regression coefficients are inferred first, and the lengths and the tuning parameters are determined by a cross validation procedure to achieve the largest prediction accuracy. We provide a theoretical result for simultaneous model selection consistency and parameter estimation consistency of our method in high dimension. This new framework is then generalized such that it can be applied to principal components analysis, partial least squares and canonical correlation analysis. We also adapt this framework for discriminant analysis. Compared to the existing methods, where there is relatively little control of the dependency among the sparse components, our method can control the relationships among the components. We present efficient algorithms and related theory for solving the sparse regression by projection problem. Based on extensive simulations and real data analysis, we demonstrate that our method achieves good predictive performance and variable selection in the regression setting, and the ability to control relationships between the sparse components leads to more accurate classification. In supplemental materials available online, the details of the algorithms and theoretical proofs, and R codes for all simulation studies are provided.

Some Key Words: Discriminant analysis; Sparse discriminant analysis; Sparse regression by projection; Zero within-class and between-class correlations.

Short title: Sparse Regression by Projection

1 Introduction

It is well recognized that classical multivariate statistical methods have difficulty in dealing with high-dimensional data. For example, ordinary least squares (OLS) has poor prediction accuracy as well as problems of interpretation for high dimensional data. Various penalization techniques have been proposed to improve OLS, such as ridge regression (Hoerl and Kennard, 1970), LASSO (Tibshirani, 1996), elastic net (Zou and Hastie, 2005), supervised principal components (Bair *et al.*, 2006), sparse partial least squares regression (Chun and Keles, 2010), smoothly clipped absolute deviation (SCAD) (Fan and Li, 2001), minimax concave penalty (MCP) (Zhang, 2010), and many others. Similar to regression, many standard dimension reduction methods, such as principal components analysis (PCA), partial least squares (PLS), canonical correlation analysis (CCA), and Fisher's discriminant analysis for classification also perform poorly in prediction and feature selection, and even fail, in high-dimensional settings. A common feature of these dimension reduction methods is that they all solve eigenvalue or generalized eigenvalue problems, where the eigenvectors correspond to different components and there is no correlation or within-class and between-class correlation among different components. Similar to regression, regularization techniques have been proposed to augment these methods to analyze high-dimensional data. However, one major limitation of these methods is that they are not able to control the relationships among the components, due to inherent limitations in these algorithms. As a result, the components thus constructed may be strongly correlated, which can affect both the interpretation of these components and prediction accuracy.

In this paper, we introduce a new framework, *regression by projection*, which is equivalent to OLS in the classic regression problem. Based on this new framework, we introduce a sparse version for high dimensional data. The unique feature of our new sparse regression approach is that the direction of the estimated coefficient vector is determined first and then its lengths and the tuning parameters are determined by a cross validation procedure to achieve the largest prediction accuracy. The LASSO and elastic net determine the direction and the length of the estimate simultaneously. We consider the well-known elastic net regression method, which solves

the penalized least squares problem

$$\max_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \left[\{(1-\alpha)/2\} \|\boldsymbol{\beta}\|_2^2 + \alpha \|\boldsymbol{\beta}\|_1 \right], \tag{1.1}$$

where $\lambda \geq 0$ and $0 \leq \alpha \leq 1$ are tuning parameters. Here we use the parameterization adopted in the R package "glmnet" for the elastic net and denote the estimate as $\widehat{\boldsymbol{\beta}}_{EN}$. Typically $\mathbf{X}\widehat{\boldsymbol{\beta}}_{EN}$ is not the projection of $\mathbf{y}$ along the direction of $\mathbf{X}\widehat{\boldsymbol{\beta}}_{EN}$. When $\alpha \neq 0$ and λ is large or $\lambda \neq 0$ and α is large, $\mathbf{X}\widehat{\boldsymbol{\beta}}_{EN}$ is far from the projection of $\mathbf{y}$ along the direction of $\mathbf{X}\widehat{\boldsymbol{\beta}}_{EN}$, which may lead to large bias in the estimate. We develop a regularized version, *sparse regression by projection*, under the new framework, to address this limitation in LASSO and elastic net. It has been proposed (for example, Cho and Fryzlewicz (2012)) that variable selection methods are used to identify the set of relevant variables,the final model is constructed using only the selected variables and the OLS. Hence, all the coefficients of the selected variables are estimated in a separate step other than the variable selection step. The performance of the final model will heavily rely on variable selection. Our method is different from this approach, in that, the subset of variables and the direction of the estimate are identified simultaneously, and only the length of the estimate (a scaling factor) is determined in a separate step. Zhao and Yu (2006) proved the model consistency for Lasso. Under the similar setting, we prove the simultaneous consistency of model selection and parameter estimation of our method.

In addition to regression, sparse regression by projection is also generalized to develop sparse versions of PCA, PLS, CCA, and discriminant analysis. In addition to achieving sparse components, the relationships (i.e. dependency) among the components can be controlled in our method, a distinct advantage over existing methods where the sparse components can be highly correlated. For example, in sparse PCA, CCA and PLS, we can control the components to be either orthogonal or uncorrelated; while in sparse discriminant analysis, we can achieve either zero within-class or zero between-class correlations, or both. In this paper, we focus on sparse discriminant analysis. We show that the control of within-class and between-class correlations among the sparse components can improve prediction accuracy in some situations. We develop an efficient algorithm and related theory for solving the sparse regression by projection problems. For the regression setting,

numerical examples show that the new algorithm is faster than LARS and comparable to the Coordinate Descent algorithm. When the new framework is compared with existing regression methods for high-dimensional data through extensive simulations and application to empirical data sets, the results show that our methods achieve good predictive performance and variable selection. In the classification setting, our results suggest that the control of relationships between the components leads to more accurate classification.

The rest of the paper is organized as follows. In Section 2, we introduce regression by projection and its connection to discriminant analysis. Section 3 contains the development of sparse regression by projection. Section 4 discusses our discriminant analysis method. Simulation studies and case studies are presented in Sections 5 and 6, respectively. Section 7 is a short discussion. Algorithmic details, the related theorems and all proofs are provided in the online supplementary materials.

2 Regression by projection

2.1 Regression by projection

Let us consider the classical regression. Suppose that the data set has n observations with p predictors. Let $\mathbf{y} = (y_1, \cdots, y_n)^{\mathrm{T}}$ be the response vector and let $\mathbf{X}$ be the $n \times p$ design matrix. In OLS, the response is fitted by the linear function $(\mathbf{X} - \mu_X)\widehat{\boldsymbol{\beta}}$, where the coefficient vector $\widehat{\boldsymbol{\beta}}$ minimizes $\|(\mathbf{y} - \mu_y) - (\mathbf{X} - \mu_X)\boldsymbol{\beta}\|_2^2$, μ_y and μ_X are the means of $\mathbf{y}$ and $\mathbf{X}$, respectively, and $\|\cdot\|_2$ is the L_2 norm. Equivalently, the OLS estimate $\widehat{\boldsymbol{\beta}}$ can be obtained by the following two-step method. We first obtain the direction of $\widehat{\boldsymbol{\beta}}$ by solving

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^p} (\mathbf{y} - \mu_y)^{\mathrm{T}}(\mathbf{X} - \mu_X)\boldsymbol{\alpha}, \quad \text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{X} - \mu_X)^{\mathrm{T}}(\mathbf{X} - \mu_X)\boldsymbol{\alpha} \leq 1, \tag{2.1}$$

where we assume that $(\mathbf{X} - \mu_X)^{\mathrm{T}}(\mathbf{X} - \mu_X)$ is full rank for the moment. Let $\widetilde{\boldsymbol{\alpha}}$ be a solution to (2.1). Then $\widehat{\boldsymbol{\beta}} = \left[(\mathbf{y} - \mu_y)^{\mathrm{T}}(\mathbf{X} - \mu_X)\widetilde{\boldsymbol{\alpha}}/\widetilde{\boldsymbol{\alpha}}^{\mathrm{T}}(\mathbf{X} - \mu_X)^{\mathrm{T}}(\mathbf{X} - \mu_X)\widetilde{\boldsymbol{\alpha}}\right]\widetilde{\boldsymbol{\alpha}}$, i.e., $(\mathbf{X} - \mu_X)\widehat{\boldsymbol{\beta}}$ is the orthogonal projection of $(\mathbf{y} - \mu_y)$ along the direction of $(\mathbf{X} - \mu_X)\widetilde{\boldsymbol{\alpha}}$. Because solving (2.1) is equivalent to

finding the linear combination of the columns of $(\mathbf{X} - \mu_X)$ which has the largest projection in the direction of $(\mathbf{y} - \mu_y)$ among all the linear combinations satisfying the constraint in (2.1), we call this method *regression by projection*. We will generalize this concept in the following. Without loss of generality, we will assume that the response is centered and the column means of $\mathbf{X}$ equal zero, that is, $\mu_X = \mathbf{0}$ and $\mu_y = \mathbf{0}$, except in Section 3.2 where the whole data set will be partitioned into different subsets and the means of responses and predictors for different subsets will not be the same and cannot be equal to zero simultaneously. When $\mu_X = \mathbf{0}$ and $\mu_y = \mathbf{0}$, (2.1) becomes

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^p} \mathbf{y}^{\mathrm{T}} \mathbf{X} \boldsymbol{\alpha}, \quad \text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} \boldsymbol{\alpha} \leq 1, \tag{2.2}$$

2.2 Connection between regression by projection and the discriminant analysis

Several important statistical methods, including PCA, PLS, and CCA for dimension reduction and Fisher's discriminant analysis for classification, have a common feature: they all solve eigenvalue or generalized eigenvalue problems and the eigenvectors correspond to different components. There are connections between these methods and the regression by projection discussed above. In this paper, we will focus on discriminant analysis although it is straightforward to apply the idea to other methods. As a classification method, Fisher's discriminant analysis projects the original variables to a subspace with dimension less than the number of classes such that the between-class variance is maximized relative to the within-class variance. Hence, this method finds the projection subspace such that different classes can be separated as much as possible. Suppose that the data set has n observations with p predictors and let $\mathbf{X}$ be the $n \times p$ data matrix. Let K be the number of classes and $\mathbf{Y}$ be an $n \times K$ matrix with $\mathbf{Y}_{ik}$ equal to 1 if the i^{th} observation is in the k^{th} class and equal to 0 otherwise. Then the between-class and the within-class covariance matrices can be written as $\boldsymbol{\Sigma}_b = n^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}(\mathbf{Y}^{\mathrm{T}}\mathbf{Y})^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{X}$ and $\boldsymbol{\Sigma}_w = n^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X} - \boldsymbol{\Sigma}_b$, respectively. Fisher's discriminant analysis method sequentially finds linear combinations $\mathbf{X}\boldsymbol{\alpha}_1, \cdots, \mathbf{X}\boldsymbol{\alpha}_{K-1}$, that span the projection

space or the score space, by solving

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^p} \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_b\boldsymbol{\alpha}, \quad \text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha} \le 1, \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha}_j = 0, \quad 1 \le j < k, \tag{2.3}$$

where $1 \le k \le K-1$. Assume that $\boldsymbol{\Sigma}_w$ is full rank for the moment, since (2.3) may not have a solution otherwise. Then each observation is assigned to the class whose class mean is closest to the corresponding point in the projection space. With this approach, $\alpha_1, \cdots, \alpha_{K-1}$ satisfy both $\alpha_j\boldsymbol{\Sigma}_b\alpha_k = 0$ and $\alpha_j\boldsymbol{\Sigma}_w\alpha_k = 0$, for all $j \ne k$. Consequently, the within-class covariance matrix $(\alpha_1, \cdots, \alpha_{K-1})^{\mathrm{T}}\boldsymbol{\Sigma}_w(\alpha_1, \cdots, \alpha_{K-1})$ and between-class covariance matrix $(\alpha_1, \cdots, \alpha_{K-1})^{\mathrm{T}}\boldsymbol{\Sigma}_b(\alpha_1, \cdots, \alpha_{K-1})$ among the components, $\mathbf{X}\boldsymbol{\alpha}_1, \cdots, \mathbf{X}\boldsymbol{\alpha}_{K-1}$, are both diagonal matrices. In other words, there are no within-class and between-class correlations among these components. If there are no differences among the magnitudes of the diagonal elements in the between-class covariance matrix, then in the projection space, the class means will not be concentrated along particular directions. On the other hand, by (2.3), the within-class covariance matrix $(\alpha_1, \cdots, \alpha_{K-1})^{\mathrm{T}}\boldsymbol{\Sigma}_w(\alpha_1, \cdots, \alpha_{K-1})$ is equal to the identity matrix, which implies that in the projection space, the data points are isotropically distributed about the class means. Hence, the conditions, $\alpha_j\boldsymbol{\Sigma}_b\alpha_k = 0$ and $\alpha_j\boldsymbol{\Sigma}_w\alpha_k = 0$, for all $j \ne k$, make it easier to separate the classes, which is illustrated by Figures 2 and 3 in our simulation studies. Hence, we want to keep these properties in our new method for sparse LDA.

Equation (2.3) is a generalized eigenvalue problem. Consider the well-known power method for solving generalized eigenvalue problems. For any $1 \le k \le K-1$, an initial vector $\boldsymbol{\alpha}^{(0)}$ with $\boldsymbol{\Sigma}_b\boldsymbol{\alpha}^{(0)} \ne 0$ is selected and a sequence, $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \cdots$, is iteratively calculated, where $\boldsymbol{\alpha}^{(i)}$ solves

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^p} (\boldsymbol{\alpha}^{(i-1)})^{\mathrm{T}}\boldsymbol{\Sigma}_b\boldsymbol{\alpha}, \quad \text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha} \le 1, \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha}_j = 0, \quad 1 \le j < k. \tag{2.4}$$

Then the sequence converges to $\boldsymbol{\alpha}_k$, where we assume that the eigenvalues involved have multiplicity one. It can be seen that both (2.2) (the key step of regression by projection) and (2.4) are special cases of the problem

$$\max_{\mathbf{u}} \mathbf{c}^{\mathrm{T}}\mathbf{u}, \quad \text{subject to} \quad \mathbf{u}^{\mathrm{T}}\mathbf{C}\mathbf{u} \le 1, \quad \mathbf{D}\mathbf{u} = 0, \tag{2.5}$$

where $\mathbf{c}$ is a nonzero vector, $\mathbf{C}$ is a nonnegative definite symmetric matrix and $\mathbf{D}$ is a matrix. In

fact, $\mathbf{u} = \boldsymbol{\alpha}$, $\mathbf{c} = \mathbf{y}^{\mathrm{T}}\mathbf{X}$, $\mathbf{C} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$ and $\mathbf{D} = \mathbf{0}$ in (2.2). For (2.4), $\mathbf{u} = \boldsymbol{\alpha}$, $\mathbf{c} = \boldsymbol{\Sigma}_b\boldsymbol{\alpha}^{(i-1)}$, $\mathbf{C} = \boldsymbol{\Sigma}_w$ and $\mathbf{D} = (\boldsymbol{\Sigma}_w\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\Sigma}_w\boldsymbol{\alpha}_{k-1})^{\mathrm{T}}$. Without loss of generality, we assume that $\mathbf{c} \neq \mathbf{0}, \mathbf{u} \in \mathbb{R}^p$, $\mathbf{C}$ is a $p \times p$ matrix with rank $n \leq p$ and $\mathbf{D}$ is a $d_1 \times p$ matrix. We call (2.5) *regression by projection with linear constraints*. Hence, solving discriminant analysis problems is equivalent to iteratively solving the problem of regression by projection with linear constraints.

3 Sparse regression by projection

3.1 Sparse regression by projection

In this section, we develop a regularized regression method based on regression by projection. In our approach, the direction of the estimator is determined by a regularized optimization problem and its length will be estimated by projecting the response variable along the inferred direction. Consider a sparse version of (2.2). The direction $\widetilde{\boldsymbol{\alpha}}$ of our estimate solves

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^p} \mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\alpha}, \quad \text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\alpha} + \tau||\boldsymbol{\alpha}||_\lambda^2 \leq 1, \tag{3.1}$$

where $||\boldsymbol{\alpha}||_\lambda^2 = (1-\lambda)||\boldsymbol{\alpha}||_2^2 + \lambda||\boldsymbol{\alpha}||_1^2$, and both $\tau \geq 0$ and $0 \leq \lambda \leq 1$ are tuning parameters. The introduction of $||\boldsymbol{\alpha}||_2^2$ in the constraint aims to overcome potential multicollinearity problems. When $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is not full rank, (for example, when $n < p$), the solution to (2.2) does not exist. The l_1 term in the constraint of (3.1) leads to sparse solutions. We use $||\boldsymbol{\alpha}||_1^2$ instead of $||\boldsymbol{\alpha}||_1$ as in the elastic net so that the solution to

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^p} \mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\alpha}, \quad \text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\alpha} + \tau||\boldsymbol{\alpha}||_\lambda^2 \leq t,$$

where t is any positive number, differs from the solution to (3.1) only by a multiplicative constant and thus the two solution vectors have the same directions. Hence, the sparsity penalty is actually imposed on the direction of the coefficient vector. Both λ and τ can control the sparsity of $\widetilde{\boldsymbol{\alpha}}$. When $\tau > 0$ and $0 \leq \lambda < 1$, the feasible region is a strictly convex set, and hence the solution to (3.1) is unique. Figure 1 provides some insight into (3.1) for a two-dimensional case. When λ and τ are

large, the second coordinate of $\widetilde{\boldsymbol{\alpha}}$ is equal to zero. Hence, it can be anticipated that, in general, $\widetilde{\boldsymbol{\alpha}}$ is sparse for large values of τ and λ.

A penalized version of (3.1). Although we formulate our method as a constrained optimization problem (3.1), it has the following penalized version (3.2). The solutions to (3.1) and (3.2) differ only by a scaling factor. However, the essential difference is that in (3.1), we do not determine the length, whereas both the direction and length are determined in (3.2).

Theorem 3.1. *The optimization problem* (3.1) *has the following penalized version,*

$$\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \tau \left[(1-\lambda)||\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1^2 \right], \tag{3.2}$$

where τ and λ are the same parameters as in (3.1). *The solutions to* (3.1) *and* (3.2) *differ only by a scaling factor. Specifically, let $\boldsymbol{\alpha}^*$ be the solution to* (3.1), *then the solution to* (3.2) *is*

$$\boldsymbol{\beta}^* = \frac{\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\alpha}^*}{||\mathbf{X}\boldsymbol{\alpha}^*||_2^2 + \tau ||\boldsymbol{\alpha}^*||_\lambda^2} \boldsymbol{\alpha}^*. \tag{3.3}$$

The major difference between (3.2) and the elastic-net problem is that the squared l_1 norm is used in (3.2) instead of the l_1 norm itself. This difference makes (3.2) (and (3.1)) enjoy scale invariant properties which are not possessed by the elastic-net. Specifically,

(a). If $\boldsymbol{\beta}^*$ is the solution to (3.2), then $c\boldsymbol{\beta}^*$ is the solution to (3.2) with $\mathbf{y}$ replaced by $c\mathbf{y}$, where c is any positive scaling constant.

(b). If $\boldsymbol{\beta}^*$ is the solution to (3.2), then $\boldsymbol{\beta}^*/c$ is the solution to

$$\max_{\boldsymbol{\beta}} ||\mathbf{y} - c\mathbf{X}\boldsymbol{\beta}||_2^2 + c\tau \left[(1-\lambda)||\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1^2 \right], \tag{3.4}$$

where c is any positive scaling constant.

Hence, scaling $\mathbf{y}$ does not affect the direction of the estimate of the coefficients. When we scale $\mathbf{X}$, we just need to make τ scaled by the same amount, then the direction of the estimate of the coefficients is unchanged. However, the elastic-net does not have this property. When $\mathbf{y}$ is scaled, the direction of the estimate of the coefficients is changed.

We use (3.1) instead of the penalized version (3.2) for two reasons. First, we actually consider a much more general optimization problem which is the sparse version of the (2.5). As we have

seen in Section 2.2, many multivariate methods can be connected to (2.5). Hence, based on our algorithms, we can propose sparse versions for various multivariate methods including the sparse regression problems (3.1) and (3.2). Second, it can be seen from (3.3) that, when the tuning parameter τ is large, the length of the solution $\boldsymbol{\beta}^*$ to (3.2) is quite small and can be far away from the optimal one, which can lead to large prediction errors. Instead, in (3.2), we only determine the direction of the estimate and then the length is chosen to minimize the prediction errors. In addition, the tuning parameters τ and λ need to be chosen to minimize the prediction errors. Hence, we design a cross-validation procedure to choose the tuning parameters and the length of the estimate simultaneously. The details are described in Section 3.2.

By Theorem 3.1, in the special case of $\lambda = 0$, given the tuning parameter τ, the solution of (3.1) has the same direction as ridge regression but the lengths are usually different. Moreover, since we have different cross-validation procedure from ridge regression, the selected τ can be quite different.

3.2 Choices of tuning parameters and determination of the length of the estimate

We use cross-validation (CV) to choose the tuning parameters. In our method, the length of the estimate is not determined by the optimization problem itself. Instead, it is viewed as a special tuning parameter and will be chosen to maximize prediction accuracy. To measure the prediction accuracy of the models corresponding to different values of λ and τ, we must consider the effect of the length of the estimate. Given a pair of λ and τ, we choose the length to minimize the prediction mean squared error and hence to choose λ and τ. Our cross-validation procedure is different from that of the Lasso and the elastic-net. Roughly speaking, to choose the tuning parameters $\tau \geq 0$ and $0 \leq \lambda < 1$, we randomly split the entire data set $(\mathbf{y}, \mathbf{X})$ into a "calculation set", $(\mathbf{y}_{\text{cal}}, \mathbf{X}_{\text{cal}})$, and an "evaluation set", $(\mathbf{y}_{\text{eval}}, \mathbf{X}_{\text{eval}})$. For each pair (τ, λ), we first determine the direction $\widetilde{\boldsymbol{\alpha}}(\tau, \lambda)$ of $\boldsymbol{\beta}$ using the calculation set. Then we project the centered $\mathbf{y}_{\text{eval}}$ onto the direction of the centered $\mathbf{X}_{\text{eval}}\widetilde{\boldsymbol{\alpha}}(\tau, \lambda)$, and obtain the residual vector, which is the difference between the centered $\mathbf{y}_{\text{valid}}$ and

the projection. Then we compute the mean squared error. Another important difference between our CV procedure and the usual CV procedure is that we do not split the entire data set into 10 subsets as in standard 10-fold CV procedure where, in each repeat, one subset is selected as the validation set and all the other observations as the training set. The main reason is that in our approach, the length is not determined by the training set. Hence, in order to more accurately estimate the prediction errors in the validation set, we decrease the size of the training set and increase the size of the validation set.

Specifically, we repeat the following procedure 10 times. In the i^{th} repeat, $1 \leq i \leq 10$, we randomly split the whole data set into a calculation set and an evaluation set, where the evaluation set has one third of all the observations. Let $\widehat{\boldsymbol{\mu}}_X^{(\text{cal})}$ denote the matrix with the same size as $\mathbf{X}_{\text{cal}}$ and the values in each column equal to the mean of the corresponding column of $\mathbf{X}_{\text{cal}}$. Let $\widehat{\boldsymbol{\mu}}_y^{(\text{cal})}$ be the vector with each value equal to the mean of $\mathbf{y}_{\text{cal}}$. For any pair (τ, λ), the direction, $\widetilde{\boldsymbol{\alpha}}(\tau, \lambda)$, is the solution to

$$\begin{aligned} &\max_{\boldsymbol{\alpha} \in \mathbb{R}^p} \quad (\mathbf{y}_{\text{cal}} - \widehat{\boldsymbol{\mu}}_y^{(\text{cal})})^{\mathrm{T}} (\mathbf{X}_{\text{cal}} - \widehat{\boldsymbol{\mu}}_X^{(\text{cal})}) \boldsymbol{\alpha}, \\ &\text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}} (\mathbf{X}_{\text{cal}} - \widehat{\boldsymbol{\mu}}_X^{(\text{cal})})^{\mathrm{T}} (\mathbf{X}_{\text{cal}} - \widehat{\boldsymbol{\mu}}_X^{(\text{cal})}) \boldsymbol{\alpha} + \tau \|\boldsymbol{\alpha}\|_\lambda^2 \leq 1. \end{aligned} \tag{3.5}$$

Once the direction is determined, we project $\mathbf{y}_{\text{eval}} - \widehat{\boldsymbol{\mu}}_y^{(\text{cal})}$ along the direction of $(\mathbf{X}_{\text{eval}} - \widehat{\boldsymbol{\mu}}_X^{(\text{cal})}) \widetilde{\boldsymbol{\alpha}}(\tau, \lambda)$. The projection is $(\mathbf{X}_{\text{eval}} - \widehat{\boldsymbol{\mu}}_X^{(\text{cal})}) \widehat{\boldsymbol{\beta}}(\tau, \lambda)$, where

$$\widehat{\boldsymbol{\beta}}(\tau, \lambda) = \frac{(\mathbf{y}_{\text{eval}} - \widehat{\boldsymbol{\mu}}_y^{(\text{cal})})^{\mathrm{T}} (\mathbf{X}_{\text{eval}} - \widehat{\boldsymbol{\mu}}_X^{(\text{cal})}) \widetilde{\boldsymbol{\alpha}}(\tau, \lambda)}{\widetilde{\boldsymbol{\alpha}}^{\mathrm{T}}(\tau, \lambda) (\mathbf{X}_{\text{eval}} - \widehat{\boldsymbol{\mu}}_X^{(\text{cal})})^{\mathrm{T}} (\mathbf{X}_{\text{eval}} - \widehat{\boldsymbol{\mu}}_X^{(\text{cal})}) \widetilde{\boldsymbol{\alpha}}(\tau, \lambda)} \widetilde{\boldsymbol{\alpha}}(\tau, \lambda). \tag{3.6}$$

Then we calculate the mean squared error,

$$MSE(\tau, \lambda, i) = \|\mathbf{y}_{\text{eval}} - \widehat{\boldsymbol{\mu}}_y^{(\text{cal})} - (\mathbf{X}_{\text{eval}} - \widehat{\boldsymbol{\mu}}_X^{(\text{cal})}) \widehat{\boldsymbol{\beta}}(\tau, \lambda)\|_2^2. \tag{3.7}$$

We choose the pair (τ_0, λ_0) which minimizes $10^{-1} \sum_{i=1}^{10} MSE(\tau, \lambda, i)$. Then the direction $\widetilde{\boldsymbol{\alpha}}$ of the final estimate solves

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^p} \quad (\mathbf{y} - \widehat{\boldsymbol{\mu}}_y)^{\mathrm{T}} (\mathbf{X} - \widehat{\boldsymbol{\mu}}_X) \boldsymbol{\alpha}, \quad \text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}} (\mathbf{X} - \widehat{\boldsymbol{\mu}}_X)^{\mathrm{T}} (\mathbf{X} - \widehat{\boldsymbol{\mu}}_X) \boldsymbol{\alpha} + \tau_0 \|\boldsymbol{\alpha}\|_{\lambda_0}^2 \leq 1,$$

where $\widehat{\boldsymbol{\mu}}_X$ and $\widehat{\boldsymbol{\mu}}_y$ are the mean matrix and the mean vector of the whole data set $\mathbf{X}$ and $\mathbf{y}$, respectively. The final estimate $\widehat{\boldsymbol{\beta}}$ is

$$\widehat{\boldsymbol{\beta}} = \frac{(\mathbf{y} - \widehat{\boldsymbol{\mu}}_y)^{\mathrm{T}}(\mathbf{X} - \widehat{\boldsymbol{\mu}}_X)\widetilde{\boldsymbol{\alpha}}}{\widetilde{\boldsymbol{\alpha}}^{\mathrm{T}}(\mathbf{X} - \widehat{\boldsymbol{\mu}}_X)^{\mathrm{T}}(\mathbf{X} - \widehat{\boldsymbol{\mu}}_X)\widetilde{\boldsymbol{\alpha}}}\widetilde{\boldsymbol{\alpha}}.$$

It is easy to see that $(\mathbf{X} - \widehat{\boldsymbol{\mu}}_X)\widehat{\boldsymbol{\beta}}$ is the projection of $\mathbf{y} - \widehat{\boldsymbol{\mu}}_y$ along the direction of $(\mathbf{X} - \widehat{\boldsymbol{\mu}}_X)\widetilde{\boldsymbol{\alpha}}$.

3.3 Simultaneous model selection consistency and parameter estimation consistency in high dimension

Zhao and Yu (2006) proved that the Lasso is model consistent. We will prove that our method is both model selection consistent and parameter estimation consistent simultaneously under the similar setting. Assume that we have a sequence of linear regression models

$$\mathbf{y}^n = \mathbf{X}^n\boldsymbol{\beta}^n + \boldsymbol{\varepsilon}^n, \tag{3.8}$$

where $\boldsymbol{\varepsilon}^n = (\varepsilon_1^n, \cdots, \varepsilon_n^n)^{\mathrm{T}}$ is a vector of i.i.d. standard normal variables, $\mathbf{y}^n$ is the n-dimensional response vector and $\mathbf{X}^n$ is the $n \times p$ data matrix. We will consider the situation where both n and p go to infinity. Suppose that the first q coordinates of $\boldsymbol{\beta}^n$ are nonzero and the others are zero. Let $\boldsymbol{\beta}^n = (\boldsymbol{\beta}^{n_1\mathrm{T}}, \boldsymbol{\beta}^{n_2\mathrm{T}})^{\mathrm{T}}$, where $\boldsymbol{\beta}_1^n$ and $\boldsymbol{\beta}_2^n = \mathbf{0}$ are the first q and the last $p - q$ coordinates of $\boldsymbol{\beta}^n$, respectively.

Suppose that $\widehat{\boldsymbol{\beta}^n} = (\widehat{\boldsymbol{\beta}_1^n}^{\mathrm{T}}, \widehat{\boldsymbol{\beta}_2^n}^{\mathrm{T}})^{\mathrm{T}}$ is the solution to (3.2) with the values of the tuning parameters equal to λ_n and τ_n, where $\widehat{\boldsymbol{\beta}_1^n}$ and $\widehat{\boldsymbol{\beta}_2^n}$ are the first q and the last $p - q$ coordinates of $\widehat{\boldsymbol{\beta}^n}$, respectively. Then our estimate is

$$\widehat{\boldsymbol{\gamma}^n} = \frac{(\mathbf{y}^n)^{\mathrm{T}}\mathbf{X}^n\widehat{\boldsymbol{\beta}^n}}{\widehat{\boldsymbol{\beta}^n}^{\mathrm{T}}\mathbf{X}^{n\mathrm{T}}\mathbf{X}^n\widehat{\boldsymbol{\beta}^n}}\widehat{\boldsymbol{\beta}^n}, \tag{3.9}$$

that is, $\mathbf{X}^n\widehat{\boldsymbol{\gamma}^n}$ is the projection of $\mathbf{y}^n$ along the direction of $\mathbf{X}^n\widehat{\boldsymbol{\beta}^n}$.

We say $\widehat{\boldsymbol{\beta}^n}$ has the same sign as $\boldsymbol{\beta}^n$ if each coordinate of $\widehat{\boldsymbol{\beta}_1^n}$ has the same sign as the corresponding coordinate of $\boldsymbol{\beta}_1^n$ and $\widehat{\boldsymbol{\beta}_2^n} = \mathbf{0}$. We say $\widehat{\boldsymbol{\beta}^n}$ is model selection consistent if with probability

converging to 1, it has the same sign as $\boldsymbol{\beta}^n$. If $\|\widehat{\boldsymbol{\beta}^n} - \boldsymbol{\beta}^n\|_2 \to 0$ in probability, we say $\widehat{\boldsymbol{\beta}^n}$ is parameter estimation consistent. We will prove the simultaneous model selection consistency and parameter estimation consistency for both $\widehat{\boldsymbol{\beta}^n}$ and $\widehat{\boldsymbol{\gamma}^n}$.

We consider a setting essentially the same as in Zhao and Yu (2006). Let $\mathbf{X}^n = (\mathbf{X}_1^n, \mathbf{X}_2^n)$, where $\mathbf{X}_1^n$ and $\mathbf{X}_2^n$ are the submatrices corresponding to $\boldsymbol{\beta}_1^n$ and $\boldsymbol{\beta}_2^n$. Let $\mathbf{C}^n = \mathbf{X}^{n\mathrm{T}}\mathbf{X}^n/n$, $\mathbf{C}_{11}^n = \mathbf{X}^{n_1\mathrm{T}}\mathbf{X}_1^n/n$ and $\mathbf{C}_{21}^n = \mathbf{X}^{n_2\mathrm{T}}\mathbf{X}_1^n/n$. Assume that there exist constants $0 \le c_1 < c_2 \le 1$, $0 < c_4 < \frac{c_1}{2} < \frac{c_0}{2}$, $c_3 > 0$, positive M_1, M_2 and a positive integer k such that the following conditions hold,

Condition 1. *1. The largest singular values of* $\mathbf{C}_{21}$ *are less than* $O(n^{-c_0})$.

2. *All the eigenvalues of* $\mathbf{C}^n$ *are less than* M_1*, and all the eigenvalues of* $\mathbf{C}_{11}^n$ *are greater than* M_2.

3. $n^{\frac{1-c_2}{2}} \min_{1\le i\le q} |\boldsymbol{\beta}_j^n| \ge M_3$, $\|\boldsymbol{\beta}_1^n\|_2 \sim n^{c_3}$, $E[\varepsilon_i^{2k}] < \infty$, $q_n = O(n^{c_1})$, $p_n \le O(n^{c_4 k})$.

Theorem 3.2. *Under the Condition 1, if we choose* $\tau_n = n^{d_1}$ *and* $\lambda_n = n^{d_2}$*, where* $-\infty < d_1 < \infty$ *and* $d_2 \le 0$ *are two constants satisfying*

$$-c_0 < d_2 < -\frac{c_1}{2}, \quad \frac{1}{2} + c_4 < d_1 + \max(0, c_1 + d_2) + c_3 < \frac{1+c_2}{2}, \tag{3.10}$$

then we have

$$P\left(\widehat{\boldsymbol{\beta}^n} \text{ has the same sign as } \boldsymbol{\beta}^n\right) \ge 1 - O\left(n^{-\delta k}\right), \tag{3.11}$$
$$P\left(\widehat{\boldsymbol{\gamma}^n} \text{ has the same sign as } \boldsymbol{\beta}^n\right) \ge 1 - O\left(n^{-\delta k}\right),$$

where δ *is a positive constant only depending on* $c_0 \sim c_4$ *and* $d_1 \sim d_2$*. Moreover, both* $\widehat{\boldsymbol{\beta}^n}$ *and* $\widehat{\boldsymbol{\gamma}^n}$ *are consistent estimates of* $\boldsymbol{\beta}^n$*. That is,*

$$\|\widehat{\boldsymbol{\beta}^n} - \boldsymbol{\beta}^n\|_2 \to 0, \quad \|\widehat{\boldsymbol{\gamma}^n} - \boldsymbol{\beta}^n\|_2 \to 0, \tag{3.12}$$

in probability as $n \to \infty$.

4 Sparse discriminant analysis

4.1 Motivation

Although Fisher's discriminant analysis performs well in low-dimensional settings, it faces major problems for high-dimensional data. The within-class covariance matrix, $\mathbf{\Sigma}_w$, is singular in (2.3) when the sample size is smaller than the dimension, and hence there is no solution to (2.3). Even in the case where p is close to n and $\mathbf{\Sigma}_w$ is not singular, the resulting classifier will have large variance and poor performance. To address these problems, regularized discriminant analysis methods have been proposed, including those described in Friedman (1989), Krzanowski *et al.* (1995), Dudoit *et al.* (2001), Bickel and Levina (2004), Guo *et al.* (2007), Xu *et al.* (2009), Tibshirani *et al.* (2002), Witten and Tibshirani (2011), Clemmensen *et al.* (2011), Shao *et al.* (2011) and many others. However, when sparseness penalties are imposed, existing methods cannot simultaneously achieve sparsity and zero within-class and between-class correlations among the components. As discussed in Section 2.2, the lack of correlation property controls the shapes of the distributions of the class means and the observations about the mean in the projection space or the score space, and makes the separation of classes easier. Although one can use orthogonalization to achieve this property, the components thus obtained do not achieve the optimal between-class variances, i.e., the maximum between-class variances subject to the corresponding constraints. In this section, we propose a new sparse discriminant analysis method based on the relationship between regression by projection and discriminant analysis. This method leads to both sparse and uncorrelated components.

4.2 Sparse discriminant analysis

We propose the following sequential method to find the linear combinations $\mathbf{X}\boldsymbol{\alpha}_1, \cdots, \mathbf{X}\boldsymbol{\alpha}_{K-1}$. For any $1 \le k \le K-1$, the coefficient $\boldsymbol{\alpha}_k$ solves

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in\mathbb{R}^p} \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_b\boldsymbol{\alpha}, \\ &\text{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha} + \tau\|\boldsymbol{\alpha}\|_\lambda^2 \le 1, \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_b\boldsymbol{\alpha}_j = 0, \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha}_j = 0, \quad j < k. \end{aligned} \tag{4.1}$$

Because $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha} + \tau\|\boldsymbol{\alpha}\|_\lambda^2 = \boldsymbol{\alpha}^{\mathrm{T}}\{\boldsymbol{\Sigma}_w + \tau(1-\lambda)\mathbf{I}\}\boldsymbol{\alpha} + \tau\lambda\|\boldsymbol{\alpha}\|_1^2$, where $\mathbf{I}$ is the p-dimensional identity matrix, our method resolves the singularity problem and achieves sparse components. The constraints $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_b\boldsymbol{\alpha}_j = 0$ and $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha}_j = 0$ guarantee that there is no within-class and between-class correlations, respectively.

We propose the following iterative algorithm to solve (4.1).

Algorithm 4.1. *1. Choose an initial vector $\boldsymbol{\alpha}^{(0)}$ with $\boldsymbol{\Sigma}_b\boldsymbol{\alpha}^{(0)} \ne \mathbf{0}$.*

2. Iteratively compute a sequence $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \cdots, \boldsymbol{\alpha}^{(i)}, \cdots$ until convergence as follows: for any $i \ge 1$, compute $\boldsymbol{\alpha}^{(i)}$ by solving

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in\mathbb{R}^p} (\boldsymbol{\Sigma}_b\boldsymbol{\alpha}^{(i-1)})^{\mathrm{T}}\boldsymbol{\alpha}, \\ &\textit{subject to} \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha} + \tau\|\boldsymbol{\alpha}\|_\lambda^2 \le 1, \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_b\boldsymbol{\alpha}_j = 0, \quad \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha}_j = 0, \quad j < k. \end{aligned} \tag{4.2}$$

Both the key step (4.2) of Algorithm 4.1 and the sparse regression by projection problem (3.1) are special cases of the optimization problem

$$\max_{\mathbf{u}} \mathbf{c}^{\mathrm{T}}\mathbf{u}, \quad \text{subject to} \quad \mathbf{u}^{\mathrm{T}}\mathbf{C}\mathbf{u} + \tau\|\mathbf{u}\|_\lambda^2 \le 1, \quad \mathbf{D}\mathbf{u} = 0. \tag{4.3}$$

In fact, letting $\mathbf{u} = \boldsymbol{\alpha}$, $\mathbf{c} = \boldsymbol{\Sigma}_b\boldsymbol{\alpha}^{(i-1)}$, $\mathbf{C} = \boldsymbol{\Sigma}_w$ and

$$\mathbf{D} = (\boldsymbol{\Sigma}_b\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\Sigma}_b\boldsymbol{\alpha}_{k-1}, \boldsymbol{\Sigma}_w\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\Sigma}_w\boldsymbol{\alpha}_{k-1})^{\mathrm{T}}$$

in (4.3), we obtain (4.2). Since (4.3) is a sparse version of (2.5), we call it *sparse regression by projection with linear constraints*. We will propose efficient algorithms to solve (4.3) in Section 7.

Two possible modifications of our method are worth investigating. First, remove the within-class constraints $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_w\boldsymbol{\alpha}_j = 0$ in (4.1), so that only the between-class correlations are zeros. Second, remove the constraints: $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\Sigma}_b\boldsymbol{\alpha}_j = 0$, so that only the within-class correlations are zeros among the components. Algorithm 4.1 can be applied to these two modifications with changes of the matrix $\mathbf{D}$ in (4.3). We will compare the performance of our method with the two modifications in simulation studies.

4.3 Choices of tuning parameters and the number of components

We next propose a cross-validation method to choose the tuning parameters τ and λ. As for the number of components, although it can also be chosen by cross-validation or other methods, we will just fix it to be $K-1$, the largest one of possible numbers, for two reasons: (a) in almost all our studies, the best choice is $K-1$; (b) since the number of components is a discrete parameter, its selection by cross-validation may lead to large variances of the test errors.

To choose τ and λ, we repeat the following procedure 10 times. In the i^{th} repeat, where $1 \le i \le 10$, the data set is randomly split into a training set and a validation set. The validation set has one third of all observations: the proportion of the observations assigned to the validation set has to be reduced if the total sample size is small. The coefficients $\boldsymbol{\alpha}_i$, $1 \le i \le K-1$, are calculated based on the training data and the classification errors are calculated based on the validation data for each pair (τ, λ) in a grid. Then the mean errors are calculated for the ten repeats. The pair of the parameters minimizing the mean error are chosen. The final estimates of the coefficients are determined by the whole data and the selected parameters. If there are ties in the minimum mean errors between different (τ, λ), we choose the smallest τ. If for the selected τ, there is more than one λ corresponding to the minimum error, we choose the smallest λ.

5 Simulation studies

5.1 Sparse regression by projection

In this subsection, we compare our regression method, denoted by SRP, with several sparse regression methods using publicly available software: ridge regression (Ridge) (Hoerl and Kennard, 1970), LASSO (Tibshirani, 1996), elastic net (EN) (Zou and Hastie, 2005), and sparse partial least squares regression (SPLS) (Chun and Keles, 2010), the smoothly clipped absolute deviation (SCAD) (Fan and Li, 2001), penalty and the minimax concave penalty (MCP) (Zhang, 2010). The first three methods are implemented in the R package "glmnet", the SPLS in "spls", the last two in "conreg". We will consider two sets of simulations. The first one has similar settings as those in Zou and Hastie (2005) and the other one as those in Chun and Keles (2010) and Bair *et al.* (2006). To compare variable selection, we consider the sensitivity and specificity defined by

$$\text{sensitivity} = \frac{TP}{TP+FN}, \quad \text{specificity} = \frac{TN}{TN+FP},$$

where TP is the number of the variables with $\beta_i \neq 0$ and its estimate $\widehat{\beta}_i \neq 0$, i.e., the number of the true features identified, Also, FN is the number of the variables with $\beta_i \neq 0$ and $\widehat{\beta}_i = 0$, i.e., the number of the true features not identified, TN is the number of the variables with $\beta_i = 0$ and $\widehat{\beta}_i = 0$, and FP is the number of the variables with $\beta_i = 0$ and $\widehat{\beta}_i \neq 0$.

5.1.1 First set of simulation studies

Data are simulated from the true model: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sigma\epsilon$, $\epsilon \sim \text{Normal}(0,1)$ with three examples discussed below. In each example, we compare the performance of these methods for different numbers of variables and different values of σ. For each setting in any example, we simulate 100 independent data sets. Each data set has 500 independent observations which is split into a training set with 50 observations, a validation set with 50 observations and a test set with 400 observations. Models are fitted to the training data, tuning parameters are selected based on the validation data and the mean-squared error is calculated based on the test data. Here are the details of the three

examples.

Example 1. $\boldsymbol{\beta} = (\underbrace{3, \cdots, 3}_{15}, \underbrace{0, \cdots, 0}_{p-15})$ and $\mathbf{X}$ is generated as follows:

$$\mathbf{x}_i = Z_1 + \sigma^X \epsilon_i^X, \quad \epsilon_i^X \sim \text{Normal}(0,1), \quad Z_1 \sim \text{Normal}(0,1), \quad i = 1, \cdots, 5,$$

$$\mathbf{x}_i = Z_2 + \sigma^X \epsilon_i^X, \quad \epsilon_i^X \sim \text{Normal}(0,1), \quad Z_2 \sim \text{Normal}(0,1), \quad i = 6, \cdots, 10,$$

$$\mathbf{x}_i = Z_3 + \sigma^X \epsilon_i^X, \quad \epsilon_i^X \sim \text{Normal}(0,1), \quad Z_3 \sim \text{Normal}(0,1), \quad i = 11, \cdots, 15,$$

where Z_j, $1 \le j \le 3$, ϵ_i^X, $1 \le i \le 15$, and $\mathbf{x}_i \sim \text{Normal}(0, (\sigma^X)^2)$, $16 \le i \le p$, are independent. We consider $p = 100, 300, 500$, $(\sigma^X)^2 = 1, 2$, and $\sigma^2 = 1, 5, 10$, respectively.

Example 2. $\boldsymbol{\beta} = (\underbrace{3, 2, 1.5, 0, 0, 3, 2, 1.5, 0, 0, 3, 2, 1.5, 0, 0, 3, 2, 1.5, 0, 0, 3, 2, 1.5, 0, 0}_{25}, \underbrace{0, \cdots, 0}_{p-25})$ and $\mathbf{X}$ is generated from a multinormal distribution with $\text{cov}(\mathbf{x}_i, \mathbf{x}_j) = \rho^{|i-j|}$. We consider $p = 100, 300$, $\rho = 0.50, 0.95$ and $\sigma^2 = 1, 5, 10$, respectively.

Example 3. $\boldsymbol{\beta} = (\underbrace{2, 2, \cdots, 2}_{20}, \underbrace{0, \cdots, 0}_{p-20})$ and $\mathbf{X}$ is generated from a multinormal distribution with $var(\mathbf{x}_i) = 1$ and $cov(\mathbf{x}_i, \mathbf{x}_j) = \rho, \quad i \ne j$. We consider $p = 100, 300$, $\rho = 0.50, 0.90$ and $\sigma^2 = 1, 10, 20$, respectively.

In our method, SRP, the parameters are selected from the grid of τ =0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500 and $\lambda = 0.1, 0.2, \cdots, 0.9$. For elastic net, see (1.1), the parameters are selected from the grid of λ =0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500 and $\alpha = 0.1, 0.2, \cdots, 0.9$. For LASSO, the special case of (1.1) with $\alpha = 1$, and Ridge regression, the special case of (1.1) with $\alpha = 0$, the parameters are selected from the grid of λ =0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500. We choose two parameters $0 < \eta < 1$ and K (the number of components) for SPLS from the grid of $\eta = 0.1, 0.2, \cdots, 0.9$ and $K = 1, 2, \cdots, 10$. For SCAD and MCP, we use the default setting of the R function where the tuning parameter is selected from one hundred values.

The averages and standard deviations of the test error, sensitivity and the specificity over the 100 independent data sets are listed in the following tables. Table 8 summarizes the averages and standard deviations (in parentheses) of the test errors. The second column for each competing

method is the mean squared error efficiency of the competing method (the ratio between the average MSEs of our method and the competing method). Table 1 gives the sensitivities and the specificities (in parenthesis) for Example 1. When both the number of variables and the noise are small, our method and SPLS have comparable prediction performance and are better than the other methods. In these scenarios, all the methods except SCAD and MCP identify almost all true features, but SPLS has a better specificity. When the number of variables or the noise are large, our method has the smallest prediction errors, which is statistically significant by the paired t-test. For example, in the case of $p = 300, \sigma^X = 1, \sigma = 1$, for the alternative hypothesis: "our method has a smaller expectation of MSE", the p-values are 7.3×10^{-13} (EN), 6.5×10^{-9} (LASSO), 2.2×10^{-16} (Ridge), and 4.9×10^{-6} (SPLS), respectively. In these scenarios, SPLS has the lower sensitivity and higher specificity, that is, it tends to choose a model with both fewer true signals and fewer noisy features. It seems that the prediction accuracy of both SCAD and MCP is very sensitive to feature selection. For example, consider the scenario corresponding the first line of Table 8. The average MSE of those simulation runs where all the 15 true features are selected was 1.9, but the average MSE of those runs where 14 true features were identified is 53.5. For MCP, the two averages were 1.48 and 60.3, respectively.

The results for Example 2 are shown in Tables 9 and 2. Tables 10 and 3 give results for Example 3. Our method has good prediction performance in all the scenarios.

5.1.2 Second set of simulations

We use the same simulation settings as those in Chun and Keles (2010) and Bair *et al.* (2006). In both simulated data sets, $p = 5000$ and $n = 100$. We simulate data from the general model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$, $\epsilon \sim \text{Normal}(0, 1.5^2)$, where $\boldsymbol{\beta}$ is a p vector with $\beta_j = 1/25$ for $1 \le j \le 50$ and 0 for $51 \le j \le p$. The underlying data generation for $\mathbf{X}$ is different between the two simulated data sets.

Example 4. Two hidden components H_1 and H_2 are defined as follows: H_{1j} equals 3 for $1 \le j \le 50$ and 4 for $51 \le j \le 100$ and H_{2j} equals 3.5 for $1 \le j \le 100$. The columns of $\mathbf{X}$ are generated by $\mathbf{X}_i = H_1 + \boldsymbol{\epsilon}_i^X$ for $1 \le i \le 50$ and $\mathbf{X}_i = H_2 + \boldsymbol{\epsilon}_i^X$ for $51 \le i \le p$, where $\boldsymbol{\epsilon}_i^X$ are independent and identically distributed random vectors whose elements are independent standard normal random

variables.

Example 5. Five hidden components $H_1, \cdots, H_5$ defined as follows: $H_{1j} = 3\mathbf{I}(j \leq 50)+4\mathbf{I}(j > 50)$, $H_{2j} = 3.5 + 1.5\mathbf{I}(u_{1j} \leq 0.4)$, $H_{3j} = 3.5 + 0.5\mathbf{I}(u_{2j} \leq 0.7)$, $H_{4j} = 3.5 - 1.5\mathbf{I}(u_{3j} \leq 0.7)$, $H_{5j} = 3.5$, where u_{1j}, u_{2j} and u_{3j} are independent random variables uniformly distributed in $(0, 1)$ and $\mathbf{I}$ is the indicator function. The columns of $\mathbf{X}$ are generated by $\mathbf{X}_i = H_k + \boldsymbol{\epsilon}_i^X$ for $n_{k-1} + 1 \leq i \leq n_k$, where $k = 1, \cdots, 5$ and $(n_0, \cdots, n_5) = (0, 50, 100, 200, 300, p)$.

We compare the prediction performance of our method with elastic net and SPLS. In our method, we use cross-validation to select the tuning parameters as described in Section 3.2. For the elastic net and SPLS, the ten-fold cross-validation methods provided in the corresponding packages are used to select the tuning parameters. The results are summarized in Table 4 showing the mean errors and the standard deviations (in parentheses) of 100 simulations for each setting. The p-values of the paired t-tests for the comparison of our method and each of the other two methods are also calculated. For Example 4, our method and SPLS have almost the same prediction accuracy, whereas our method is better in Example 5.

5.2 Sparse discriminant analysis

The goal of this simulation study is to show that imposing the sparsity penalty and the constraints of no within-class and between-class correlations among the components simultaneously can improve classification. We compare the prediction performances of the following regularized discriminant analysis methods: our method (denoted by sdaBP), the modification of our method without the constraint on between-class correlations (sdaBP2), the modification of our method without the constraint on within-class correlations (sdaBP3), RDA (Guo *et al.* (2007), R package "rda"), PDA (Witten and Tibshirani (2011), "penalizedLDA"), and SDA (Clemmensen *et al.* (2011), "sparseLDA"). Four simulation models are considered. In each simulation, 50 independent data sets are simulated each of which has 1500 observations and three classes. Each observation is randomly assigned to one class and then the values of the covariates are generated from the model. Then the observations are randomly split into the training set with 150 observations and the test

set with 1350 observations. Each simulation consists of measurements on 500 features. For the first three methods, we use the cross-validation methods in Section 4.3 to select the parameters. For RDA and PDA, the cross-validation methods in the corresponding packages are used. Since there is no method of parameter selection available in "sparseLDA" and only one parameter can be tuned there, we use ten-fold cross-validation in the grid of $10^{-6}, 10^{-4}, 10^{-2}, 1, 10^2, 10^4$, to choose the parameter.

(a). *Simulation 1:* There is no overlap between the features for different classes and different variables are independent. Specifically, let x_{ij} be the i^{th} observation on the j^{th} variable. If the i^{th} observation belongs to class $k(= 1, 2, 3)$, then $x_{ij} \sim \text{Normal}(\mu_{kj}, \sigma_j)$. The mean vector of class k, $\boldsymbol{\mu}_k = (\mu_{k1}, \cdots, \mu_{kp})$ with $\mu_{1j} = 1$ if $1 \leq j \leq 10$, $\mu_{2j} = 1$ if $11 \leq j \leq 30$, $\mu_{3j} = 1$ if $31 \leq j \leq 60$, and $\mu_{kj} = 0$ otherwise. Finally, σ_j is a random number generated from the uniform distribution in $(0.5, 2)$.

(b). *Simulation 2:* There is no overlap between the features for different classes, but the variables are correlated. If the i^{th} observation is in class $k(= 1, 2, 3)$, then $x_{ij} = \mu_{kj} + Z_{1i} + \epsilon_{ij}$ if $1 \leq j \leq 20$, $x_{ij} = \mu_{kj} + Z_{1i} + Z_{2i} + \epsilon_{ij}$ if $21 \leq j \leq 30$, $x_{ij} = \mu_{kj} + Z_{2i} + \epsilon_{ij}$ if $31 \leq j \leq 50$ and $x_{ij} = \mu_{kj} + \epsilon_{ij}$ otherwise, where $Z_{1i} \sim \text{Normal}(0, 1)$, $Z_{2i} \sim \text{Normal}(0, 1)$ and $\epsilon_{ij} \sim \text{Normal}(0, 0.8^2)$ are independent. $\mu_{1j} \sim \text{Normal}(1, 0.8^2)$ if $1 \leq j \leq 20$, $\mu_{2j} \sim \text{Normal}(4, 0.8^2)$ if $21 \leq j \leq 30$, $\mu_{3j} \sim \text{Normal}(1, 0.8^2)$ if $31 \leq j \leq 50$ and $\mu_{kj} = 0$ otherwise.

(c). *Simulation 3:* There are overlaps between the features for different classes and the variables are correlated. The vector $\mathbf{x}_i \sim \text{Normal}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ if observation i is in class k, where the covariance structure is block diagonal, with five blocks each of dimension 100×100. The blocks have (j, j') element $0.6^{|j-j'|}$. Also, $\mu_{1j} \sim \text{Normal}(1, 1)$, $\mu_{2j} \sim \text{Normal}(2, 1)$ and $\mu_{3j} \sim \text{Normal}(3, 1)$ if $1 \leq j \leq 10$ or $101 \leq j \leq 110$ and $\mu_{kj} = 0$ otherwise.

(d). *Simulation 4:* In the first three simulations, observations in all the classes have the same distributions about the class means. A different situation is considered here. If the i^{th} observation is in class k, $\mathbf{x}_i \sim \text{Normal}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. We take $\mu_{1j} = 3$ if $1 \leq j \leq 10$, $\mu_{2j} = 2$ if $1 \leq j \leq 20$, $\mu_{3j} = 1$ if $1 \leq j \leq 30$, and $\mu_{kj} = 0$ otherwise. The covariance matrix $\boldsymbol{\Sigma}_1$ is diagonal with the

diagonal elements generated from the uniform distribution in $(0.5, 2)$. $\mathbf{\Sigma}_2$ is block diagonal, with five blocks each of dimension 100×100. The blocks have (j, j') element $0.9^{|j-j'|}$. And $\mathbf{\Sigma}_3$ is block diagonal, with five blocks each of dimension 100×100. The blocks have (j, j') element 0.6 if $j \neq j'$ and 1 otherwise.

The mean misclassification rates (percentages) of 50 data sets for each simulation are shown in Table 5, with standard deviations in parentheses. The p-value for the paired t-test between sdaBP and each of the other methods is also calculated. Our method has good prediction accuracies in all the simulations. The unusual large error rates of SDA may be due to our choices of the parameters. The sdaBP and the sdaBP2 have similar performances and are better than the sdaBP1. Hence, to remove the between-class correlation has a larger effect on the prediction than the within-class correlation. The benefit of controlling of the between-class and the within-class correlations can be illustrated by Figures 2 and 3. In Figure 2, the class means of PDA lie approximately along a straight line, i.e., large between-class correlation, which leads to large overlaps of different classes. In Figure 3, the observations in the plot of PDA distribute along a particular direction, i.e., large within-class correlation, which leads to large overlaps of the red and the blue classes.

6 Case studies

6.1 Predictive modelling of anticancer drug sensitivity

In Barretina and et al. (2012), the elastic net was used to construct predictive models that explained drug sensitivity profiles based on genetic features of the cell lines. In this study, we apply our method (SRP), ridge regression (Ridge), LASSO (LASSO), elastic net (EN), and SPLS to this data set. The numbers of variables and observations are 54,675 and 491, respectively. There are 24 drugs considered. For each drug, we construct a regression model to predict drug sensitivity. We randomly split the observations into the training set with 100 observations, the validation set with 100 observations and the test set with 291 observations. We repeat the procedure 20 times and calculate the means and the standard deviations of the MSE and the number of the features

selected for each drug. The results for the first fives drugs are shown in Table 6. The results for all the drugs can be found in Table 3 of the Appendix. All methods except LASSO have almost the same prediction performance. The prediction errors of LASSO are slightly larger than others. Our method and LASSO included the smallest numbers of features in the models for all drugs.

6.2 Classification

We next apply our sdaBP method, RDA, SDA and PDA to four data sets which are randomly split into training sets and test sets. For each data set, the procedure is repeated 50 times and the mean and standard deviation of misclassification rates are calculated.

(a). *UPP data*: Gene expression data from a breast cancer study published by Miller and et al. (2005). There are 44,928 features and 249 samples classified into three grades with 67, 128, 54 observations, respectively. The data is randomly split into a training set with 150 observations and a test data set with 99 observations. The data is available in the package "breastCancerUPP" of "Bioconductor".

(b). *NKI data*: Gene expression data from a breast cancer study published by vanť Veer *et al.* (2002). There are 24,481 features and 337 samples classified into three grades with 79, 109, 149 observations, respectively. The data is randomly split into a training set with 150 observations and a test data set with 187 observations. The data is available in the package "breastCancerNKI" of "Bioconductor".

(c). *DLBCL-D data*: Microarray data from the Broad Institute "Cancer Program Data Sets" which was produced by Yujin *et al.* (2007). There 3,741 features and 129 samples classified into four groups with 19, 37, 24 and 49 observations, respectively. The training set has 109 observations and the test set has 20 observations.

(d). *Handwriting data*: This data set consists of features of handwritten numerals, $0, 1, \cdots, 9$, extracted from a collection of Dutch utility maps. For each numerals (that is, each class), there are 200 observations. Hence, there are 10 classes, 649 features and 2,000 observations

randomly split into a training set with 450 observations and a test set with 1,550 observations. The data is available at the UCI Machine Learning Repository.

The results are summarized in Table 7. Our method has good performance both for high-dimensional data and for the data with a relatively large number of classes.

7 Discussion

In this paper, we have proposed a the new framework, regression by projection, and its sparse version for high dimensional data analysis. The unique feature of our new approach is that the direction of the estimate of the coefficient vector is determined first and then its length. Tuning parameters are determined by cross-validation. Comparisons with other methods through simulations and data examples show that our method achieves good predictive performance and effective variable selection.

This framework can be generalized to PCA, PLS, CCA and discriminant analysis to develop sparse versions of these methods. In addition to the achievement of sparse components, the relationship among the components can be controlled. In this paper, we focused on sparse discriminant analysis. We showed that the control of within-class and between-class correlations among the sparse components can improve prediction accuracy. An efficient algorithm and the related theory for solving the sparse regression by projection were developed. Numerical examples show that the new algorithm is faster than LARS and comparable to the Coordinate Descent algorithm.

Supplementary Materials

Web Appendix: the detailed description of all the algorithms and the proofs of theorems. (webAppendix.pdf; pdf file)

R codes: code files to run the simulation studies. (simulation.R.codes.zip; zip file)

Acknowledgments

Carroll's research was supported by a grant from the National Cancer Institute (R37-CA057030). This publication is based in part on work supported by Award Number KUS-CI-016-04, made by King Abdullah University of Science and Technology (KAUST). Zhao's research was supported in part by NIH R01 GM59507, P01 CA154295 and NSF DMS 1106738.

References

Bair, E., Hastie, T., Paul, D. and Tibshirani, R. (2006) Prediction by supervised principal components. *Journal of the American Statistical Association*, **101**, 119–137.

Barretina, J. and et al. (2012) The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, **483**, 603–607.

Bickel, P. and Levina, E. (2004) Some theory for fishers linear discriminant function, naive bayes, and some alternatives when there are many more variables than observations. *Bernoulli*, **6**, 989–1010.

Cho, H. and Fryzlewicz, P. (2012) High dimensional variable selection via tilting. *Journal of the Royal Statistical Society: series B (statistical methodology)*, **74**, 593–622.

Chun, H. and Keles, S. (2010) Sparse partial least squares regresssion for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society*, **72**, 3–25.

Clemmensen, L., Hastie, T., Witten, D. and Ersbll, B. (2011) Sparse discriminant analysis,. *Technometrics*, **53**, 406–413.

Dudoit, S., Fridlyand, J., and Speed, T. (2001) Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, **96**, 1151–1160.

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

Friedman, J. (1989) Regularized discriminant analysis. *Journal of the American Statistical Association*, **84**, 165–175.

Guo, Y., Hastie, T. and Tibshirani, R. (2007) Regularized linear discriminant analysis and its applications in microarrays. *Biostatistics*, **8**, 86–100.

Hoerl, A. E. and Kennard, R. W. (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12:1**, 55–67.

Krzanowski, W., Jonathan, P., McCarthy, W., and Thomas, M. (1995) Discriminant analysis with singular covariance matrices: Methods and applications to spectroscopic data. *Journal of the Royal Statistical Society*, **44**, 101–115.

Miller, L. D. and et al. (2005) An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival. *Proceedings of the National Academy of Sciences of the United States of America*, **102**, 13550–13555.

Shao, J., Wang, Y., Deng, X. and Wang, S. (2011) Sparse linear discriminant analysis by thresholding for high dimensional data. *Ann. Statist.*, **39**, 12411265.

Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, **58**, 267–288.

Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, G. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, **99**, 6567–6572.

vant́ Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., an Rene Bernards, P. S. L. and Friend, S. H. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Witten, D. and Tibshirani, R. (2011) enalized classification using fishers linear discriminant. *Journal of the Royal Statistical Society, Ser. B*, **73**, 753–772.

Xu, P., Brock, G. and Parrish, R. (2009) Modified linear discriminant analysis approaches for classification of high-dimensional microarray data. *Computational Statistics and Data Analysis*, **53**, 1674–1687.

Yujin, H., Jean-Philippe, B., Pablo, T., Todd, G. and Jill, M. (2007) Subclass mapping: Identifying common subtypes in independent disease data sets. *PLoS ONE*, **2**, e1195.

Zhang, C.-H. (2010) Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, **38**, 894–942.

Zhao, P. and Yu, B. (2006) On model selection consistency of lasso. *The Journal of Machine Learning Research*, **7**, 2541–2563.

Zou, H. and Hastie, T. (2005) Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, **67**, 301–320.

Table 1: The averages of the sensitivities and specificities (in parentheses) for Example 1.

$(p, (\sigma^X)^2)$	σ^2	SRP	EN	LASSO	Ridge	SPLS	SCAD	MCP
(100, 1)	1	1(0.89)	1(0.90)	1(0.90)	1(0)	1(0.96)	0.86(0.93)	0.72(0.96)
	5	1(0.87)	1(0.82)	1(0.76)	1(0)	0.99(0.96)	0.83(0.91)	0.66(0.95)
	10	0.99(0.86)	0.99(0.81)	0.99(0.71)	1(0)	0.99(0.95)	0.77(0.89)	0.60(0.95)
(100, 2)	1	1(0.89)	1(0.89)	1(0.90)	1(0)	1(0.95)	0.94(0.95)	0.85(0.96)
	5	1(0.85)	1(0.78)	1(0.75)	1(0)	1(0.94)	0.93(0.95)	0.85(0.96)
	10	1(0.85)	1(0.78)	1(0.71)	1(0)	0.99(0.95)	0.91(0.94)	0.80(0.96)
(300, 1)	1	0.99(0.95)	0.99(0.95)	0.99(0.95)	1(0)	0.98(0.98)	0.59(0.96)	0.40(0.99)
	5	1(0.94)	1(0.92)	1(0.91)	1(0)	0.97(0.97)	0.61(0.96)	0.38(0.99)
	10	0.99(0.93)	0.99(0.91)	0.99(0.89)	1(0)	0.95(0.97)	0.60(0.97)	0.37(0.99)
(300, 2)	1	1(0.95)	1(0.94)	1(0.94)	1(0)	0.96(0.95)	0.65(0.97)	0.41(0.99)
	5	0.99(0.93)	0.99(0.91)	0.99(0.90)	1(0)	0.94(0.95)	0.68(0.97)	0.44(0.99)
	10	0.99(0.92)	0.99(0.90)	0.99(0.89)	1(0)	0.93(0.95)	0.66(0.97)	0.43(0.99)
(500, 1)	1	0.99(0.96)	0.99(0.96)	0.99(0.96)	1(0)	0.93(0.98)	0.59(0.98)	0.31(0.99)
	5	0.99(0.95)	0.99(0.94)	0.99(0.94)	1(0)	0.94(0.98)	0.58(0.98)	0.32(0.99)
	10	0.99(0.96)	0.99(0.94)	0.98(0.93)	1(0)	0.93(0.98)	0.58(0.98)	0.32(0.99)
(500, 2)	1	0.97(0.96)	0.97(0.95)	0.97(0.95)	1(0)	0.88(0.96)	0.58(0.98)	0.29(0.99)
	5	0.97(0.95)	0.97(0.94)	0.97(0.94)	1(0)	0.86(0.96)	0.57(0.98)	0.31(0.99)
	10	0.96(0.94)	0.97(0.93)	0.96(0.93)	1(0)	0.86(0.97)	0.61(0.98)	0.31(0.99)

Table 2: The averages of the sensitivities and specificities (in parentheses) for Example 2.

(p, ρ)	σ^2	SRP	EN	LASSO	Ridge	SPLS	SCAD	MCP
(100, 0.5)	1	0.99(0.82)	0.99(0.76)	0.99(0.74)	1(0)	0.99(0.90)	0.60(0.89)	0.51(0.90)
	5	0.98(0.80)	0.98(0.73)	0.97(0.65)	1(0)	0.90(0.84)	0.48(0.84)	0.41(0.89)
	10	0.95(0.86)	0.96(0.80)	0.93(0.62)	1(0)	0.86(0.87)	0.44(0.85)	0.36(0.91)
(100, 0.95)	1	0.99(0.89)	0.99(0.90)	0.98(0.93)	1(0)	0.94(0.73)	0.38(0.96)	0.36(0.96)
	5	0.96(0.87)	0.96(0.87)	0.85(0.90)	1(0)	0.95(0.73)	0.35(0.94)	0.32(0.96)
	10	0.95(0.87)	0.92(0.85)	0.75(0.88)	1(0)	0.96(0.77)	0.33(0.95)	0.30(0.96)
(300, 0.5)	1	0.96(0.92)	0.97(0.89)	0.97(0.89)	1(0)	0.82(0.94)	0.63(0.97)	0.45(0.99)
	5	0.91(0.91)	0.90(0.89)	0.89(0.87)	1(0)	0.75(0.94)	0.58(0.96)	0.42(0.99)
	10	0.83(0.91)	0.85(0.88)	0.82(0.86)	1(0)	0.70(0.94)	0.53(0.96)	0.36(0.98)
(300, 0.95)	1	0.99(0.96)	0.99(0.96)	0.97(0.97)	1(0)	0.94(0.93)	0.31(0.98)	0.27(0.99)
	5	0.95(0.96)	0.94(0.95)	0.84(0.94)	1(0)	0.93(0.93)	0.30(0.98)	0.26(0.99)
	10	0.96(0.96)	0.92(0.94)	0.74(0.93)	1(0)	0.95(0.94)	0.27(0.97)	0.23(0.99)

Table 3: The averages of the sensitivities and specificities (in parentheses) for Example 3.

(p,ρ)	σ^2	SRP	EN	LASSO	Ridge	SPLS	SCAD	MCP
$(100, 0.5)$	1	0.99(0.70)	0.99(0.65)	0.96(0.75)	1(0)	0.97(0.22)	0.71(0.87)	0.57(0.88)
	10	0.93(0.80)	0.93(0.73)	0.85(0.65)	1(0)	0.97(0.12)	0.62(0.87)	0.50(0.89)
	20	0.89(0.57)	0.89(0.57)	0.74(0.70)	1(0)	0.94(0.15)	0.59(0.86)	0.43(0.90)
$(100, 0.9)$	1	0.99(0.89)	0.99(0.90)	0.98(0.93)	1(0)	0.94(0.73)	0.47(0.90)	0.17(0.97)
	10	0.96(0.87)	0.96(0.87)	0.85(0.90)	1(0)	0.95(0.73)	0.24(0.93)	0.15(0.97)
	20	0.95(0.87)	0.92(0.85)	0.75(0.88)	1(0)	0.96(0.77)	0.13(0.95)	0.15(0.97)
$(300, 0.5)$	1	0.86(0.75)	0.88(0.68)	0.62(0.87)	1(0)	0.93(0.18)	0.42(0.93)	0.22(0.96)
	10	0.75(0.75)	0.79(0.73)	0.58(0.87)	1(0)	0.9(0.21)	0.37(0.93)	0.19(0.96)
	20	0.68(0.74)	0.71(0.74)	0.51(0.87)	1(0)	0.91(0.19)	0.35(0.92)	0.16(0.97)
$(300, 0.9)$	1	0.87(0.66)	0.95(0.46)	0.57(0.86)	1(0)	1(0)	0.32(0.94)	0.10(0.99)
	10	0.78(0.51)	0.75(0.65)	0.41(0.89)	1(0)	1(0)	0.17(0.96)	0.08(0.98)
	20	0.74(0.45)	0.61(0.71)	0.27(0.90)	1(0)	1(0)	0.11(0.97)	0.08(0.98)

Table 4: The averages and standard deviations (in parentheses) of the MSE for the simulations in Section 5.1.2.

		SRP	EN	SPLS
Example 4	mean errors(sd)	2.65(0.44)	3.23(0.54)	2.71(0.44)
	p-value		< 0.001	0.023
Example 5	mean errors(sd)	2.64(0.42)	3.16(0.42)	2.73(0.44)
	p-value		< 0.001	0.002

Table 5: The averages and standard deviations of misclassification rates (%) for the simulation in Section 5.2. Here "Sim" is the simulation setting number.

Sim		sdaBP	sdaBP2	sdaBP3	RDA	SDA	PDA
1	rates (sd)	1.15(0.67)	3.40(4.76)	1.06(0.75)	6.02(1.98)	52(2.9)	0.71(0.49)
	p-value		< 0.001	0.91	< 0.001	< 0.001	1.00
2	rates (sd)	1.31(1.06)	6.39(3.48)	1.44(1.14)	1.40(0.97)	14.6(3.7)	17.5(2.8)
	p-value		< 0.001	0.04	0.25	< 0.001	< 0.001
3	rates (sd)	0.38(0.70)	2.0(2.4)	0.35(0.67)	0.71(0.82)	35(12)	11.3(6.9)
	p-value		< 0.001	0.82	< 0.001	< 0.001	< 0.001
4	rates (sd)	1.28(1.17)	2.63(2.66)	1.20(1.09)	1.24(0.77)	42(4.3)	14.5(4.6)
	p-value		< 0.001	0.83	0.62	< 0.001	< 0.001

Table 6: The MSE and number of selected features of the first five drugs for the drug data

Drug name		SRP	EN	LASSO	Ridge	SPLS
17-AAG	MSE (sd)	1.01(0.08)	1.00(0.09)	1.14(0.10)	1.00(0.10)	0.98(0.07)
	features (sd)	117.6(56.3)	193.6(128.7)	110(4.95)	54675(0)	26321(21705)
AEW541	MSE (sd)	0.33(0.03)	0.34(0.03)	0.39(0.03)	0.34(0.03)	0.34(0.03)
	features (sd)	86(46)	125(88)	103(5)	54675(0)	12908(19988)
AZD0530	MSE (sd)	0.60(0.06)	0.59(0.04)	0.71(0.07)	0.58(0.04)	0.63(0.07)
	features (sd)	103(52)	117(84)	106(4)	54675(0)	23033(20418)
AZD6244	MSE (sd)	1.05(0.08)	1.04(0.07)	1.18(0.11)	1.01(0.05)	0.99(0.04)
	features (sd)	101(60)	184(100)	110(4)	54675(0)	16725(20690)
Erlotinib	MSE (sd)	0.35(0.03)	0.36(0.03)	0.43(0.04)	0.36(0.02)	0.37(0.03)
	features (sd)	92(37)	90(78)	100(3)	54675(0)	13717(22489)

Table 7: The averages and standard deviations of misclassification rates for the examples in Section 6.2.

Data Set	sdaBP	RDA	SDA	PDA
UPP	39.7(3.8)	42.6(5.1)	67(10.6)	48.8(3.9)
NKI	41.5(3.3)	43.0(3.3)	45.3(2.4)	70.8(12.9)
DLBCL-D	18.5(8.6)	30.1(9.5)	54.8(10.8)	71.4(9.1)
Handwriting	1.74(0.30)	2.06(0.38)	86(5)	19.5(4.0)

Table 8: The averages and standard deviations (in parentheses) of MSE for Example 1. The second column for each competing method is the mean squared error efficiency (the ratio between the average MSEs of our method and the competing method.

$(p,(\sigma^X)^2)$	σ^2	SRP	EN		LASSO		Ridge		SPLS		SCAD		MCP	
(100, 1)	1	2.05 (0.75)	2.69 (1.19)	0.76	2.66 (1.15)	0.77	446 (44)	0	2.40 (1.99)	0.85	51 (67)	0.04	94 (79)	0.02
	5	10.4 (4.5)	12.9 (5.7)	0.80	13.4 (5.4)	0.77	449 (47)	0.02	9.5 (6.2)	1.09	79 (74)	0.14	121 (77)	0.09
	10	20 (6.3)	24.3 (7.8)	0.82	27.5 (9.9)	0.72	450 (47)	0.04	18.2 (7.1)	1.09	109 (65)	0.23	146 (63)	0.17
(100, 2)	1	2.74 (1.49)	3.49 (1.65)	0.78	3.41 (1.56)	0.80	571 (63)	0	5.39 (9.33)	0.51	70 (132)	0.04	103 (165)	0.03
	5	12.4 (5.9)	15.1 (7.0)	0.82	15.3 (7.5)	0.81	581 (59)	0.02	12.5 (9.1)	0.99	78 (107)	0.16	152 (154)	0.08
	10	24.2 (10.2)	29.8 (13.7)	0.81	31.8 (15.9)	0.76	588 (61)	0.04	23.2 (16.2)	1.04	77 (103)	0.3	198 (160)	0.12
(300, 1)	1	6.51 (16.3)	9.85 (25.4)	0.66	9.17 (23.0)	0.71	561 (69)	0.01	22.8 (45)	0.28	219 (69)	0.03	258 (82)	0.03
	5	19.6 (28.4)	26.3 (38.7)	0.74	25.7 (34.5)	0.76	546 (51)	0.03	24.7 (28.9)	0.79	197 (58)	0.09	250 (71)	0.08
	10	28.2 (12.8)	38.0 (29.6)	0.74	40.5 (22.3)	0.69	558 (60)	0.05	40 (32.9)	0.70	221 (72)	0.14	261 (86)	0.11
(300, 2)	1	10.8 (19.4)	14.0 (25.2)	0.77	12.79 (22.7)	0.84	681 (56.2)	0.02	62.4 (81.3)	0.17	297 (127)	0.04	390 (163)	0.03
	5	31.1 (39.8)	41.4 (47.3)	0.75	40.2 (49)	0.77	705 (65)	0.04	65.2 (84.7)	0.47	339 (105)	0.1	409 (118)	0.08
	10	61.7 (55.4)	78.0 (65.3)	0.79	81.4 (75)	0.75	702 (62)	0.08	97.0 (82)	0.63	328 (106)	0.2	447 (139)	0.15
(500, 1)	1	10.9 (22.8)	16.9 (34.2)	0.64	17.6 (38.1)	0.61	603 (56)	0.01	39.0 (46.5)	0.27	233 (69)	0.05	298 (100)	0.04
	5	22.4 (22.6)	33.1 (33.5)	0.67	34.5 (37.9)	0.64	602 (52)	0.03	38.1 (41.1)	0.58	241 (69)	0.1	282 (94)	0.09
	10	45.6 (42)	65.3 (59)	0.69	66.6 (61)	0.68	623 (55)	0.07	63.4 (47)	0.71	251 (92)	0.18	302 (80)	0.15
(500, 2)	1	34.0 (72)	41.8 (84)	0.81	40.2 (85)	0.84	759 (58)	0.05	117 (124)	0.28	349 (110)	0.09	465 (142)	0.07
	5	63.2 (72.4)	82.8 (87.3)	0.76	82.4 (90.5)	0.76	762 (67)	0.08	164 (117)	0.38	384 (113)	0.2	478 (142)	0.15
	10	90.6 (77.7)	114 (90)	0.79	115 (93)	0.78	767 (68)	0.11	164 (114)	0.55	386 (121)	0.24	485 (131)	0.19

Table 9: The averages and standard deviations (in parentheses) of MSE for Example 2. The second column for each competing method is the mean squared error efficiency.

(p,ρ)	σ^2	SRP	EN		LASSO		Ridge		SPLS		SCAD		MCP	
(100, 0.5)	1	2.81 (1.21)	3.18 (1.31)	0.88	3.17 (1.30)	0.88	89 (9)	0.03	3.58 (3.56)	0.78	9.05 (14)	0.31	15.1 (17)	0.18
	5	13.2 (4.0)	14.9 (4.3)	0.88	17.1 (5.1)	0.77	93 (10)	0.14	17.7 (6.4)	0.74	26.8 (13)	0.5	34.7 (19)	0.38
	10	24.6 (6.7)	26.4 (6.6)	0.93	33.5 (9.7)	0.73	98 (10)	0.25	26.3 (7.2)	0.93	45.5 (14.7)	0.56	50.4 (17.8)	0.5
(100, 0.95)	1	1.65 (0.32)	1.85 (0.38)	0.89	2.04 (0.49)	0.80	104 (14)	0.02	3.19 (0.79)	0.51	7.67 (2.3)	0.22	9.2 (3.5)	0.18
	5	7.11 (0.97)	7.37 (1.07)	0.96	8.01 (1.37)	0.88	109 (15)	0.06	8.91 (1.26)	0.79	14.8 (3.9)	0.5	16.8 (5.1)	0.43
	10	13.3 (1.7)	13.9 (1.8)	0.95	15.4 (2.2)	0.86	111 (14)	0.11	15.2 (2.4)	0.87	23.3 (6.28)	0.57	25.5 (7.5)	0.52
(300, 0.5)	1	9.8 (8.8)	11.6 (10.2)	0.84	11.4 (10.4)	0.85	109 (9)	0.08	20.6 (15.3)	0.47	38.3 (22)	0.25	50 (28)	0.2
	5	27.0 (13.1)	30.4 (13.9)	0.88	31.9 (13.8)	0.84	111 (9)	0.24	35.3 (16.1)	0.76	49.7 (17)	0.55	58 (22)	0.46
	10	41.6 (16.2)	45.9 (17.7)	0.90	49.3 (17.7)	0.84	119 (9)	0.34	48.5 (18.4)	0.85	64.7 (20)	0.64	76.2 (29)	0.54
(300, 0.95)	1	1.72 (0.28)	1.95 (0.39)	0.88	2.14 (0.50)	0.80	135 (18)	0.01	4.38 (1.04)	0.39	11.7 (6.6)	0.15	16 (8.1)	0.1
	5	7.37 (0.92)	8.03 (1.17)	0.91	8.88 (1.34)	0.82	138 (18)	0.05	9.69 (1.86)	0.76	18.7 (7)	0.39	23 (9.5)	0.32
	10	13.5 (1.9)	15.0 (2.6)	0.9	18.1 (3.5)	0.74	145 (20)	0.09	15.2 (2.7)	0.89	27 (8)	0.5	33 (12)	0.4

Table 10: The averages and standard deviations (in parentheses) of MSE for Example 3. The second column for each competing method is the mean squared error efficiency.

(p,ρ)	σ^2	SRP	EN		LASSO		Ridge		SPLS		SCAD		MCP	
(100, 0.5)	1	7.55 (3.91)	12.0 (4.81)	0.62	14.75 (8.4)	0.51	47 (5)	0.16	17.9 (3.5)	0.42	56 (19)	0.13	60 (18)	0.12
	10	29.8 (5.5)	29.6 (5.4)	1.01	35.0 (8.0)	0.85	57 (7)	0.52	33.6 (4.4)	0.88	79 (19)	0.37	84 (21)	0.35
	20	46.3 (5.8)	45.7 (5.5)	1.01	56.2 (8.9)	0.82	68 (8)	0.68	48.7 (5.7)	0.95	104 (22)	0.44	108 (23)	0.42
(100, 0.9)	1	4.16 (0.92)	5.48 (1)	0.75	8.17 (2.2)	0.51	21 (2)	0.20	5.39 (0.75)	0.77	22 (5)	0.19	20 (4)	0.20
	10	16.3 (1.3)	16.5 (1.4)	0.98	19.9 (2.23)	0.82	30 (4)	0.54	17.0 (1.4)	0.96	33 (6)	0.5	29 (4)	0.56
	20	27.5 (2.6)	27.5 (2.6)	1	32.8 (3.4)	0.83	39.8 (6)	0.69	28.0 (2.5)	0.98	45 (6)	0.61	41 (5)	0.67
(300, 0.5)	1	28.2 (6.7)	31.7 (5.9)	0.89	46 (11.3)	0.61	39.8 (4.2)	0.71	34.9 (4.1)	0.81	95 (20)	0.3	108 (23)	0.26
	10	42.8 (6.1)	42.8 (6.0)	1	54.1 (10.9)	0.79	48.8 (4.1)	0.88	45.7 (4.6)	0.93	115 (25)	0.37	123 (23)	0.35
	20	57.9 (7.4)	57.2 (6.5)	1.01	69.8 (9.3)	0.83	58.7 (5.4)	0.99	57.5 (4.5)	1.01	126 (26)	0.50	147 (26)	0.39
(300, 0.9)	1	7.12 (1.06)	8.22 (1.05)	0.87	13.8 (3.0)	0.52	10.4 (1.2)	0.68	7.91 (0.81)	0.90	22.4 (5.2)	0.32	22.6 (4.3)	0.32
	10	18.1 (1.4)	18.1 (1.5)	1	22.4 (2.5)	0.81	19.6 (1.8)	0.92	18.2 (1.27)	0.99	33 (5.6)	0.54	31 (4.2)	0.58
	20	29.0 (2.5)	29.4 (2.4)	0.99	35.0 (3.1)	0.83	29.7 (2.7)	0.98	28.9 (2.2)	1	47.3 (7.2)	0.61	45.1 (7.4)	0.64

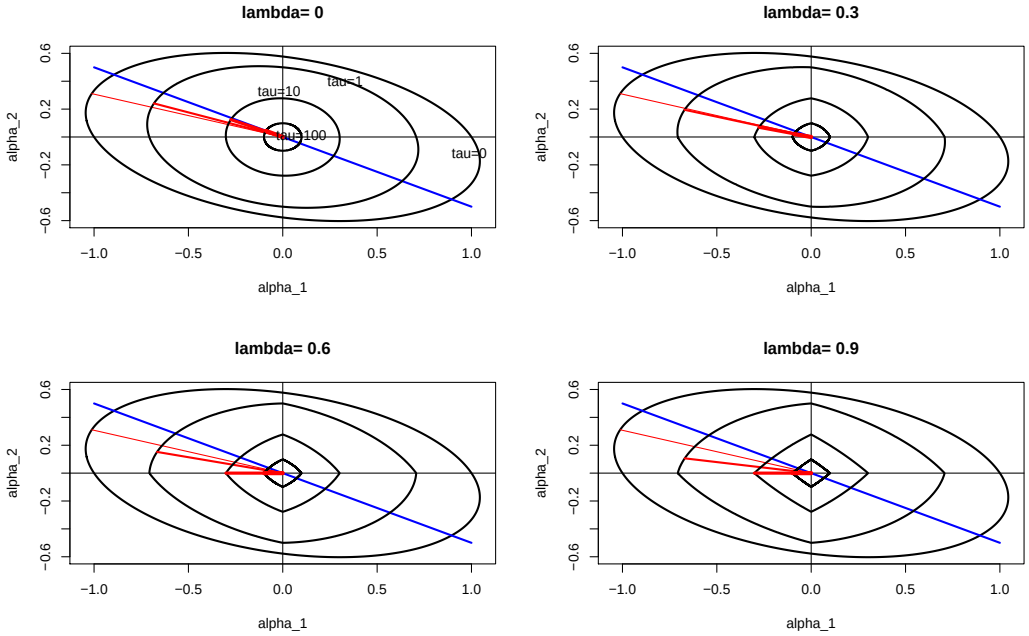

Figure 1: In each plot, for the same λ, the black curves are the contours: $\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\alpha} + \tau\|\boldsymbol{\alpha}\|_{\lambda}^{2} = 1$, for $\tau = 0, 1, 10, 100$, where $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ has diagonal elements 1 and 3 and off diagonal element 0.5. The blue line is the direction of $\mathbf{y}^{\mathrm{T}}\mathbf{X}$ and the red lines are the solutions $\widetilde{\boldsymbol{\alpha}}$ to (3.1) for the corresponding τ and λ.

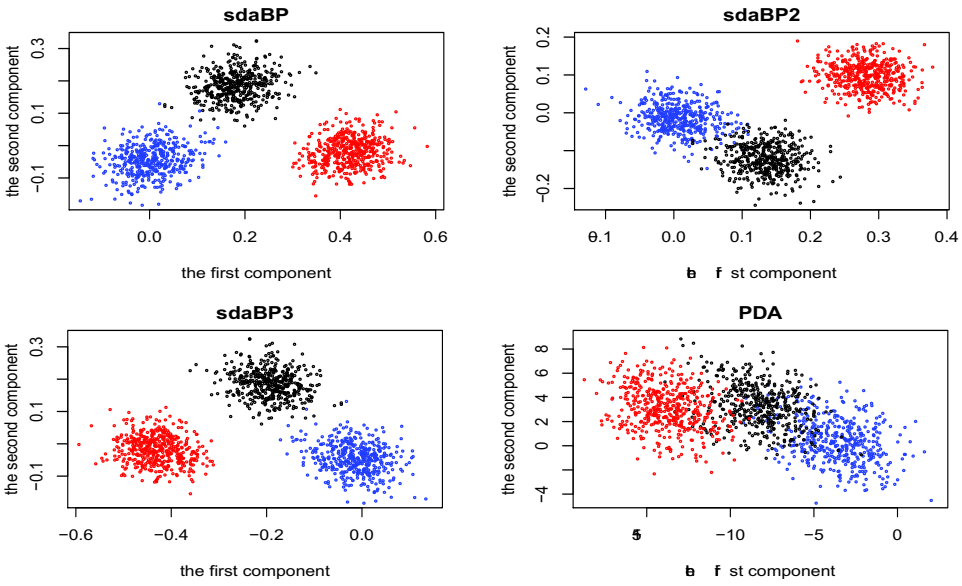

Figure 2: The projection (or the scores) of one test data set for sdaBP, sdaBP2, sdaBP3, PDA in Simulation 3 in Section 5.2.

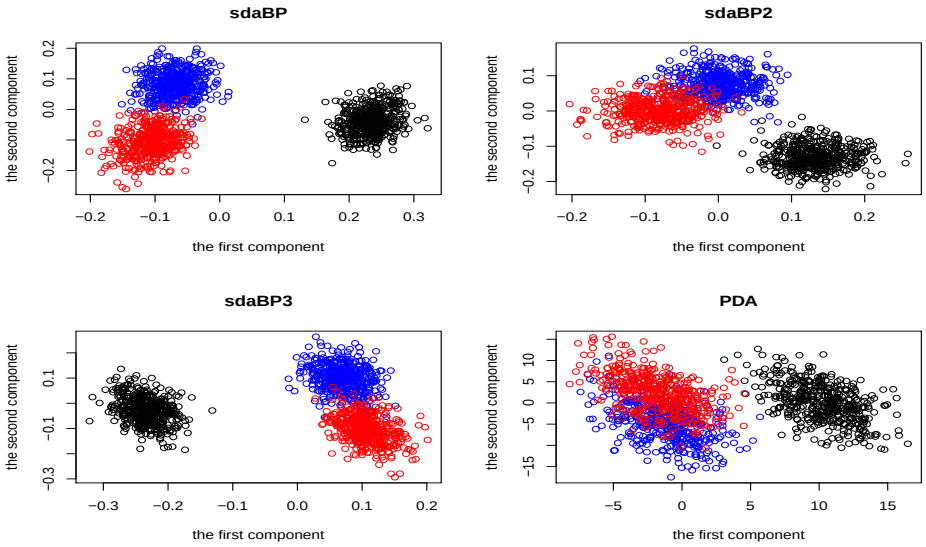

Figure 3: The projection (or the scores) of one test data set for sdaBP, sdaBP2, sdaBP3, PDA in Simulation 1 in Section 5.2.