RESPONSE-GUIDED PRINCIPAL COMPONENT CLASSIFICATION

by

Duncan Tai Bennett

Copyright © Duncan Tai Bennett 2021

A Thesis Submitted to the Faculty of the

DEPARTMENT OF MATHEMATICS

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2021

place holder for signature page

Thank you to my adviser Helen Zhang, to my wife Samantha Kao and to my family. This would not have been possible without you.

Contents

List of Abbreviations and Symbols

$\mathbf{X}$ a design matrix, usually with dimensions $N \times p$.

$\mathbf{U}$ of the svd of $\mathbf{X} = \mathbf{UDV}^T$.

$\mathbf{D}$ of the svd of $\mathbf{X} = \mathbf{UDV}^T$.

$\mathbf{V}$ of the svd of $\mathbf{X} = \mathbf{UDV}^T$.

$\mathbf{W}$ a weight matrix.

$\widetilde{\mathbf{X}}$ a weighted design matrix.

$\mathbf{p}$ a vector of probabilities of data belonging to class 1.

$\widehat{\mathbf{p}}$ a fitted vector of probabilities of data belonging to class 1, typically superscripted with the method of fit.

$\boldsymbol{\beta}$ a coefficient vector in the original coordinates.

$\widehat{\boldsymbol{\beta}}$ a fitted coefficient vector in the original coordinates, typically superscripted with the method of fit.

$\boldsymbol{\gamma}$ a coefficient vector in the principal components space.

$\widehat{\boldsymbol{\gamma}}$ a fitted coefficient vector in the principal components space, typically superscripted with the method of fit.

p_λ a penalty function with parameter λ.

ℓ the log likelihood function.

List of Figures

Abstract

Response-guided principal component regression (RgPCR) is a generalization of ridge regression. This method improves upon principal component regression (PCR) by taking the response values into account during variable selection. In this paper, we will modify RgPCR for binary classification problems using ideas from logistic regression. This technique is called *response-guided principal component classification (RgPCC)*.

1 Introduction

Principal component analysis (PCA) is a prominent unsupervised tool used in the analysis of high dimensional data. This method, as described in (1), changes the basis of the data to that of the eigenbasis of the sample covariance matrix. These eigenvectors are called *principal components*. After this change of basis, we may perform dimension reduction by prioritizing the principal components with high variance (or large eigenvalues). In other words, we project the data to a low dimensional linear subspace spanned by a subset of the eigenbasis. This dimension reduction is then typically followed with a supervised task such as regression or classification.

Principal component regression (PCR) was introduced (by Jeffers, 1967) to deal with multicollinearity. This method performms ordinary least square regression on the top K principal components. This K is typically picked so that the top K principal components account for 90% of the variance. Alternatively, K can be tuned with information criterion or crossvalidation. We can also accomplish classification tasks by following PCA with logistic regression (principal component logistic regression). While these methods have found success with respect to overtuning and dimension reduction, their downfall is that the principal components depend solely on the design $\mathbf{X}$ and in this sense the variable selection is "blind", as it does not take the response into account.

In early 2020 Lang and Zou (2) introduced Response-guided Principal Component Regression (RgPCR) to remedy the "blind" selection of PCR. This is done by replacing the hard-thresholding of PCR with soft-thresholding via a penalty function. The result is that both the variance of the predictors and the association with the response of principal components is taken into account during thresholding. This achieves dimension reduction and regression simultaneously as well a regularization to prevent overfitting.

In this paper, we will modify RgPCR to be used for binary classification. To do this we replace the penalized least squares error of RgPCR with a penalized log likelihood function. We approximate the optimal solution by quadratically approximating the log-likelihood function in a way that can be interpretted as a RgPCR problem on weighted "psuedo data". As in RgPCR, for the lasso penalty we have a closed form solution to this problem. By taking iterative quadratic approximations of the above type, we can approximate the of the penalized log likelihood function. The result is a principal component classification algorithm that takes the response into account during variable selection.

In section 2 we will give our motivation and cover the background knowledge necessary. In section 3 we will derive the RgPCC problem and give an algorithm for when the penalty is LASSO. In sections 4 we will compare the performance of RgPCC against other methods on simulated and realworld data. Lastly, in

section 5 we will summarize our conclusions and propose further topics of study. Section 6 is left for figures and tables.

2 Literature Review

RgPCC is a combination of several existing techniques of machine learning. This method improves upon methods of PCA like PCR, uses the likelihood frame works of logistic regression and is ultimately a modification of RgPCR. Here we describe the following methods that have motivate RgPCC. In section 2.1 and 2.2 we summarize PCA and logistic regression respectively. In section 2.3 we motivate and outline RgPCR, a method by Lang and Zou (2).

2.1 Principal Component Analysis (PCA)

The central idea of PCA is to reduce the dimensionality of data given that there is some colinearity between the variables. Our goal is to retain as much of the variance as possible while reducing the dimensionality. Intuitively, we are looking for a linear function α_1 such that $\text{var}(\alpha_1^T \mathbf{X})$ is maximized. This α_1 is then called the *first principal component*, the direction in which the data $\mathbf{X}$ varies the most. We can then find a direction α_2 orthogonal to α_1 that maximizes $\text{var}(\alpha_2^T \mathbf{X})$ for the second principal component and so on.

To simplify the above computation, the above can be recognized as a generalized eigenvalue problem $\max \alpha_1^T \Sigma \alpha_1$ where $\Sigma = \mathbf{X}^T \mathbf{X}$. Thus we can simply compute the spectrum of Σ and or the singular value decomposition of $\mathbf{X}$. Thus, principal components make up the eigenbasis of Σ which are the columns of $\mathbf{V}$ in $\mathbf{X} = \mathbf{UDV}^T$. We thus work with the definition that the k^{th} principal component of $\mathbf{X}$ is the eigenvector v_k of Σ associated to the k^{th} largest eigenvalue. In particular, when $\mathbf{D}$ is written from largest to smallest singular values, then the k^{th} principal component is simply the k^{th} column of $\mathbf{V}$. More information on this definition can be found in (1).

In practice, we wish to use our data projected down to the linear subspace spanned by the first K principal components. This is computed by rearranging the SVD as $\mathbf{XV} = \mathbf{UD}$. In some cases, we may wish to work with $\mathbf{U}$ itself which can be interpretted as a normalized version of the projected data.

2.2 Response-guided Principal Component Regression (RgPCR)

RgPCR is a generalization of ridge regression which we will briefly summarize. Let $\mathbf{X} = \mathbf{UDV}^T$ be the singular value decomposition of an $N \times p$ design matrix

$\mathbf{X}$. Then with response $\mathbf{y}$ we can write the ridge regression solution as

$$\widehat{\boldsymbol{\beta}}^{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda \sum_{j=1}^{p} \beta_j^2. \tag{1}$$

We may rewrite this as

$$\widehat{\boldsymbol{\gamma}}^{\text{ridge}} = \arg\min_{\boldsymbol{\gamma}} ||\mathbf{y} - \mathbf{U}\boldsymbol{\gamma}||_2^2 + \lambda \sum_{j=1}^{p} \frac{\gamma_j^2}{d_j^2}. \tag{2}$$

where $\boldsymbol{\gamma} = \mathbf{D}\mathbf{V}^T\boldsymbol{\beta}$ and d_j is the j^{th} diagonal entry of $\mathbf{D}$. Hence, ridge regression can be viewed as a weighted L_2 penalized regression in the space of principal components. The RgPCR generalization comes from replacing the L_2 penalization with an arbitrary non-decreasing function $p_\lambda(\cdot)$. Some examples of these are LASSO, SCAD and MCP. In general, we write this RgPCR solution as

$$\widehat{\boldsymbol{\gamma}} = \arg\min_{\boldsymbol{\gamma}} ||\mathbf{y} - \mathbf{U}\boldsymbol{\gamma}||_2^2 + \sum_{j=1}^{p} p_\lambda\left(\frac{\gamma_j}{d_j}\right), \tag{3}$$

There are three main advantages to (3). First, the regularization helps prevent overfitting. Secondly, the use of PCA and the orthogonality of U allows for this minimizer to be calculated component-wise. Lastly, the penalty allows the variable selection to be determined by large variance in the predictors and the association with the response of principal components (we can see this by the presence of $\widehat{\gamma}_j^{\text{ols}}$ in the following).

The orthogonality of $\mathbf{U}$ allows us to solve (3) component-wise. In particular,

$$\widehat{\gamma}_j = \arg\min_{\gamma_j} \left(\widehat{\gamma}_j^{\text{ols}} - \gamma_j\right)^2 + p_\lambda\left(\frac{\gamma_j}{d_j}\right). \tag{4}$$

where $\widehat{\gamma}_j^{\text{ols}} = \mathbf{y}^T U_j$ (a calculation can be found in (2)). When $p_\lambda(t) = |t|$, the LASSO penalty, then we can find a closed form of (4),

$$\widehat{\gamma}_j^{\text{lasso}} = \left(|\gamma_j^{\text{ols}}| - \frac{\lambda}{2d_j}\right)^+ \cdot \text{sgn}\left(\widehat{\gamma}_j^{\text{ols}}\right). \tag{5}$$

where a^+ denotes the positive real part of a real number a. We can recover the solution in the original coordinates by

$$\widehat{\boldsymbol{\beta}} = \sum_{j=1}^{p} \widetilde{V}_j \frac{\widehat{\gamma}_j}{d_j} \tag{6}$$

2.3 Logistic Regression

Logistic regression is a common technique for classification. For now we consider binary classification. Let $\mathbf{X}$ be an $N \times p$ design matrix with $\mathbf{y}$ the $N \times 1$ classifications. Then if we assume that

$$\log\left(\frac{Pr(G=1|X=x)}{Pr(G=0|X=x)}\right) = \boldsymbol{\beta}^T x \tag{7}$$

the log-odds are linear, then we may calculate the conditional probability

$$Pr(G=1|X=x) = \frac{\exp(\boldsymbol{\beta}^T x)}{1+\exp(\boldsymbol{\beta}^T x)}. \tag{8}$$

Note that we may compute all probabilities for the data simultaneously by

$$\mathbf{p} = \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}. \tag{9}$$

where p_i is the probability that x_i is in class 1.

To fit such a line for the log-odds (7), we fit by maximizing the log-likelihood,

$$\ell(\mathbf{X},\beta) = \sum_{i=1}^{N} \left[y_i \log(p(x_i;\boldsymbol{\beta})) + (1-y_i)\log(1-p(x_i;\boldsymbol{\beta}))\right] \tag{10}$$

This calculation can be found in (3) and has a convenient interpretation. When maximizing by the Newton-Ralphson method, each iteration can be interpreted as solving a weighted least squares problem. That is

$$\boldsymbol{\beta}^{\text{new}} = \boldsymbol{\beta}^{\text{old}} + (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y}-\mathbf{p}) \tag{11}$$

$$= (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z} \tag{12}$$

where

$$\mathbf{p} = \frac{\exp(\mathbf{X}\boldsymbol{\beta}^{\text{old}})}{1+\exp(\mathbf{X}\boldsymbol{\beta}^{\text{old}})} \tag{13}$$

$$\mathbf{W} = \text{diag}[p_i(1-p_i)] \tag{14}$$

$$\mathbf{z} = \mathbf{X}\boldsymbol{\beta}^{\text{old}} + \mathbf{W}^{-1}(\mathbf{y}-\mathbf{p}). \tag{15}$$

In particular, $\boldsymbol{\beta}^{\text{new}}$ is the solution to a weighted least squares problem with response $\mathbf{z}$, design $\mathbf{X}$ and weights $\mathbf{W}$.

For our purposes, it will be more convenient to state this as the following equivalent problem,

$$\boldsymbol{\beta}^{\text{new}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{W}^{1/2}(\mathbf{z}-\mathbf{X}\boldsymbol{\beta})||^2 \tag{16}$$

3 Response-guided Principal Component Classification (RgPCC)

First, a note on notation. We let p denote the number of predictors in our data, p_λ to a non-decreasing penalty function and p_i to be the i^{th} component of a vector of probabilities $\mathbf{p}$.

Suppose we have a binary classification problem with design $\mathbf{X}$ and classes $\mathbf{y}$. In logistic regression we fit by maximizing the log-likelihood, or equivalently, by minimizing the negative log-likelihood. We propose that we penalize this objective function to get the following problem:

$$\boldsymbol{\beta}^{\text{RgPCC}} = \arg\min_{\boldsymbol{\beta}} -\ell(\mathbf{X}, \boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(\beta_j) \tag{17}$$

3.1 Method

To solve (17), we can approximate the log-likelihood $\ell(\mathbf{X}, \boldsymbol{\beta})$ with a quadratic approximation centered at $\boldsymbol{\beta}^*$ (the minimizer of $-\ell(\mathbf{X}, \boldsymbol{\beta})$). This quadratic approximation can be interpreted as the least squares error

$$-\ell(\mathbf{X}, \boldsymbol{\beta}) \approx ||\mathbf{W}^{1/2}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta})||^2 = ||\widetilde{\mathbf{y}} - \widetilde{\mathbf{X}}\boldsymbol{\beta}||^2 \tag{18}$$

where $\mathbf{W}$, and $\mathbf{z}$ are as in (26 - 28) with (26) evaluated at $\boldsymbol{\beta} = \boldsymbol{\beta}^*$ and $\mathbf{W}^{1/2}\mathbf{X} = \widetilde{\mathbf{X}}$, $\widetilde{\mathbf{y}} = \mathbf{W}^{1/2}\mathbf{z}$. The derivation of this approximation can be found in (4). Note that, we may often refer to $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{y}}$ as the *pseudo data* and *pseudo response* respectively.

Equation (17) can be rewritten as an approximate RgPCR problem

$$\arg\min_{\boldsymbol{\beta}} ||\widetilde{\mathbf{y}} - \widetilde{\mathbf{X}}\boldsymbol{\beta}||^2 + \sum_{j=1}^{p} p_\lambda(\beta_j) \tag{19}$$

For particular choices of p_λ, this may be solved using techniques from (2). That is, by viewing the above in terms of principal components we may solve the equivalent problem

$$\arg\min_{\boldsymbol{\gamma}} ||\widetilde{\mathbf{y}} - \widetilde{\mathbf{U}}\boldsymbol{\gamma}||^2 + \sum_{j=1}^{p} p_\lambda\left(\frac{\gamma_j}{d_j}\right) \tag{20}$$

where $\widetilde{\mathbf{X}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{D}}\widetilde{\mathbf{V}}^T$ is the SVD decomposition of $\widetilde{\mathbf{X}}$ and $\boldsymbol{\gamma} = \widetilde{\mathbf{D}}\widetilde{\mathbf{V}}^T\boldsymbol{\beta}$. The matrices $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{V}}$ are orthogonal matrices and are $N \times p$ and $p \times p$ matrices respectively. Let d_j denotes the diagonal entries of $\widetilde{\mathbf{D}}$ for $1 \leq j \leq p$. For our description here, we assume that $\widetilde{\mathbf{X}}$ has full rank, but the following still holds with a thin SVD decomposition.

As with RgPCR, we may take advantage of the orthogonality of $\widetilde{\mathbf{U}}$ and optimize componentwise. That is, with $\widehat{\gamma}_j^{\text{ols}} = \mathbf{y}^T\widetilde{U}_j$,

$$\arg\min_{\gamma_j}(\widehat{\gamma}_j^{\text{ols}} - \gamma_j)^2 + p_\lambda\left(\frac{\gamma_j}{d_j}\right). \tag{21}$$

From (2), when $p_\lambda(t) = |t|$ is the LASSO penalty, we have the closed form solution of (21)

$$\widehat{\gamma_j} = \left(|\widehat{\gamma}_j^{\text{ols}}| - \frac{\lambda}{d_j}\right)^+ \cdot \text{sgn}\left(\widehat{\gamma}_j^{\text{ols}}\right), \tag{22}$$

We can then rewrite this in the original coordinates using

$$\widehat{\boldsymbol{\beta}} = \sum_{j=1}^{p} \widetilde{V}_j \frac{\widehat{\gamma}_j}{d_j}. \tag{23}$$

3.2 Iterative Approximations

In practice, a single quadratic approximation of $-\ell(\mathbf{X}, \boldsymbol{\beta})$ may not produce a minimum close to the true minimum. Just as in logistic regression, we take the quadratic approximations iteratively to improve our approximation of the minimum. That is, for some approximate minimizer $\boldsymbol{\beta}^{(n)}$, we solve

$$\arg\min_{\boldsymbol{\beta}} -\ell(\mathbf{X}, \boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(\beta_j) \tag{24}$$

which can be approximated by

$$-\ell(\mathbf{X}, \boldsymbol{\beta}) \approx ||\widetilde{\mathbf{y}}^{(n)} - \widetilde{\mathbf{X}}^{(n)}\boldsymbol{\beta}||^2 \tag{25}$$

where

$$\mathbf{p}^{(n)} = \frac{\exp(\mathbf{X}\boldsymbol{\beta}^{(n-1)})}{1+\exp(\mathbf{X}\boldsymbol{\beta}^{(n-1)})} \tag{26}$$

$$\mathbf{W}^{(n)} = \text{diag}[p_i^{(n)}(1-p_i^{(n)})] \tag{27}$$

$$\mathbf{z}^{(n)} = \mathbf{X}\boldsymbol{\beta}^{(n-1)} + (\mathbf{W}^{(n)})^{-1}(\mathbf{y}-\mathbf{p}^{(n)}). \tag{28}$$

$$\widetilde{\mathbf{X}}^{(n)} = (\mathbf{W}^{(n)})^{1/2}\mathbf{X} \tag{29}$$

$$\widetilde{\mathbf{y}}^{(n)} = (\mathbf{W}^{(n)})^{1/2}\mathbf{z}^{(n)} \tag{30}$$

As with the first iteration, we can solve

$$\arg\min_{\boldsymbol{\beta}} ||\widetilde{\mathbf{y}}^{(n)} - \widetilde{\mathbf{X}}^{(n)}\boldsymbol{\beta}||^2 + \sum_{j=1}^{p} p_\lambda(\beta_j) \tag{31}$$

to estimate the solution to (25). Then, as with the first iteration, we can express the above using principal components

$$\arg\min_{\boldsymbol{\gamma}} ||\widetilde{\mathbf{y}}^{(n)} - \widetilde{\mathbf{U}}^{(n)}\boldsymbol{\gamma}||^2 + \sum_{j=1}^{p} p_\lambda\left(\frac{\gamma_j}{d_j^{(n)}}\right) \tag{32}$$

and take advantage of the orthogonality of $\widetilde{\mathbf{U}}$ so solve componentwise

$$\arg\min_{\gamma_j} (\widehat{\gamma}_j^{\text{ols},(n)} - \gamma_j)^2 + p_\lambda\left(\frac{\gamma_j}{d_j^{(n)}}\right). \tag{33}$$

3.3 Algorithm for LASSO penalty

Here, we will briefly describe the algorithm used to approximate a solution to (17) when we use the LASSO penalty. This will generalize to any penalty that has an RgPCR closed form solution.

(Step 1) Guess an initial $\widehat{\beta}^{(0)}$

(Step 2) With $\widehat{\beta}^{(0)}$ calculate the following

$$\begin{aligned}
\mathbf{p}^{(1)} &= \frac{\exp(\mathbf{X}\boldsymbol{\beta}^{(0)})}{1+\exp(\mathbf{X}\boldsymbol{\beta}^{(0)})} = \frac{\exp(\mathbf{U}\boldsymbol{\gamma}^{(0)})}{1+\exp(\mathbf{U}\boldsymbol{\gamma}^{(0)})} \\
\mathbf{W}^{(1)} &= \operatorname{diag}[p_i^{(1)}(1-p_i^{(1)})] \\
\mathbf{z}^{(1)} &= \mathbf{X}\boldsymbol{\beta}^{(0)} + (\mathbf{W}^{(1)})^{-1}(\mathbf{y}-\mathbf{p}^{(1)}) \\
\widetilde{\mathbf{X}}^{(1)} &= (\mathbf{W}^{(1)})^{1/2}\mathbf{X} \\
\widetilde{\mathbf{y}}^{(1)} &= (\mathbf{W}^{(1)})^{1/2}\mathbf{z}^{(1)}
\end{aligned}$$

(Step 3) Compute the solution to the RgPCR problem

$$\arg\min_{\boldsymbol{\gamma}} ||\widetilde{\mathbf{y}}^{(0)} - \widetilde{\mathbf{U}}^{(0)}\boldsymbol{\gamma}||^2 + \lambda \sum_{j=1}^{p} \left| \frac{\gamma_j}{d_j^{(0)}} \right| \tag{34}$$

using the closed form solution

$$\widehat{\gamma_j}^{(1)} = \left(|\widehat{\gamma}_j^{\text{ols}}| - \frac{\lambda}{d_j} \right)^+ \cdot \operatorname{sgn}\left(\widehat{\gamma}_j^{\text{ols}}\right), \tag{35}$$

where $\widehat{\gamma}_j^{\text{ols}} = \widetilde{y}^{(1)}\widetilde{U}_j^{(1)}$

(Step 4) repeat steps 2 and 3 using $\widehat{\boldsymbol{\gamma}}^{(1)}$ and similarly using $\widehat{\boldsymbol{\gamma}}^{(k)}$ until $\varepsilon^{(k)} = \frac{||\boldsymbol{\beta}^{(k)}-\boldsymbol{\beta}^{(k+1)}||}{||\boldsymbol{\beta}^{(k)}||} < \varepsilon$ for some desired tolerance ε.

(Step 5) Recover the solution in the original coordinates with

$$\widehat{\boldsymbol{\beta}}^{(n)} = \sum_{j=1}^{p} \widetilde{V}_j^{(n)} \frac{\widehat{\gamma}_j^{(n)}}{d_j^{(n)}}.$$

4 Numerical Studies

In this section we compare RgPCC with LASSO penalty to conventional logistic regression and other classification methods. This section is meant to be a "proof of concept" for the RgPCC method. At the very least it shows that this method is comparable to other methods used in term so prediction performation and probability density estimation.

In section 4.1 we discuss parameters of our experiments and why we chose such parameters. In sections 4.2 and 4.3 we presented the simulated and realworld experiments respectively and discuss the results of those experiments and how we measured the performance of each method.

4.1 Parameters of Experiments

We will consider a variety of sample sizes, dimensions and sparsities in $\boldsymbol{\gamma}$ and compare error rates in $\mathbf{p}$ and classification. We will also vary which principal components of $\boldsymbol{\gamma}$ are nonzero. The following gives the parameters for our simulated data:

p	N	$\boldsymbol{\gamma}$ sparsity
12	100, 200	1, 3, 5
50	100, 200	1, 3
80	200, 300	1, 3

Figure 1: Parameters of data creation.

The convergence of our algorithm is based on the tolerance of $\varepsilon = 0.1$, which means we will terminate our Newton-Ralphson method when

$$\varepsilon^{(k)} = \frac{||\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^{(k+1)}||}{||\boldsymbol{\beta}^{(k)}||} < 0.1.$$

4.2 Creation of Simulated Data

Let $\mathbf{X}$ be our design matrix where each row of $\mathbf{X}$ is independently generated from $N(0, \Sigma)$ where $\Sigma_{ij} = \rho^{|i-j|}$ with $\rho = 0.8$.

To generate our response data, let $\mu = \mathbf{X}\boldsymbol{\beta}^* = \mathbf{U}\boldsymbol{\gamma}^*$ where $\boldsymbol{\gamma}^* = \mathbf{D}\mathbf{V}^T\boldsymbol{\beta}^*$. Then the classes $\mathbf{y}$ come from a Bernoulli distribution with parameter $\mathbf{p} = \frac{\exp(\mathbf{U}\boldsymbol{\gamma}^*)}{1+\exp(\mathbf{U}\boldsymbol{\gamma}^*)} = \frac{\exp(\mu)}{1+\exp(\mu)}$. Therefore the response data is dependent on $\boldsymbol{\gamma}^*$ and the data $\mathbf{X}$. We make a variety of choices for $\boldsymbol{\gamma}^*$ (denoted just as $\boldsymbol{\gamma}$ below) which have varying amounts of sparsity and created varying levels of linear separability.

4.3 Performance of RgPCC on Simulated Data

Here we fit RgPCC, logistic regression and PC logistic regression to our simulated data. To test the quality of the fit, we see how well our models predict the true probability vector $\mathbf{p}$ (the parameter of the Bernoulli distribution above) over a test set of size $5N$. We measure the accuracy of our predicted $\widehat{\mathbf{p}}$ using the 1-norm, 2-norm and EGKL. These are defined as follows:

$$
\begin{aligned}
\text{1-norm error} \qquad & \sum_{i=1}^{N} |\widehat{\mathbf{p}}_i - \mathbf{p}_i| \\
\text{2-norm error} \qquad & \sum_{i=1}^{N} (\widehat{\mathbf{p}}_i - \mathbf{p}_i)^2 \\
\text{EGKL loss} \qquad & \sum_{i=1}^{N} \mathbf{p}_i \log \left(\frac{\mathbf{p}_i}{\widehat{\mathbf{p}}_i} \right)
\end{aligned}
$$

We fit the models using samples of size N for dimension p from Figure 1. We also vary the nonzero components of the true $\boldsymbol{\gamma}$. We neglect the case where $N < p$, although there are no theoretical issues in this case and will be covered in further research.

We use the following values for the true $\boldsymbol{\gamma}$:

$$
\begin{aligned}
\boldsymbol{\gamma}_1 &= (25, 0, \ldots, 0) \\
\boldsymbol{\gamma}_2 &= (20, 10, 10, 0, \ldots, 0) \\
\boldsymbol{\gamma}_3 &= (15, 10, 5, 5, 3, 0, \ldots, 0) \\
\boldsymbol{\gamma}'_1 &= (0, 0, 0, 0, 0, 25, 0, \ldots, 0) \\
\boldsymbol{\gamma}'_2 &= (0, 0, 0, 0, 0, 20, 10, 10, 0, \ldots, 0) \\
\boldsymbol{\gamma}'_3 &= (0, 0, 0, 0, 0, 15, 10, 5, 5, 3, 0, \ldots, 0)
\end{aligned}
$$

Below we show only a few cases of all the tests. The rest of the results are present in the last section but have the same conclusion as the one presented here.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	128.9002	48.2978	135.2297	0.3776	12
2	pcalog	100	99.0854	28.8548	38.2041	0.4807	5.14
3	RgPCC.AIC	100	117.0917	41.3108	92.3249	0.3775	10.78
4	RgPCC.BIC	100	77.461	20.1746	28.3024	0.3784	6.19
5	RgPCC.MSE	100	80.1013	18.9059	25.2497	0.3936	1.97
6	RgPCC.pMSE	100	76.0027	17.8124	23.8169	0.3849	2.81
7	RgPCC.MSECV	100	128.7374	48.1967	134.5861	0.3776	12
8	log	200	189.2017	49.8042	87.5984	0.389	12
9	pcalog	200	151.9253	34.7084	42.483	0.4834	5.16
10	RgPCC.AIC	200	177.856	44.8999	73.7718	0.3891	11.17
11	RgPCC.BIC	200	101.4748	17.4053	19.2933	0.3887	5.53
12	RgPCC.MSE	200	118.2799	22.4926	26.6875	0.386	7.63
13	RgPCC.pMSE	200	94.8438	15.3939	16.9056	0.3907	4.25
14	RgPCC.MSECV	200	155.4774	35.7876	51.6239	0.3893	10.2

Figure 2: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_2'$ and $p = 12$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	219.7315	119.8399	3337.828	0.4469	50
2	pcalog	100	147.4434	62.2117	241.1183	0.412	16.61
3	RgPCC.AIC	100	168.0875	77.802	288.5035	0.4282	32.88
4	RgPCC.BIC	100	88.2784	25.8719	37.0038	0.4077	5.7
5	RgPCC.MSE	100	86.9706	25.1717	35.639	0.4087	5.55
6	RgPCC.pMSE	100	86.0024	24.6228	34.8048	0.4066	5.33
7	RgPCC.MSECV	100	95.7481	30.1532	46.0895	0.4088	8.06
8	log	200	295.6583	112.8611	392.142	0.4326	50
9	pcalog	200	204.7182	58.4586	107.592	0.4175	18.21
10	RgPCC.AIC	200	148.4187	33.7531	45.4301	0.4129	10.64
11	RgPCC.BIC	200	127.0452	25.9331	31.4919	0.4123	7.18
12	RgPCC.MSE	200	174.8136	44.7253	68.5684	0.4161	19.38
13	RgPCC.pMSE	200	127.0452	25.9331	31.4919	0.4123	7.18
14	RgPCC.MSECV	200	127.0452	25.9331	31.4919	0.4123	7.18

Figure 3: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_2'$ and $p = 50$.

Overall, we see that RgPCC provides a slight improvement in the testing classification error compared to logistic regression and PC logistic regression. However, the real improvement is in the accuracy of our prediction of $\mathbf{p}$.

Overall, we can see that RgPCC provides slight improvements in the classification error to logistic regression and principal component logistic regression when the true $\boldsymbol{\gamma}$ is nonzero in the leading principal components. However, when the nonzero components of $\boldsymbol{\gamma}$ are not the leading principal components we see a significant different between principal component logistic regression and our other methods.

We can see that the testing error for $\mathbf{p}$ is improved with RgPCC by at least a factor of 4 in each case. RgPCC avoids overfitting the data as much as logistic regression does. In particular, there is drastic improvement in classification when $\boldsymbol{\gamma}^*$ is very sparse (that is for $\boldsymbol{\gamma}_1$). Similar results occur when $N = 200$ but are left in the table and results section.

We also run a few examples for higher dimensional data. Our method should excel in this via its use of principal components and sparsity in $\boldsymbol{\gamma}$.

where

$$\boldsymbol{\gamma}_1 = (25, 0, 0, 0, 0, 0, \ldots, 0)$$
$$\boldsymbol{\gamma}_2 = (10, 5, 5, 0, 0, 0, \ldots, 0)$$

Here we see similar results as in the low dimensional case. At this point, we know that RgPCC performs better, if no just as well, as logistic regression does. However, there are still many aspect of the method we wish to study.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	200	445.8727	228.9881	6086.0289	0.4572	80
2	pcalog	200	243.8371	85.2713	162.1112	0.429	27.26
3	RgPCC.AIC	200	176.0586	49.8262	66.7715	0.4234	7.76
4	RgPCC.BIC	200	163.8553	44.4008	57.1169	0.4231	5.64
5	RgPCC.MSE	200	182.8881	53.0481	73.0579	0.4238	10.16
6	RgPCC.pMSE	200	151.7795	38.8976	48.2282	0.4222	4.8
7	RgPCC.MSECV	200	151.7795	38.8976	48.2282	0.4222	4.8
8	log	300	466.9288	184.6317	574.1098	0.4544	80
9	pcalog	300	302.701	87.0586	150.5586	0.4426	28.68
10	RgPCC.AIC	300	199.9717	43.4651	56.9527	0.4361	5.96
11	RgPCC.BIC	300	192.9372	40.9562	52.309	0.4367	5.36
12	RgPCC.MSE	300	213.3833	48.2201	65.9976	0.4374	8.7
13	RgPCC.pMSE	300	192.9372	40.9562	52.309	0.4367	5.36
14	RgPCC.MSECV	300	194.8935	41.7434	53.6679	0.437	5.54

Figure 4: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_2$ and $p = 80$.

4.4 Performance of RgPCC on Realworld Data

In this section we apply four real datasets to compare the prediction performance of RgPCC with conventional binary classification methods. We look at the following data sets. These data sets can be found in the UCI data repository and are cited in the references.

- Audit (777 instances, 18 predictors, binary response) This data set contains information about the risk factors of fraud firms. The response is whether or not the firm has committed fraud. (5)
- Cryotherapy (90 instances, 7 predictors, binary response) This data set contains information about the use of cryotherapy to treat a variety of warts. The response is whether or not the treatment was successful. (6)
- Divorce Predictors (170 instances, 54 predictors, binary response) This data set contains numerical answers from a questionnaire to individuals regarding their marriage. The response is whether or not the marriage ended in divorce. (7)
- Ecoli (336 instances, 8 predictors, binary response) This data set contains information about ecoli signal sequencing. The response is the localization site of the ecoli is cytoplasm or not. (8)

We will compare the methods RgPCC, logistic, PCA logistic and ridge regression. We will tune RgPCC by AIC and BIC. Thus we will compare a total of five different methods.

To compare the performance of our binary classification methods based on the error of each fold of 5-fold cross validation. We then perform Tukey's multiple pairwise comparison to test if the methods differ significantly or not. We also compare these methods as probability density estimators and use the loglikelihood as our metric.

The following is a summary of the error results, more can be found in the section 6.

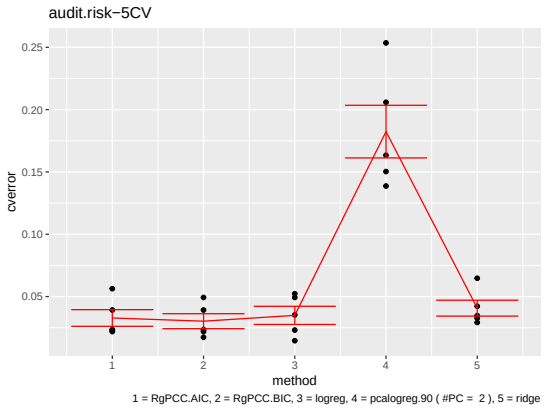
	RgPCC AIC	RgPCC BIC	Logistic	PC Logistic	Ridge
Audit	0.033548	0.030967	0.034838	0.18193	0.04129
Cryo	0.12222	0.12222	0.12222	0.5	0.14444
Divorce	0.023529	0.023529	0.058823	0.035294	0.023529
Ecoli	0.041666	0.038690	0.041666	0.053571	0.044642

Figure 5: The mean classification errors over each of the 5 folds.

For each data set, we see that the errors are very similar, except for PC logistic regression. We confirm this with a Tukey's multiple pairwise comparision.

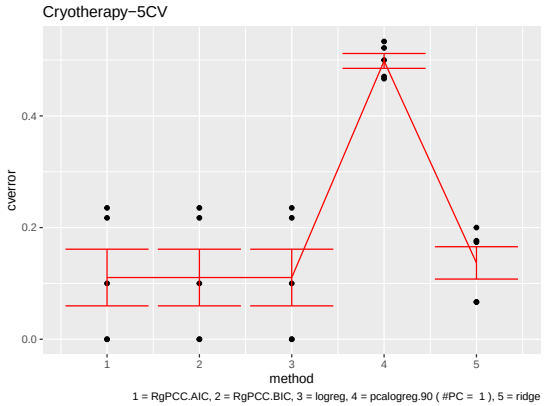
During this comparison, we noticed that in most cases the difference in the errors was insignificantly different with p-values around $0.99 \geq \alpha = 0.05$. However, there were a few cases in which PC logistic was significantly different than all the other methods (Audit data and Cryotherapy). This data is given below. Tables for the results of Tukey's multiple pairwise comparison can be found in section 6.

As a probability density estimator, RgPCC outperforms logistic, PC logistic and Ridge in all of the realworld data sets. In particular, we see that the log-likelihood for RgPCC is half as much as the next best method (Ridge) for the divorce data (??) and around a tenth as much as the worst method (logistic).



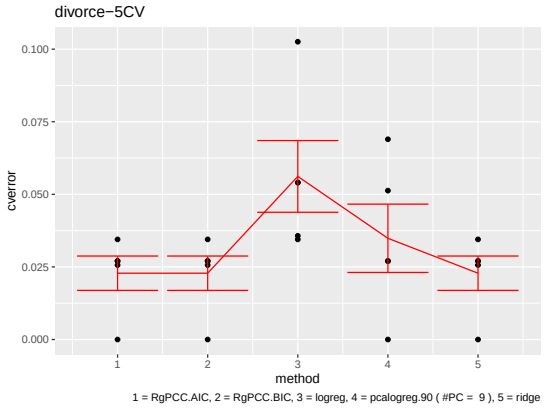
	RgPCC.AIC	RgPCC.BIC	logistic	PCA.logistic	Ridge
loglike	−83.6686158378557	−82.6653217108159	−150.591033919281	−301.205708128731	−118.64877430207

Figure 6: Results from analyzing audit risk data



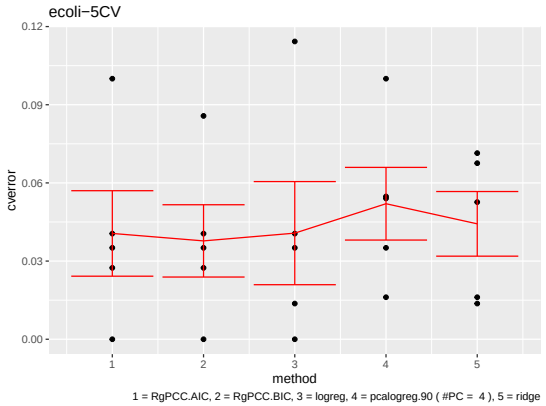
	RgPCC.AIC	RgPCC.BIC	logistic	PCA.logistic	Ridge
loglike	−35.2419137159939	−35.2419137159939	−35.6471021737713	−66.8959089939404	−44.8856308591776

Figure 7: Results from analyzing cryotherapy data



	RgPCC.AIC	RgPCC.BIC	logistic	PCA.logistic	Ridge
loglike	−17.9720663442284	−15.0211838327529	−147.40466225644	−137.736014499879	−42.8530954379232

Figure 8: Results from analyzing divorce survey data



	RgPCC.AIC	RgPCC.BIC	logistic	PCA.logistic	Ridge
loglike	−56.4361262492897	−55.0614092483722	−84.7823361409379	−61.5128727783695	−73.9074965364359

Figure 9: Results from analyzing ecoli data

Thus, in terms of classification error RgPCC is comparable to logistic and ridge regression in application settings. As a probability density estimator we see a more significant improvement.

5 Conclusions and Further Research

In this section we give some conclusions and mention topics for further research.

5.1 Conclusions

In the above we have proposed response-guided principal component classification (RgPCC) as alternative to principal component classification and logistic regression. We have illustrated its practice on simulated and real data as a classification method and a probability density esitmator with reasonable evidence of its better performance.

We have limited the above to the lasso penalty for our thresholding rule as an initial "proof of concept" for this method. In further research we plan to vary the thresholding function or even use data to inform the choice of thresholding function.

5.2 Further Research

5.2.1 Speed

This method of RgPCC improves upon testing error in binary classification compared to logistic regression. We also believe that this use of principal component analysis should improve the speed of the computation, especially when there is sparsity in the solution with respect to the principal components. We wish to further explore this, especially in higher dimensions and when $p > N$.

5.2.2 C_p-type Statistic for Parameter Tuning

Using cross-validation for parameter selection is data and time intensive. In Zou and Lang's paper (2), the regularization parameter in RgPCR can be selected using Stein's unbiased risk estimation. In later works, we will also investigate if this method can be generalized to RgPCC.

5.2.3 Further Generalization

In many cases, data is not presented in a manner such that classes are linearly separable. Kernel methods embed data in higher dimensional spaces in a way that makes classes linearly separable, without have to work in the higher dimensional space directly. We would like to be able to apply RgPCC in these nonlinear cases as well.

We would also like to extend this to multiclass classification through ideas from logistic regression for multiclass classification. We can also take the ideas from (4) to generalize this method to other classifiers such as SVM.

6 Figures

6.1 Tables of Simulated Results

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	91.4193	22.4534	84.183	0.3085	12
2	pcalog	100	70.2857	12.9409	41.9257	0.2953	5.14
3	RgPCC.AIC	100	52.2256	7.2499	20.5037	0.2887	3.25
4	RgPCC.BIC	100	37.4527	3.9344	10.6117	0.2862	1.89
5	RgPCC.MSE	100	13.0542	0.5737	0.6827	0.2836	1
6	RgPCC.pMSE	100	47.7924	6.1286	17.0532	0.2875	2.61
7	RgPCC.MSECV	100	13.3021	0.5916	0.9724	0.2836	1.01

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	200	149.961	28.9641	48.8889	0.3612	12
2	pcalog	200	128.5254	21.1671	32.6773	0.3582	5.16
3	RgPCC.AIC	200	105.4343	14.3677	18.7833	0.3596	2.92
4	RgPCC.BIC	200	85.579	9.6329	10.1515	0.3591	1.42
5	RgPCC.MSE	200	31.7527	1.7369	4.308	0.3591	1
6	RgPCC.pMSE	200	91.6193	10.9608	12.4709	0.3594	1.62
7	RgPCC.MSECV	200	65.8826	5.9728	4.6136	0.3591	1.05

Figure 10: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_1$ and $p = 12$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	210.0333	111.4804	3043.972	0.4216	50
2	pcalog	100	120.832	42.9534	133.9528	0.3665	16.61
3	RgPCC.AIC	100	80.3707	22.2387	35.1172	0.3609	7.22
4	RgPCC.BIC	100	63.9284	15.7583	19.9299	0.358	3.93
5	RgPCC.MSE	100	58.5378	12.4238	12.4689	0.3586	2.02
6	RgPCC.pMSE	100	62.5865	14.9977	18.4198	0.3569	3.7
7	RgPCC.MSECV	100	58.6641	13.2569	14.5907	0.356	2.85

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	200	292.2532	111.7781	327.7571	0.4232	50
2	pcalog	200	195.8345	55.9057	96.0681	0.4022	18.21
3	RgPCC.AIC	200	145.4733	34.351	46.6281	0.394	6.81
4	RgPCC.BIC	200	120.4454	25.6383	31.3403	0.3921	3.52
5	RgPCC.MSE	200	137.4826	31.367	41.0866	0.3929	5.38
6	RgPCC.pMSE	200	107.4107	21.7626	25.4853	0.3941	2.85
7	RgPCC.MSECV	200	107.4107	21.7626	25.4853	0.3941	2.85

Figure 11: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_1$ and $p = 50$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	200	447.4376	230.4442	6306.2512	0.4565	80
2	pcalog	200	244.6892	86.3616	163.7597	0.4287	27.21
3	RgPCC.AIC	200	174.8162	50.1733	67.5138	0.4228	8.12
4	RgPCC.BIC	200	142.1117	35.8792	44.0206	0.4223	3.91
5	RgPCC.MSE	200	174.8162	50.1733	67.5138	0.4228	8.12
6	RgPCC.pMSE	200	143.7075	36.5861	44.9593	0.4225	4.22
7	RgPCC.MSECV	200	150.7253	39.6019	49.4351	0.422	4.87

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	300	452.0082	173.9669	515.7646	0.4522	80
2	pcalog	300	290.4376	79.9708	138.9784	0.438	28.67
3	RgPCC.AIC	300	200.8527	42.5285	58.9005	0.4326	8.19
4	RgPCC.BIC	300	170.7052	32.663	40.9651	0.4326	4.43
5	RgPCC.MSE	300	126.9332	20.117	20.1521	0.4326	2.24
6	RgPCC.pMSE	300	126.9332	20.117	20.1521	0.4326	2.24
7	RgPCC.MSECV	300	182.119	36.2148	47.0899	0.4324	5.34

Figure 12: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_1$ and $p = 80$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	121.1215	42.8983	118.2832	0.3733	12
2	pcalog	100	99.1586	28.8846	41.8466	0.4808	5.14
3	RgPCC.AIC	100	120.9643	42.8073	117.6959	0.3733	12
4	RgPCC.BIC	100	64.7351	14.3025	20.2286	0.3659	3.3
5	RgPCC.MSE	100	81.4171	18.8461	25.7643	0.3684	1.08
6	RgPCC.pMSE	100	64.7351	14.3025	20.2286	0.3659	3.3
7	RgPCC.MSECV	100	64.7351	14.3025	20.2286	0.3659	3.3
8	log	200	178.524	44.304	80.5032	0.377	12
9	pcalog	200	156.3044	35.6168	46.5895	0.4799	5.16
10	RgPCC.AIC	200	178.2843	44.208	80.226	0.377	12
11	RgPCC.BIC	200	77.339	10.6363	12.1643	0.3736	2.58
12	RgPCC.MSE	200	73.6894	9.0733	11.2684	0.3712	1.53
13	RgPCC.pMSE	200	138.6472	27.8215	35.5525	0.4341	0.55
14	RgPCC.MSECV	200	73.6894	9.0733	11.2684	0.3712	1.53

Figure 13: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_1'$ and $p = 12$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	217.994	118.5425	3297.8675	0.4435	50
2	pcalog	100	144.7081	60.4268	248.417	0.4101	16.61
3	RgPCC.AIC	100	97.6191	31.6963	52.565	0.4056	9.03
4	RgPCC.BIC	100	78.4488	20.9776	29.8746	0.4054	3
5	RgPCC.MSE	100	84.067	24.5299	36.75	0.4053	5
6	RgPCC.pMSE	100	76.4405	18.8724	26.5078	0.4084	1.37
7	RgPCC.MSECV	100	97.6191	31.6963	52.565	0.4056	9.03
8	log	200	289.9151	109.1209	358.0482	0.4209	50
9	pcalog	200	198.285	55.4996	105.0613	0.4024	18.21
10	RgPCC.AIC	200	153.1185	35.4864	51.9848	0.3996	11.63
11	RgPCC.BIC	200	95.1742	15.8074	18.1015	0.3968	2.44
12	RgPCC.MSE	200	89.1983	13.9244	15.8187	0.3964	1.85
13	RgPCC.pMSE	200	90.021	13.1113	16.0325	0.395	1.02
14	RgPCC.MSECV	200	88.5974	13.3567	15.6131	0.3966	1.27

Figure 14: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_1'$ and $p = 50$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	91.4193	22.4534	84.183	0.3085	12
2	pcalog	100	70.2857	12.9409	41.9257	0.2953	5.14
3	RgPCC.AIC	100	52.2256	7.2499	20.5037	0.2887	3.25
4	RgPCC.BIC	100	37.4527	3.9344	10.6117	0.2862	1.89
5	RgPCC.MSE	100	23.214	1.6237	4.2134	0.2849	1.33
6	RgPCC.pMSE	100	47.7924	6.1286	17.0532	0.2875	2.61
7	RgPCC.MSECV	100	19.0432	1.1745	2.8072	0.2851	1.17
8	log	200	149.961	28.9641	48.8889	0.3612	12
9	pcalog	200	128.5254	21.1671	32.6773	0.3582	5.16
10	RgPCC.AIC	200	102.1291	13.5034	17.1095	0.3593	2.48
11	RgPCC.BIC	200	89.1385	10.4045	11.4629	0.3594	1.48
12	RgPCC.MSE	200	72.2776	7.0366	6.0998	0.359	1.15
13	RgPCC.pMSE	200	93.8281	11.4705	13.371	0.3594	1.75
14	RgPCC.MSECV	200	70.1922	6.6682	5.5661	0.3591	1.11

Figure 15: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_2$ and $p = 12$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	210.0333	111.4804	3043.972	0.4216	50
2	pcalog	100	120.832	42.9534	133.9528	0.3665	16.61
3	RgPCC.AIC	100	80.3707	22.2387	35.1172	0.3609	7.22
4	RgPCC.BIC	100	63.9284	15.7583	19.9299	0.358	3.93
5	RgPCC.MSE	100	61.5187	14.6671	17.5473	0.3571	3.55
6	RgPCC.pMSE	100	62.5865	14.9977	18.4198	0.3569	3.7
7	RgPCC.MSECV	100	58.6641	13.2569	14.5907	0.356	2.85
8	log	200	292.2532	111.7781	327.7571	0.4232	50
9	pcalog	200	195.8345	55.9057	96.0681	0.4022	18.21
10	RgPCC.AIC	200	147.2218	34.9867	47.8487	0.3944	7.27
11	RgPCC.BIC	200	119.3539	25.3686	30.8969	0.392	3.46
12	RgPCC.MSE	200	137.4826	31.367	41.0866	0.3929	5.38
13	RgPCC.pMSE	200	117.0069	24.5538	29.5989	0.3919	3.47
14	RgPCC.MSECV	200	132.7294	29.7291	38.2034	0.3925	4.52

Figure 16: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_2$ and $p = 50$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	128.9002	48.2978	135.2297	0.3776	12
2	pcalog	100	99.0854	28.8548	38.2041	0.4807	5.14
3	RgPCC.AIC	100	117.0917	41.3108	92.3249	0.3775	10.78
4	RgPCC.BIC	100	77.461	20.1746	28.3024	0.3784	6.19
5	RgPCC.MSE	100	80.1013	18.9059	25.2497	0.3936	1.97
6	RgPCC.pMSE	100	76.0027	17.8124	23.8169	0.3849	2.81
7	RgPCC.MSECV	100	128.7374	48.1967	134.5861	0.3776	12
8	log	200	189.2017	49.8042	87.5984	0.389	12
9	pcalog	200	151.9253	34.7084	42.483	0.4834	5.16
10	RgPCC.AIC	200	177.856	44.8999	73.7718	0.3891	11.17
11	RgPCC.BIC	200	101.4748	17.4053	19.2933	0.3887	5.53
12	RgPCC.MSE	200	118.2799	22.4926	26.6875	0.386	7.63
13	RgPCC.pMSE	200	94.8438	15.3939	16.9056	0.3907	4.25
14	RgPCC.MSECV	200	155.4774	35.7876	51.6239	0.3893	10.2

Figure 17: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_2'$ and $p = 12$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	219.7315	119.8399	3337.828	0.4469	50
2	pcalog	100	147.4434	62.2117	241.1183	0.412	16.61
3	RgPCC.AIC	100	168.0875	77.802	288.5035	0.4282	32.88
4	RgPCC.BIC	100	88.2784	25.8719	37.0038	0.4077	5.7
5	RgPCC.MSE	100	86.9706	25.1717	35.639	0.4087	5.55
6	RgPCC.pMSE	100	86.0024	24.6228	34.8048	0.4066	5.33
7	RgPCC.MSECV	100	95.7481	30.1532	46.0895	0.4088	8.06
8	log	200	295.6583	112.8611	392.142	0.4326	50
9	pcalog	200	204.7182	58.4586	107.592	0.4175	18.21
10	RgPCC.AIC	200	148.4187	33.7531	45.4301	0.4129	10.64
11	RgPCC.BIC	200	127.0452	25.9331	31.4919	0.4123	7.18
12	RgPCC.MSE	200	174.8136	44.7253	68.5684	0.4161	19.38
13	RgPCC.pMSE	200	127.0452	25.9331	31.4919	0.4123	7.18
14	RgPCC.MSECV	200	127.0452	25.9331	31.4919	0.4123	7.18

Figure 18: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_2'$ and $p = 50$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	92.8424	22.5978	63.6225	0.3575	12
2	pcalog	100	73.1611	13.9248	35.3714	0.3494	5.14
3	RgPCC.AIC	100	59.2527	9.6751	21.5075	0.3501	5.85
4	RgPCC.BIC	100	44.1646	5.9479	11.666	0.3488	3.64
5	RgPCC.MSE	100	41.6895	5.448	10.1906	0.3481	3.34
6	RgPCC.pMSE	100	52.3885	7.8156	16.6285	0.349	4.73
7	RgPCC.MSECV	100	59.2527	9.6751	21.5075	0.3501	5.85
8	log	200	139.2422	25.9081	39.7457	0.3904	12
9	pcalog	200	118.455	18.7518	27.0324	0.3888	5.16
10	RgPCC.AIC	200	100.1149	13.7903	16.1594	0.3863	6.08
11	RgPCC.BIC	200	79.0253	9.1298	6.5746	0.3828	3.78
12	RgPCC.MSE	200	93.7817	12.2363	13.0377	0.3849	5.24
13	RgPCC.pMSE	200	93.7817	12.2363	13.0377	0.3849	5.24
14	RgPCC.MSECV	200	108.7195	16.0981	20.6227	0.3868	7.36

Figure 19: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_3$ and $p = 12$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	220.9788	114.6394	3348.697	0.4446	50
2	pcalog	100	122.883	42.1172	106.2969	0.3977	16.61
3	RgPCC.AIC	100	77.3434	19.528	29.0651	0.3859	6.62
4	RgPCC.BIC	100	64.2328	14.6446	18.8655	0.387	4.27
5	RgPCC.MSE	100	58.8031	11.9666	12.2717	0.3936	2.82
6	RgPCC.pMSE	100	58.0424	11.404	11.363	0.3916	2.41
7	RgPCC.MSECV	100	71.5654	17.3656	24.3613	0.386	5.55
8	log	200	285.8948	105.8863	274.7502	0.4368	50
9	pcalog	200	187.6784	50.8358	85.5098	0.4154	18.21
10	RgPCC.AIC	200	127.3539	26.6455	36.1314	0.4087	5.99
11	RgPCC.BIC	200	105.2677	19.4904	24.2934	0.4075	3.87
12	RgPCC.MSE	200	99.5549	17.8544	21.9139	0.4072	3.51
13	RgPCC.pMSE	200	105.2677	19.4904	24.2934	0.4075	3.87
14	RgPCC.MSECV	200	119.7252	24.1306	31.7303	0.4081	5.07

Figure 20: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_3$ and $p = 50$.

	method	sample_size	one.norm	two.norm	EGKL	class.error	gamma.size
1	log	100	124.1939	43.2941	104.1191	0.411	12
2	pcalog	100	87.3151	23.0029	32.9166	0.4904	5.14
3	RgPCC.AIC	100	112.1846	36.5856	74.3954	0.4115	10.86
4	RgPCC.BIC	100	65.2582	13.808	20.5513	0.408	4.19
5	RgPCC.MSE	100	69.4398	14.3246	21.6984	0.4268	2.01
6	RgPCC.pMSE	100	64.9672	13.5437	20.2168	0.4087	3.69
7	RgPCC.MSECV	100	100.1533	30.2561	53.6707	0.4107	10.05
8	log	200	176.8593	43.9783	67.9536	0.4118	12
9	pcalog	200	130.3992	26.1145	31.8081	0.4892	5.16
10	RgPCC.AIC	200	153.8135	34.595	47.6248	0.4116	10.5
11	RgPCC.BIC	200	93.559	14.5554	15.6513	0.4093	5.91
12	RgPCC.MSE	200	153.8135	34.595	47.6248	0.4116	10.5
13	RgPCC.pMSE	200	84.4838	11.8632	13.0448	0.413	4
14	RgPCC.MSECV	200	176.5554	43.845	67.6322	0.4118	12

Figure 21: Simulated results for RgPCC with parameters $\boldsymbol{\gamma}_3'$ and $p = 12$.

6.2 Graphs of Tuning

References

[1] Jolliffe IT. Principal component analysis. Springer; 2011.

[2] Lang W, Zou H. A simple method to improve principal components regression. Stat. 2020;9(1).

[3] Hastie T, Friedman J, Tisbshirani R. The Elements of statistical learning: data mining, inference, and prediction. Springer; 2017.

[4] Wang H, Leng C. Unified LASSO Estimation by Least Squares Approximation. Journal of the American Statistical Association. 2007;102(479):1039–1048.

[5] UCI Machine Learning Repository: Audit Data Data Set;. Available from: https://archive.ics.uci.edu/ml/datasets/AuditData.

[6] UCI Machine Learning Repository: Cryotherapy Dataset Data Set;. Available from: https://archive.ics.uci.edu/ml/datasets/CryotherapyDataset.

[7] UCI Machine Learning Repository: Divorce Predictors data set Data;. Available from: https://archive.ics.uci.edu/ml/datasets/DivorcePredictorsdataset.

[8] UCI Machine Learning Repository: Ecoli Data Set;. Available from: https://archive.ics.uci.edu/ml/datasets/Ecoli.