Supervised Ridge Regression in High Dimensional Linear Regression

ZHU, Xiangchen

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy

in

Information Engineering

The Chinese University of Hong Kong
July 2013

Abstract of thesis entitled: Supervised Ridge Regression for High Dimensional Linear Regression

In the field of statistical learning, we usually have a lot of features to determine the behavior of some response. For example in gene testing problems we have tons of genes as features and their relations with certain disease need to be determined. Without specific knowledge available, the most simple and fundamental way to model this kind of problem would be a linear model. There are tons of existing methods to solve linear regression, like conventional ordinary least squares, ridge regression and LASSO (least absolute shrinkage and selection operator). Let N denote the number of samples and p denote the number of predictors, in ordinary settings where we have enough samples (N > p),ordinary linear regression methods like ridge will usually give reasonable predictions for the future values of the response. In the development of modern statistical learning, it's quite often that we meet high dimensional problems (N << p), like microarray data testing problems. In these kinds of high-dimensional problems it is generally quite difficult to identify the relationship between the predictors and the response without any further assumptions.

Despite the fact that tons of predictors exist for prediction, most of the predictors are actually spurious in a lot of real problems. A predictor being spurious means that it is not directly related to the response. Conventional techniques in linear regression like LASSO and ridge both have their limitations in high-dimensional problems. The LASSO is one of the state of the art" technique for sparsity recovery, but when applied to high-dimensional problems, LASSO's performance is degraded a lot due to the presence of the measurement noise, which will result in high variance prediction and large prediction error. Ridge on the other hand is more robust to the

additive measurement noise, but also has its obvious limitation of not being able to separate true predictors from spurious predictors. As mentioned previously in many high-dimensional problems a large number of the predictors could be spurious, then in these cases ridge's disability in separating spurious and true predictors will result in poor interpretability of the model as well as poor prediction performance. The new technique that I will propose in this thesis aims to accommodate for the limitations of both of the above two methods thus resulting in more accurate and stable prediction performance in a high-dimensional linear regression problem with significant measurement noise. The idea is simple, instead of the doing a single step regression, we divide the regression procedure into two steps. In the first step we try to do a separation between the seemingly relevant predictors and those that are obviously spurious by calculating the uni-variant correlations between the predictors and the response, and discard those predictors that have very small or zero correlation with the response. After the first step we should have obtained a reduced predictor set, then we will perform a ridge regression between the reduced predictor set and the response, the result of this ridge regression will then be our desired output.

Submitted by Zhu Xiangchen

for the degree of Master of Philosophy in Information Engineering

at The Chinese University of Hong Kong in (July, 2013)

ABSTRACT

In the field of statistical learning, we usually have a lot of features to determine the behavior of some response. For example in gene testing problems we have lots of genes as features and their relations with certain disease need to be determined. Without specific knowledge available, the most simple and fundamental way to model this kind of problem would be a linear model. There are many existing method to solve linear regression, like conventional ordinary least squares, ridge regression and LASSO (least absolute shrinkage and selection operator). Let N denote the number of samples and p denote the number of predictors, in ordinary settings where we have enough samples ($N > p$), ordinary linear regression methods like ridge regression will usually give reasonable predictions for the future values of the response. In the development of modern statistical learning, it's quite often that we meet high dimensional problems ($N << p$), like documents classification problems and microarray data testing problems. In high-dimensional problems it is generally quite difficult to identify the relationship between the predictors and the response without any further assumptions. Despite the fact that there are many predictors for prediction, most of the predictors are actually spurious in a lot of real problems. A predictor being spurious means that it is not directly related to the response. For example in microarray data testing problems, millions of genes may be available for doing prediction, but only a few hundred genes are actually related to the target disease. Conven-

tional techniques in linear regression like LASSO and ridge regression both have their limitations in high-dimensional problems. The LASSO is one of the "state of the art technique for sparsity recovery, but when applied to high-dimensional problems, LASSO's performance is degraded a lot due to the presence of the measurement noise, which will result in high variance prediction and large prediction error. Ridge regression on the other hand is more robust to the additive measurement noise, but has its obvious limitation of not being able to separate true predictors from spurious predictors. As mentioned previously in many high-dimensional problems a large number of the predictors could be spurious, then in these cases ridge's disability in separating spurious and true predictors will result in poor interpretability of the model as well as poor prediction performance. The new technique that I will propose in this thesis aims to accommodate for the limitations of these two methods thus resulting in more accurate and stable prediction performance in a high-dimensional linear regression problem with significant measurement noise. The idea is simple, instead of the doing a single step regression, we divide the regression procedure into two steps. In the first step we try to identify the seemingly relevant predictors and those that are obviously spurious by calculating the uni-variant correlations between the predictors and the response. We then discard those predictors that have very small or zero correlation with the response. After the first step we should have obtained a reduced predictor set. In the second step we will perform a ridge regression between the reduced predictor set and the response, the result of this ridge regression will then be our desired output. The thesis will be organized as follows, first I will start with a literature review about the linear regression problem and introduce in details about the ridge and LASSO and explain more precisely about their

limitations in high-dimensional problems. Then I will introduce my new method called supervised ridge regression and show the reasons why it should dominate the ridge and LASSO in high-dimensional problems, and some simulation results will be demonstrated to strengthen my argument. Finally I will conclude with the possible limitations of my method and point out possible directions for further investigations.

論文摘要：高維線性回歸的監督嶺回歸

在機器學習領域，我們通常有很多的特徵變量，以確定一些回應變量的行為。例如在基因測試問題，我們有數以萬計的基因用來作為特徵變量，而它們與某些疾病的關係需要被確定。沒有提供具體的知識，最簡單和基本的方法來模擬這種問題會是一個線性的模型。有很多現成的方法來解決線性回歸問題，像傳統的普通最小二乘回歸法，嶺回歸和套索回歸。設 N 為樣本數和，p 為特徵變量數，在普通的情況下，我們通常有足夠的樣本（N> P）。在這種情況下，普通線性回歸的方法，例如嶺回歸通常會給予合理的對未來的回應變量測值的預測。隨著現代統計學的發展，我們經常會遇到高維問題（N << P），如 DNA 芯片數據的測試問題。在這些類型的高維問題中，確定特徵變量和回應變量之間的關係在沒有任何進一步的假設的情況下是相當困難的。在很多現實問題中，儘管有大量的特徵變量存在，但是完全有可能只有極少數的特徵變量和回應變量有直接關係，而大部分其他的特徵變量都是無效的。 套索和嶺回歸等傳統線性回歸在高維問題中有其局限性。套索回歸在應用於高維問題時，會因為測量噪聲的存在而表現得很糟糕，這將導致非常低的預測準確率。嶺回歸也有其明顯的局限性。它不能夠分開真正的特徵變量和無效的特徵變量。我提出的新方法的目的就是在高維線性回歸中克服以上兩種方法的局限性，從而導致更精確和穩定的預測。想法其實很簡單，與其做一個單一步驟的線性回歸，我們將回歸過程分成兩個步驟。第一步，我们丟棄那些預測有相關性很小或為零的特徵變量。第二步，我們應該得到一個消減過的特徵變量集，我們將用這個集和回應變量來進行嶺回歸從而得到我們需要的結果。

朱張向晨

信息工程課程

哲學碩士論文

香港中文大學（2013 年 7 月）

CONTENTS

1. BASICS ABOUT LINEAR REGRESSION

1.1 Introduction

A linear regression model is a model that assumes a linear relation between the input predictors $X_1, X_2, X_3, ..., X_p$ and the output response Y. Linear models have been well developed in the old ages of statistics when computers were not invented. But even today, linear regression models are still of great importance in many practical problems. Moreover, despite the existence of many fancy non-linear regression methods, linear regression models still give better results in many situations especially when we have small training set or sparse input data. In this chapter, we will start with the introduction to the basic problem formulation of linear regression, and then demonstrate how least squares can be used to solve for the regression coefficients. Then we will introduce two penalized least squares methods called ridge and LASSO in details. A complete understanding of these two methods is crucial to uncover the motivations and rationales of the new technique that will be proposed in this thesis.

1.2 Linear Regression and Least Squares

1.2.1 Standard Notations

Let y be the real valued output response that we wish to predict. And let vector $x^T = (x_1, x_2, ...x_p)$ be the input predictors. The linear model assumes that the

relationship between predictors and the responses is linear or we can use a linear relationship to do a reasonable approximation. The model has the form:

$$y = \beta_0 + \sum_{j=1}^{p} x_j \beta_j \tag{1.1}$$

Here the β_j's are the unknown coefficients, and the purpose of linear regression is to find a good technique to estimate the values of β_j's so as to predict the future values of the response y. To be more general, let's introduce the matrix notation for the model. Let N denote the number of samples we have in the training set, the responses are represented by a 1 $\times$ N vector $\mathbf{Y}^T = (y_1, y_2, ...y_N)$ and the input predictors are denoted as an N $\times$ $p+1$ matrix $\mathbf{X} = (\vec{1}, X_1, X_2, ...X_p)$, with each X_i denoting an N by 1 column vector consisting of all N samples of a single predictor. Let $\beta^T = (\beta_0, \beta_1, \beta_2, ..., \beta_p)$ be the $p+1$ unknown regression coefficients. Now we can express the linear model in (1.1) in the following matrix form.

$$\mathbf{Y} = \mathbf{X}\beta \tag{1.2}$$

Now before we move into the actual techniques to solve for this kind of regression problem, a few common assumptions and justifications of the data must be made. First of all, the solutions of β are not equi-variant under the scaling of the inputs, so we normalize the inputs using z-score standardization(which means that we will normalize the inputs to be zero mean and unit variance) to simplify the

problem. The β_0 is a constant intercept term in the model, which we want to get rid of for computational convenience. To accomplish this task, we perform the following centering procedure.

1. Calculate the mean of each predictor and response

$$\bar{x_j} = \frac{1}{N}\sum_{i=1}^{N} x_{ij},\ \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

2. Replace each x_{ij} by $x_{ij} - \bar{x_j}$, and each y_i by $y_i - \bar{y}$, at the same time we remove the first column of 1s of the input design matrix $\mathbf{X}$ so it would now be of size $N \times p$ instead of $(N+1) \times p$ and also we remove the β_0 from β so it would have p elements rather than $p+1$ elements.

After z-score standardization, the predictors now have zero mean and unit variance and the intercept term is zero. This will be held as a common assumption for the rest of the discussions if not mentioned specifically.

1.2.2 Least Squares and Its Geometric Meaning

The most popular and traditional estimation method to find the values of the regression coefficients β_js is called Least Squares. The Least Squares method aims to find β that minimizes the residual sum of squares, which is represented in the following equation:

$$\hat{\beta} = \arg\min_{\beta} \mathbf{RSS}(\beta). = \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \tag{1.3}$$

The β_0 in (1.3) is actually zero following the centering and standardization assumption we made in the previous subsection. Before solving the coefficients,

let's look at the geometric meaning of minimizing the residue sum of squares. Since $\mathbf{X}\beta = X_1\beta_1 + X_2\beta_2 + ...X_p\beta_p$, we can see that it's a linear combination of the columns of the design matrix $\mathbf{X}$. The result is a vector that lies in the column space of the matrix $\mathbf{X}$. As a result, to minimize $RSS(\beta) = ||\mathbf{Y} - \mathbf{X}\beta||^2$, we are actually seeking the vector in the column space of $\mathbf{X}$ that is closest to the observed response $\mathbf{Y}$, which should simply be the linear projection of $\mathbf{Y}$ onto the column space of $\mathbf{X}$. The resulting $\hat{\beta}$ is thus the coordinates of that projected vector $\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta}$ using the columns of $\mathbf{X}$ as basis vectors. Figure 1.1 illustrates this geometry in a two-dimensional case.

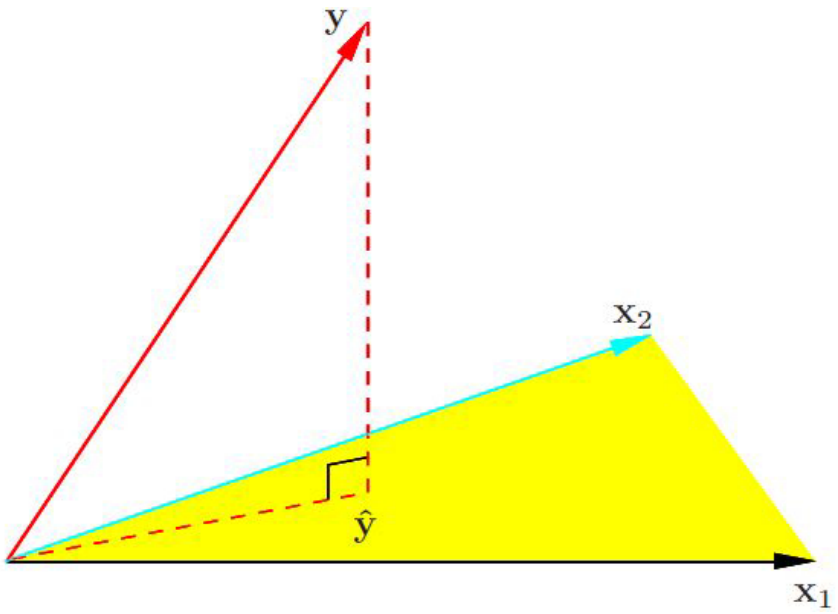

Fig. 1.1: Two dimensional geometry for least squares estimates

Now to solve for the regression coefficients $\hat{\beta}$ we write the residual sum-of-squares in its matrix form:

$$RSS(\beta) = (\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta) \tag{1.4}$$

It's clear that this is actually a quadratic function of p unkown variables. To minimize this function, we differentiate it with respect to β and get the following result

$$\frac{\partial RSS(\beta)}{\partial \beta} = -2\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\beta)$$

$$\frac{\partial^2 RSS(\beta)}{\partial \beta \partial \beta^T} = 2\mathbf{X}^T\mathbf{X}$$

(1.5)

First of all we assume that the matrix $\mathbf{X}$ is of full rank, which might not be true generally and we only assume it for this moment. Then the matrix $\mathbf{X}^T\mathbf{X}$ is positive definite, so to minimize (1.4), we only need to set the first order derivative to zero.

$$\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\beta) = 0$$

(1.6)

The solution we obtained from this equation is unique

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

(1.7)

The resulting least squares estimate for the response will be of the following form:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

(1.8)

The matrix $H = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the projection matrix that projects the original

response $\mathbf{Y}$ onto the column space of the design marix $\mathbf{X}$. It is called the hat matrix in the literature since it puts a hat onto the observed response.

As we have already mentioned that the columns of $\mathbf{X}$ may not be linearly independent in the reality, which may occur when some predictors are correlated. In this case, we cannot find a closed form unique solution like (1.7), but nevertheless the predicted response $\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta}$ is still the projection of $\mathbf{Y}$ onto the column space of $\mathbf{X}$, while the representation $\hat{\beta}$ is not unique. So far we have omitted the possible randomness observed together with the true response $\mathbf{Y}$, now let's put this randomness back into the model as an additive Gaussian measurement noise:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \tag{1.9}$$

Where ϵ is an i.i.d Gaussian noise with $E[\epsilon] = \mathbf{0}$ and $E[\epsilon\epsilon^T] = \sigma^2\mathbf{I}_N$ (the $\mathbf{I}_N$ here is a N by N identity matrix). With these settings, we can now derive some statistical properties of the ordinary least squares estimate $\hat{\beta}$. From (1.7), we can easily derive the expectation and variance-covariance matrix of $\hat{\beta}$

$$E(\hat{\beta}) = \beta \tag{1.10}$$

$$Var(\hat{\beta}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} \tag{1.11}$$

As we can see from (1.10) the ordinary least squares estimate is actually an

unbiased estimation since the expectation is the true underlying coefficients β. However, as we will illustrate in the next chapter the variance of the least squares estimates could become a serious problem when the input predictors are correlated or in other words when the matrix $\mathbf{X}^T\mathbf{X}$ is ill-conditioned. Then we will describe a penalized least squares technique called ridge regression(Hoerl, 1996) that can deal with ordinary least squares' deficiency in the correlated predictors case.

2. PENALIZED LINEAR REGRESSION

2.1 Introduction

In linear regression, it has been shown that estimates based on the ordinary least squares method have a very high probability of being undesirable, when the predictor vectors are not orthogonal (orthonormal under our general assumption that the predictors have been centered and standardized). Also note that in the case when the predictors are orthonormal, the correlation matrix $\mathbf{X}^T\mathbf{X}$ should be nearly an identity matrix and all of its eigenvalues should be around one. In the situation where certain correlations exist between the predictors, the correlation matrix $\mathbf{X}^T\mathbf{X}$ starts to deviate from the identity matrix and could be ill-conditioned and the eigenvalues are not uniformly one anymore. We will show in the following section that due to the existence of the additive measurement noise ϵ, the distance between the ordinary least squares estimate $\hat{\beta}$ and the true coefficients β can be very far if the smallest eigenvalue of the correlation matrix is close to zero.

2.2 Deficiency of the Ordinary Least Squares Estimate

Using the additive Gaussian linear model in (1.9) and let $\hat{\beta}$ denote the ordinary least squares estimate, β the true regression coefficients and σ^2 the variance of the additive measurement noise. In this section we will provide a concise argument

about why the ordinary least squares estimate would fail when the predictors are not independent. In order to demonstrate the behavior of the least squares estimate, we will analyze the quantity that represents the distance between $\hat{\beta}$ and β:

$L_1 \equiv$ Distance from $\hat{\beta}$ to β

$$L_1^2 = (\hat{\beta} - \beta)^T(\hat{\beta} - \beta) \tag{2.1}$$

Recall the mean and variance-covariance matrix of the ordinary least squares estimate that we have already derived in the end of the previous chapter:

$$E(\hat{\beta}) = \beta \tag{2.2}$$

$$Var(\hat{\beta}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} \tag{2.3}$$

From (2.1), (2.2) and (2.3) we can calculate directly the mean and variance of the squared distance measure L_1^2, the result is as follows:

$$E[L_1^2] = \sigma^2 Trace\left[(\mathbf{X}^T\mathbf{X})^{-1}\right] \tag{2.4}$$

$$Var[L_1^2] = 2\sigma^4 Trace\left[(\mathbf{X}^T\mathbf{X})^{-2}\right]$$

(2.5)

When correlations exist in the predictors, the correlation matrix $\mathbf{X}^T\mathbf{X}$ deviates from the identity matrix and becomes ill-conditioned. As a result, the sizes of the eigenvalues become non-uniform. Let us denote the eigenvalues of the correlation matrix as:

$$\lambda_{max} = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p = \lambda_{min} \geq 0$$

(2.6)

Then the expectation and variance of the squared distance between the ordinary least squares estimate and the true coefficients can be expressed explicitly in the following form.

$$E[L_1^2] = \sigma^2 \sum_{i=1}^{p} (1/\lambda_i)$$

(2.7)

$$Var[L_1^2] = 2\sigma^4 \sum_{i=1}^{p} (1/\lambda_i)^2$$

(2.8)

From (2.7) and (2.8) we can easily find the lower bounds for both the expectation and variance of the squared distance. The expectation is lower bounded by σ^2/λ_{min}, while the variance is lower bounded by $2\sigma^4/\lambda_{min}^2$. With the data collected in the design matrix $\mathbf{X}$, if one or more small eigenvalues ($\lambda_{min} \to 0$) exist for the correlation matrix $\mathbf{X}^T\mathbf{X}$, the distance between our ordinary least squares estimates and the true regression coefficients could be very far, thus resulting in poor prediction performance. Moreover if one or more λ_i becomes exactly zero then the regression result would be completely random and uncontrollable,

so without loss of generality we will assume that none of the eigenvalues of the correlation matrix will hit exactly zero. This kind of deficiency has been observed by those working on non-orthogonal data problems. In the next section we will introduce a very important regularized linear regression technique called ridge regression to deal with the limitation of the above ordinary least squares estimate.

2.3 Ridge Regression

2.3.1 Introduction to Ridge Regression

To deal with the possible limitation and general instability of the least squares estimate, A.E.Hoerl proposed a new linear regression technique called ridge regression[1]. Instead of using the possible ill-conditioned correlation matrix to calculate the estimation, Hoerl proposed to add positive values to the diagonal of the correlation matrix, the ridge estimate $\hat{\beta}^*$ is expressed in the following formula (The $\mathbf{I}$ in the formula represent the $p \times p$ identity matrix):

$$\hat{\beta}^* = [\mathbf{X}^T\mathbf{X} + k\mathbf{I}]^{-1}\mathbf{X}^T\mathbf{Y}; k \geq 0 \tag{2.9}$$

This estimate was first proposed by Hoerl in 1962 and later revised by Hoerl and Kennard in 1968 and named it "ridge regression" because the family of estimates given by $k \geq 0$ has many mathematical similarities with the portrayal of quadratic response functions, which is also proposed by Horel in 1964. The above explicit form of the ridge estimate is also the closed form solution to the

following minimization problem:

$$\hat{\beta}^* = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + k \sum_{j=1}^{p} \beta_j^2 \right\} \tag{2.10}$$

This minimization problem above can be written in its Lagarangian dual form to give us a clearer idea of what optimization problem ridge regression is trying to solve:

$$\hat{\beta}^* = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2,$$

$$subject\ to \sum_{j=1}^{p} \beta_j^2 \leq t \tag{2.11}$$

Here we can see that the ridge regression is actually a regularized version of the least squares regression, by adding a constraint to the L_2 norm of the regression coefficients. With a larger value of k (smaller value of t), a larger amount of shrinkage is applied to the coefficients and the problem of high-variance and instability caused by correlations between the predictors can be alleviated through this process. We will give a more precise argument about how ridge would solve ordinary least squares' problem by analyzing its expected prediction error in the next subsection.

2.3.2 Expected Prediction Error And Noise Variance Decomposition of Ridge Regression

First of all, we need to define some symbols and notations in order to facilitate our analysis of the expected prediction error (mean square error) of the ridge

estimate. Using the notations we have used previously, let $\hat{\beta}$ denote the least squares estimate, β denote the true regression coefficients, and let $\hat{\beta}^*$ denote the ridge estimate:

$$\hat{\beta} = [\mathbf{X}^T\mathbf{X}]^{-1}\mathbf{X}^T\mathbf{Y} \tag{2.12}$$

$$\hat{\beta}^* = [\mathbf{X}^T\mathbf{X} + k\mathbf{I}]^{-1}\mathbf{X}^T\mathbf{Y};\ k \geq 0 \tag{2.13}$$

From the above two formulas we can easily derive the relation between these two estimates

$$\hat{\beta}^* = [\mathbf{I}_p + k(\mathbf{X}^T\mathbf{X})^{-1}]^{-1}\hat{\beta} = \mathbf{Z}\hat{\beta} \tag{2.14}$$

We will use $\mathbf{Z}$ to represent the above matrix $[\mathbf{I}_p + k(\mathbf{X}^T\mathbf{X})^{-1}]^{-1}$ in our following discussion for convenience's sake (where $\mathbf{I}_p$ is the p dimensional identity matrix). The eigenvalues for this matrix is easy to calculate just by finding the solution to the characteristic equation $|\mathbf{Z} - \xi\mathbf{I}|$. Let λ_i denote the eigenvalues of the original correlation matrix $\mathbf{X}^T\mathbf{X}$, the resulting eigenvalues are as follows:

$$\xi_i(\mathbf{Z}) = \frac{\lambda_i}{\lambda_i + k} \tag{2.15}$$

Now we have obtained all the notations we need to write out the expression for

the mean square error $E[L_1^2(k)]$ (expected square distance from the ridge estimates $\hat{\beta}^*$ to the true coefficients β) by simply applying the expectation operator on $(\hat{\beta}^* - \beta)^T(\hat{\beta}^* - \beta)$. The result is as follows:

$$
\begin{aligned}
&E[L_1^2(k)] \\
&= E[(\hat{\beta}^* - \beta)^T(\hat{\beta}^* - \beta)] \\
&= E[(\hat{\beta} - \beta)^T\mathbf{Z}^T\mathbf{Z}(\hat{\beta} - \beta)] + (\mathbf{Z}\beta - \beta)^T(\mathbf{Z}\beta - \beta) \\
&= E[(\hat{\beta}^* - E(\hat{\beta}^*))^T(\hat{\beta}^* - E(\hat{\beta}^*))] + (E[\hat{\beta}^*] - \beta)^T(E[\hat{\beta}^*] - \beta)
\end{aligned}
\tag{2.16}
$$

Here we can see that the two terms in this decomposition stand for the total variance and the squared bias for $\hat{\beta}^*$, let's expand these two terms by applying (2.4) to (2.16) and see how they change with the value of the tuning parameter k. We can show that:

$$
\begin{aligned}
&R_1(k) = \sigma^2 \textstyle\sum_1^p \frac{\lambda_i}{(\lambda_i + k)^2} \\
&R_2(k) = k^2\beta^T(\mathbf{X}^T\mathbf{X} + k\mathbf{I})^{-2}\beta
\end{aligned}
\tag{2.17}
$$

Here we use the two terms $R_1(k)$ and $R_2(k)$ to represent the total variance (sum of the variances of the parameter estimates) and the squared bias (the squared distance from $\hat{\beta}^*$ to β). Referring back to the original paper by Hoerl and Kennard 1994, they gave a plot in quantitative form of the relationship between the parameter k, $R_1(k)$ and $R_2(k)$, this plot is shown as follows:

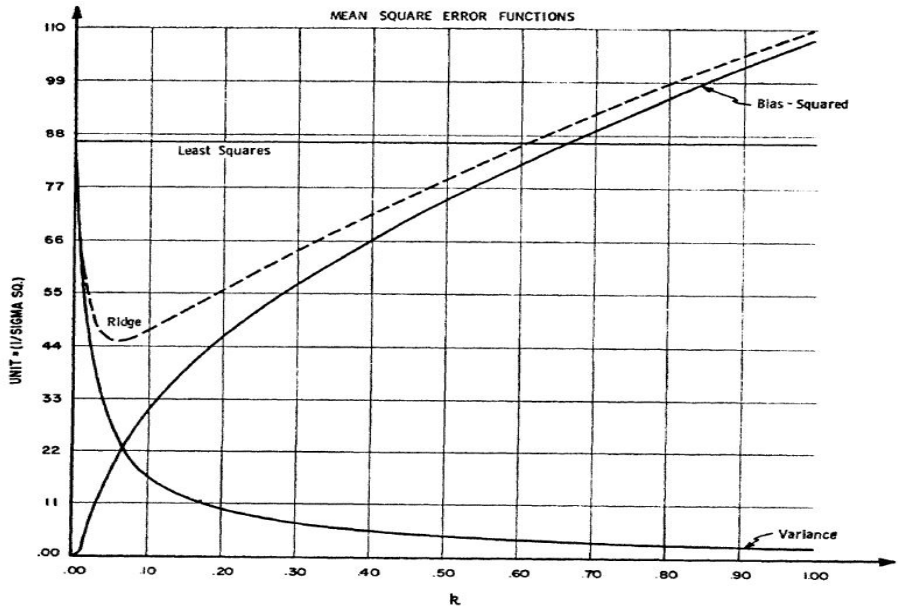

Fig. 2.1: Noise-variance decomposition for the mean square error of the ridge estimate

Here the dotted line represents the summation of $R_1(k)$ and $R_2(k)$, which is exactly the mean squared error of the ridge estimate. We can see that in the case drawn in the above plot, with a proper choice of the tuning parameter k, the ridge estimate can achieve a much smaller mean square error than the ordinary least squares estimate (the horizontal line marked as Least Squares). We will now give an explanation for why this can be achieved when the correlation matrix $\mathbf{X}^T\mathbf{X}$ is ill-conditioned. Note that we will only list and demonstrate the mathematical properties sufficient for the understanding of our following up discussions, for more precise and detailed proof, please refer back to the original paper by Hoerl and Kennard 1994 [1]. First of all, it's straight-forward that the total variance $R_1(k)$ is a monotonic decreasing function of the tuning parameter k and the squared bias $R_2(k)$ is a monotonic increasing function of the tuning parameter k. To see how their sum varies, we take the derivatives of these two functions with

respect to k and see how they change in the neighborhood of the origin.

$$\lim_{k\to 0^+}(\frac{dR_1(k)}{dk}) = -2\sigma^2\Sigma(\frac{1}{\lambda_i^2}) \tag{2.18}$$

$$\lim_{k\to 0^+}(\frac{dR_2(k)}{dk}) = 0 \tag{2.19}$$

From (2.18) we can see that if the correlation matrix $\mathbf{X}^T\mathbf{X}$ is orthonormal then the negative derivative of the total variance approaches the constant $-2p\sigma^2$ as $k \to 0^+$. However, if the matrix $\mathbf{X}^T\mathbf{X}$ becomes ill-conditioned and the smallest eigenvalue comes close to zero, the derivative of $R_1(k)$ will approach $-\infty$. As a result, the change of the total variance around zero could be very sharp. The situation is completely different for the squared bias. We can see from (2.19) that the derivative of $R_2(k)$ around 0 is always going to be 0, which means that the function is flat and changes slowly around zero. Concluding from the properties of these two derivatives, if the correlation matrix $\mathbf{X}^T\mathbf{X}$ is ill-conditioned, it's then not hard to believe that we can find a $k > 0$, resulting in a significant decrease of the total variance while at the same time sacrificing only a small amount of bias. As a result, a proper choice of k will result in an improvement in the mean square error over the ordinary least squares estimate. The original paper offered a much more detailed proof for the above argument, interested reader can refer to [1] for more information.

Summarizing from the above argument, we can see that when there exist correlations between the predictors and additive measurement noise, the ordinary

least squares estimate tends to have large variance. Ridge regression can deal with this issue caused by the correlation between the predictors by adding an L_2 constraint to the ordinary least squares problem. Moreover, through a proper choice of k, ridge regression controls the mean square error by trading a large total variance off with a small increase in squared bias. Despite the fact that the additive measurement noise ϵ introduces randomness and variability to the estimation of the regression parameters, ridge regression is robust to this noise and shrinks the total variance to achieve a small prediction error. In the next subsection we will offer another angle to look at how ridge regression shrinks and constrains different predictors.

2.3.3 Shrinkage effects on different principal components by ridge regression

In this subsection, we will offer a deeper insight into the nature of the ridge regression. To accomplish this, we first perform a singular value decomposition (SVD) on the input design matrix $\mathbf{X}$. :

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{2.20}$$

In this decomposition, $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices of size $N \times p$ and $p \times p$, with the columns of $\mathbf{V}$ spanning the row space of $\mathbf{X}$ and the columns of $\mathbf{U}$ spanning the column space of $\mathbf{X}$. Also we will define $\mathbf{u}_j$ and $\mathbf{v}_j$ to be the jth column of $\mathbf{U}$ and $\mathbf{V}$ respectively. Moreover the columns of $\mathbf{U}$ are also the eigenvectors of the correlation matrix $\mathbf{X}^T\mathbf{X}$. The middle matrix $\mathbf{D}$ is a $p \times p$ diagonal matrix, with the diagonal entries $d_1 \geq d_2 \geq ... \geq d_p \geq 0$, which are called

the singular values of the matrix $\mathbf{X}$. More precisely, these singular values are actually the square roots of the eigenvalues $\lambda_{max} = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p = \lambda_{min} \geq 0$ of the correlation matrix $\mathbf{X}^T\mathbf{X}$. If one or more values of d_js equal to zero, then the design matrix $\mathbf{X}$ is singular, in this case we can infer from (2.7) and (2.8) that the prediction error based on minimizing the residue sum of squares will be out of control. As we have already pointed out in the previous section, without loss of generality we assume that none of the singular values will hit zero. Having obtained the singular value decomposition of the design matrix, we will rewrite the ordinary least squares estimate and the ridge estimate of the response in the following forms:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\mathbf{Y} = \mathbf{U}\mathbf{U}^T\mathbf{Y} \tag{2.21}$$

$$\hat{\mathbf{Y}}^* = \mathbf{X}\hat{\beta}^* = \mathbf{X}(\mathbf{X}^T\mathbf{X}+k\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} = \mathbf{U}\mathbf{D}(\mathbf{D}^2+k\mathbf{I})^{-1}\mathbf{D}\mathbf{U}^T\mathbf{Y} = \sum_{j=1}^{p}\mathbf{u}_j\frac{d_j^2}{d_j^2+k}\mathbf{u}_j^T\mathbf{Y} \tag{2.22}$$

The $\mathbf{u}_j$s are the columns of the matrix $\mathbf{U}$, in the literature they are also known as the principal components of the column space of the design matrix $\mathbf{X}$, and they are actually an orthonormal basis for the column space of $\mathbf{X}$. Recall in 1.2.2 we have explained that, least squares is actually projecting $\mathbf{Y}$ on to the column space of $\mathbf{X}$ and the regression coefficients are just the coordinates of the projected vector using the columns of $\mathbf{X}$ as basis. Now by doing a SVD on the matrix $\mathbf{X}$ we represent the estimated response using the orthonormal basis $\mathbf{u}_j$ instead of the original basis of the columns of $\mathbf{X}$. From (2.21) $\mathbf{U}^T\mathbf{Y}$ are the coordinates of $\hat{\mathbf{Y}}$ with respect to the orthonormal basis $\mathbf{u}_j$ using ordinary least

squares. Comparing 2.20 and 2.21 we can see that ridge also compute the estimate by finding the coordinates with respect to the orthonormal basis $\mathbf{u}_j$, the essential difference here is that ridge shrinks these coordinates by the factors $d_j^2/(d_j^2 + k)$.

Now let's take a closer look at the effect of these shrinkage coefficients $d_j^2/(d_j^2 + k)$, it's easy to observe that for the principal components $\mathbf{u}_j$ associated with a smaller value of d_j a greater amount of shrinkage is then applied. To have a more precise understanding of what the sizes of these singular values d_j represent, we need to first understand the relationship between the principal components and the original basis formed by the columns of $\mathbf{X}$. From (2.21), by multiplying the orthogonal matrix $\mathbf{V}$ to the right of both sides of the equation we can then easily derive the following equation:

$$\mathbf{XV} = \mathbf{UD} \tag{2.23}$$

$$\mathbf{X}\mathbf{v}_j = \mathbf{u}_j d_j, for\ j = 1...p \tag{2.24}$$

Here $\mathbf{v}_j$ and $\mathbf{u}_j$ are the columns of $\mathbf{V}$ and $\mathbf{U}$, we can easily see from (2.24) that the new orthonormal basis vectors (principal components) are linear combinations of the column vectors of the design matrix $\mathbf{X}$. Also we can calculate from (2.21), and find the eigen-decomposition of the correlation matrix $\mathbf{X}^T\mathbf{X}$ and obtain the following results:

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^T \tag{2.25}$$

$$\mathbf{V}^T\mathbf{X}^T\mathbf{X}\mathbf{V} = \mathbf{D}^2$$

(2.26)

From (2.25) we can then derive the variance of the N samples of each principal component as follows:

$$Var(\mathbf{u}_j d_j) = Var(\mathbf{X}v_j) = \frac{d_j^2}{N} = \frac{\lambda_j}{N}$$

(2.27)

Up to this point, we can see that a small singular value d_j or a small eigenvalue λ_j of the correlation matrix $\mathbf{X}^T\mathbf{X}$ correspond to directions ($\mathbf{u}_j$) in the column space of $\mathbf{X}$ having small variance, and ridge regression shrinks the effect of these directions most. The following figure illustrates the two principal components in a two-dimensional data case.

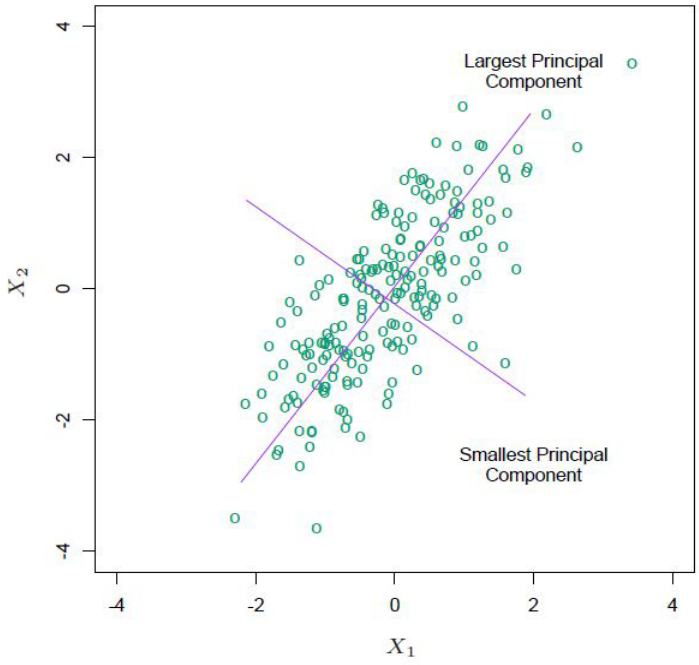

Fig. 2.2: Principal components in a two dimensional data set

Recall in the previous chapter we stated that the ordinary least squares will have large mean square error when there exists small eigenvalues λ_j for the cor-

relation matrix $\mathbf{X}^T\mathbf{X}$, and we now see that these small eigenvalues correspond to small variances for the principal components and it's now quite clear why ridge regression can alleviate the effect of these small eigenvalues, since it will apply a larger shrinkage on these small variance directions (principal components) and the impact of these directions on the regression will be shrunken a lot more as compared to those directions that establish large variances.

In conclusion, we have introduced a traditional regularized linear regression technique called the ridge regression that can compensate for the ordinary least squares' high variability and instability due to the existence of the additive measurement noise and the correlations between the input predictors. By choosing a suitable value for the tuning parameter k, ridge regression can reduce the total variance for the mean squared error substantially by incurring only a little increase in the squared bias. We also explained why ridge can accomplish this by exploring its nature of shrinkage to different principal components. In the next section we will introduce another traditional linear regression technique call the LASSO which will also be of critical importance for the understanding of our new technique.

2.4 The LASSO

2.4.1 Introduction to the LASSO

The ordinary least squares estimate has the problem of being unstable and having high variance, and we have introduced the ridge regression in the previous section to accommodate for this limitation of the ordinary least squares estimate. However we didn't mention another very crucial problem for the ordinary least squares estimate, which is its poor model interpretability. The regression coeffi-

cients for all of the predictors of the least squares estimate are non-zero. When there are a large number of predictors, the resulting linear model will be very hard to interpret. Actually, there are a number of regression techniques in the literature, known as subset selection methods specially designed to accommodate for this interpretability issue and produce estimates where the coefficients of many "unimportant" predictors will be exactly zero. For convenience's sake, we will refer to the ability of producing sparse and more interpretable models as being able to do variable selection. As the name of the subset selection methods suggests, they choose a subset of the predictors to perform the least squares regression. The criterions and methods in finding these subsets will not be talked about in details here, interested reader can refer to [2] for more information. Due to the discrete nature (predictors are either completely retained or dropped from the model) of the subset selection methods, some small changes in the data can result in very different set of predictors being selected and this means that subset selection methods can have very high variance and have poor prediction accuracy. Ridge regression has strong stability and good prediction performance. However, it cannot be used for variable selection. Due to the above reasons, researchers want to find a continuous regression technique to produce interpretable regression models while exhibiting reasonable stability. One very obvious method to find a continuous variable selection technique is to directly constrain the least squares problem with an L_0 constrain (the number of none-zero regression coefficients):

$$\hat{\beta}^{ideal} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

$$subject\ to\ ||\beta||_0 \leq t.$$

(2.28)

Even though the formulation seems to be easy, it is generally an N-P hard problem to solve. Since constraining using L_0 norm is not generally solvable, Tibshirani[3] in 1996 proposed a regression technique called the LASSO (short for least absolute shrinkage and selection operator) that uses the L_1 norm to approximate the effect of the L_0 norm. The LASSO is also known as basis pursuit in the signal-processing literature. The exact formulation of the LASSO method is as follows:

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

$$subject\ to \sum_{j=1}^{p} |\beta_j| \leq t.$$

(2.29)

By applying the Lagarangian duality on (2.29) we can rewrite it by putting the constraint into the objective function, then LASSO will also have a form similar to the ridge regression:

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \left\{ 0.5 \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + k \sum_{j=1}^{p} |\beta_j| \right\}$$

(2.30)

Recall the common assumption that the data are centered and standardized, so the β_0 in (2.29) and (2.30) are zero and can be deleted. This optimization problem can be converted into a standard quadratic programming problem for which there exist efficient algorithms to solve for the values of the regression coefficients

for all values of $k > 0$. These algorithms are called "path-algorithms" for LASSO. Famous and efficient path algorithms include the "least angle regression (LAR)" and the "coordinate descent" from [4] and [5]. Later we will introduce the coordinate descent algorithm in detail to offer a clearer idea of how the solution to a LASSO problem might be computed. But before doing that, we will first give an intuitive explanation offered [3] why LASSO can perform variable selection.

2.4.2 The Variable Selection Ability and Geometry of LASSO

Why the L_1 penalty of LASSO can set some of the regression coefficients to be exactly zero ? According to Tibshirani[3], if the design matrix $\mathbf{X}$ is orthonormal, then the solution to (2.30) can be expressed explicitly in terms of soft-thresholding the ordinary least squares estimates by k. Let $\hat{\beta}^o$ denote the ordinary least squares estimate. The LASSO solution to (2.30) is given by:

$$\hat{\beta}_j^{lasso} = sign(\hat{\beta}_j^o)(|\hat{\beta}_j^o| - k)^+ \tag{2.31}$$

We can see from (2.31) that the LASSO solution for an orthonormal design matrix case is just a soft thresholding of the ordinary least squares estimate by an amount k. The operation of soft thresholding is quite straight forward, if the absolute value of the coefficient exceeds k we will shrink its absolute value by k, otherwise the coefficient will be set to zero. The following figure illustrates the relations between the least squares coefficients and the LASSO coefficients, in the orthonormal design case. The dashed line stands for the original values (the coefficient values from ordinary least squares estimate) of the coefficient and the

k is set to be two.

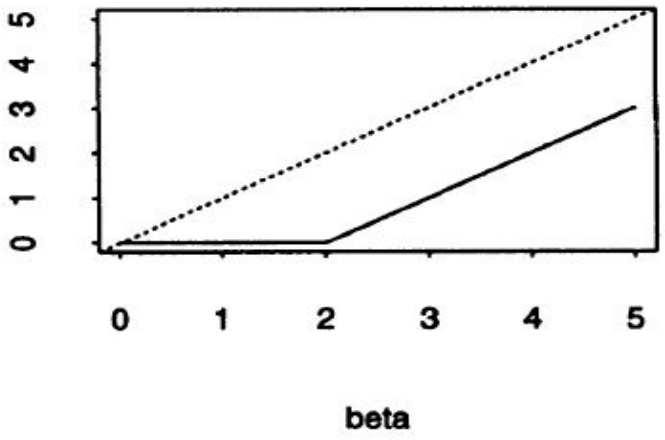

Fig. 2.3: Soft-thresholding effect for the LASSO estimate in the orthonormal design case

We can see an obvious difference between the shrinkage effect for the LASSO and Ridge. It is easy to calculate that in an orthonormal design case, the ridge estimate (solution to 2.10) only divides the ordinary least squares estimate by a constant value, $\hat{\beta}_j^* = \hat{\beta}_j^o/(1+k)$, thus none of the regression coefficients will be set to zero. On the other hand the coefficients for some of the LASSO estimate will be zero when the absolute value of the original least squares estimate is smaller than the threshold k.

The situation becomes somehow complicated when the design matrix $\mathbf{X}$ is not orthonormal since we don't have a closed form solution for (2.30). We here show a geometric explanation by Tibshirani in a two dimensional case and compare it with the ridge estimate in the same geometric plot. In the two dimensional case the unknown coefficients will contain only two values $\beta = (\beta_1, \beta_2)$. To pinpoint the solutions to (2.30) and (2.10) in a two-dimensional graph, we first draw the equi-contours of the residual sum of squares centered at the ordinary least squares

estimate:

$$(\beta - \hat{\beta}^o)^T \mathbf{X}^T \mathbf{X}(\beta - \hat{\beta}^o) = \gamma^2, \gamma \in R \tag{2.32}$$

In the two-dimensional case, the contours will be of elliptical shape. The next thing to draw on the 2-D plot is the constrained region induced by the LASSO and ridge constrains from (2.29) and (2.11). The resulting constrained region for LASSO is a diamond ($|\beta_1| + |\beta_2| \leq t$), while constrained region for ridge is a disk ($\beta_1^2 + \beta_2^2 \leq t^2$). Then the solutions to (2.30) and (2.10) will be the first intersection of the contours with the constrained regions. The following figure shows the geometry described above:

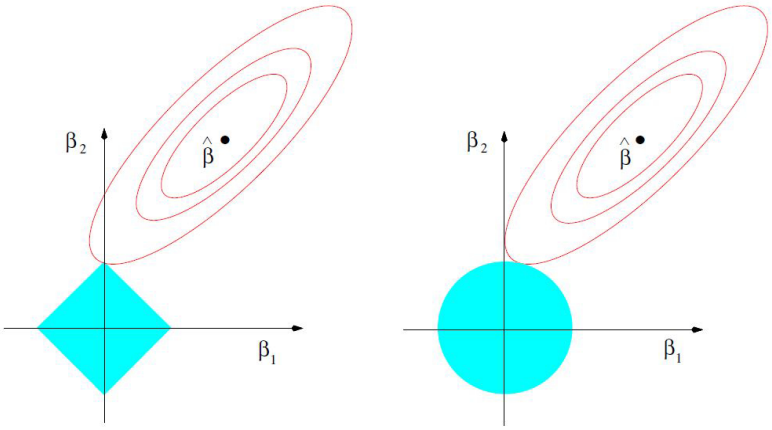

Fig. 2.4: LASSO(left) and ridge (right) estimates in a two-dimensional regression.

In the 2-D LASSO situation, the constrained region is a diamond which has corners, so it is all the more likely for the solution to occur at a corner and then one parameter β_j will be equal to zero. But for ridge regression, as we can see that the constrained region is a disk which doesn't have any corners, so it is not likely that the intersection between a disk and an elliptical curve will have any axis being zero. When $p > 2$, the diamond will become a rhomboid, which will have

many hyperplanes and there are many more opportunities for the intersections to contain zero coordinates. The above is only an intuitive illustration that why LASSO may be able to perform variable selection while ridge regression can't. But we didn't show the precise conditions for which LASSO might give out sparse solutions. For more detailed proofs, interested reader can refer back to original paper [4] of Tibshirani.

2.4.3 Coordinate Descent Algorithm to solve for the LASSO

In this subsection we will introduce a very recently developed path algorithm called "coordinate descent" by Hastie and Tibshirani to solve the LASSO problem. Apart from its high efficiency in computing the result of LASSO (much faster than the least angle regression algorithm), this algorithm will also provide us some more insights into the characteristics of the LASSO estimate.

The general idea of coordinate descent is very simple, for each value of the tuning parameter k, instead of solving (2.30) for all β_js at the same time, the coordinate descent algorithm tries to optimize each β_j one by one while holding all the other $p-1$ regression coefficients fixed. This operation is cycled until the values of all β_j's converge, namely until the values of β_j's stay unchanged for two consecutive iterations. The beauty of coordinate descent is to break the optimization problem (2.30) into sequences of p uni-variant regression problems which are very simple to solve individually. We now give a concise mathematical description of coordinate descent to solve (2.30) with a given value of k:

Cycle the following steps over $j = 1, 2, ...p$ till convergence with the initial values of all β_j's equal to zero:

1. Compute the partial residue. $r_{ij} = y_i - \sum_{k \neq j} X_{ik}\beta_k$, denote the vector

$(r_{1j}, r_{2j}, ..., r_{Nj})^t$ as R_j.

2. Compute the simple uni-variant regression coefficient of these residuals on the jth predictor.

$$\beta_j^u = \frac{1}{N} \times \frac{< X_j, R_j >}{< X_j, X_j >} = \frac{1}{N} \sum_{i=1}^{N} x_{ij} r_{ij} \tag{2.33}$$

3. Update β_j by soft-thresholding β_j^*, this is the step where the penalty is taken into account. (Subtract its absolute value by λ, if it hits 0 the value will be changed to 0, otherwise decrease the absolute value by λ).

$$\beta_j \longleftarrow S(\beta_j^u, k) = sign(\beta_j^u)(|\beta_j^u| - k)_+ \tag{2.34}$$

We can repeat the above iteration for all values of k, which will then result in the entire LASSO path. Here we offer an intuitive explanation for why the coordinate descent is a very fast algorithm. If the true underlying model is the kind of sparse model that we desired for LASSO, during the soft-thresholding step many β_js as long as they hit zero will stay at zero for the rest of the iterations so that we can immediately skip to the next round. The asymptotic analysis of the convergence and computational complexity of this algorithm will not be demonstrated here, interested readers can refer back to the original paper [5], where greater details are presented.

This coordinate descent algorithm also offers us some additional insights into the behavior of the LASSO (2.30) when the dimension p becomes large. Recall in

the beginning of the thesis we have mentioned that the true linear model of concern will be of the form (1.9) which contains the additive Gaussian noise ϵ. Then for each iteration where we compute the partial residues and the corresponding regression coefficient, the effect of this additive noise is counted in. When the variance of this additive measurement noise is significant and the dimension p is large, the effect of this noise is accumulated and magnified by the many rounds of iterations. As a result the LASSO estimate could establish high variance and instability together with a poor prediction performance in this situation. Apart from the above problem of LASSO, for the usual cases when $N > p$ if there are high correlations between the predictors, it has been empirically observed (Tibshirani 1996) that the prediction performance of the LASSO is dominated by ridge regression, which means that LASSO also can't deal with the correlated predictors very well. In the next chapter we are going to offer a more detailed description to the high-dimensional linear regression and demonstrate more precisely how traditional regression techniques would behave under the high dimensional setting, which will then offer us the motivation for our new technique.

3. LINEAR REGRESSION IN HIGH-DIMENSIONAL PROBLEMS

3.1 Introduction

With the vast development pace of modern science and technologies, it's quite often that we meet problems with extreme large feature space (p very large), which may be in the order of tens of thousands while the available sample size may be only a few hundred. Survival analysis is one of the typical example for a high-dimensional problem. In this problem the set of predictors $X_1, X_2, ...X_p$ represent the gene expression measurements collected from DNA microarrays and the output response Y is a quantitative variable related to the survival time (or just the survival time itself) of the patients. The task is then to find the relationship between the gene-expressions and the survival time in order for the doctors to diagnose more accurately. In this kind of survival analysis problem, the typical magnitude of the number of gene expressions would be a few thousands while the number of patients investigated would be at most a hundred or two. The survival analysis is only one of the many important applications for high-dimensional linear regression, so finding a good high-dimensional linear regression technique has become a very important and hot research topic in the statistics society. In this chapter we will give the common assumptions, notations and set up the model for the typical type of high-dimensional linear regression problems

of our concern and will then discuss limitations of the usual regression techniques (ridge and LASSO) when applied to problems in a high-dimensional situation.

3.2 Spurious Predictors and Model Notations for High-dimensional Linear Regression

In the survival analysis problem mentioned above, though we have a huge pool of genes available as predictors, it is quite possible that only a very small number of the genes may truly have effects on the occurrence of that specific disease. In this thesis, we will call the predictors that don't have any direct connection with the response the "spurious predictors". In many problems (for example in survival analysis), the majority of the predictors are actually spurious, so apart from the poor prediction performance when the spurious predictors are counted in, the interpretability of the resulting model will also be unacceptable. Due to the above reason, being able to filter out those spurious predictors would be a critical step towards a good regression technique in high-dimensional linear regression problems.

Just as what we have done before, we will give the model formulation, common assumptions and notations for the type of high-dimensional linear regression problems of our concern. And these assumptions and notations will be used for the rest of the discussions in this thesis. Let $\mathbf{X} = (X_1, X_2, ..., X_p)$ denote the $N \times p$ design matrix and $\mathbf{Y}$ denote the $N \times 1$ observed response, moreover the predictor data is assumed to come from a multivariate Gaussian distribution with a certain covariance structure. The difference here between the high-dimensional problems and the usual type of problems lies in the relationship between p and N. For usual low dimensional problems we have $p < N$, while in high dimensional

problems we have $p >> N$. Let m denote the number of true predictors in the model which means that in the true underlying model the regression coefficients of all the other $p - m$ predictors should be exactly zero. We also assume the following relationship between N, p and m:

$$m < N << p \tag{3.1}$$

Let ϵ denote the additive measurement noise and we further assume that this noise follows a Gaussian distribution that is independent of all the other random quantities in the model with mean zero and variance σ^2. As a result the high-dimensional linear model of our concern will be of the same form as (1.9), and all the data is also assumed to be centered and standardized.

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon, \quad \textit{where } \beta \textit{ is a } p \times 1 \textit{ vector} \tag{3.2}$$

Moreover we also assume that all the spurious predictors are statistically independent from (ie, if we regard each predictor as a random variable they should be not correlated) the true predictors (we will call this as the ideal-separation assumption). Under this assumption, a true predictor will have a more substantial marginal correlation with the response than a spurious predictor. Though the above assumption may not be always true in practice, it will often be a very good working hypothesis in many practical problems (We can also see from (3.2) that the marginal correlation between a spurious predictor and the response should be

approaching zero). These general notations and assumptions will be used latter on for the discussion of our new technique. Before that let's first have a brief look at how the traditional linear regression techniques would behave under the high-dimensional situation.

3.3 Ridge and LASSO in High-dimensional Linear Regression

First of all, let's see what the performance of ridge regression will be when it is applied to a high-dimensional problem. As we have stated in the above section that when $p >> N$ many of the predictors in the design matrix will be spurious, an obvious disadvantage of ridge regression is that the interpretability of the resulting model will be very poor. For example in the microarray data problem described above, if we uses ridge regression, then we are trying to interpret the behavior of a certain disease by analyzing thousands of genes, which is not so acceptable and meaningful. In addition to this, the poor interpretability of ridge regression is not the only problem that occurs when the dimension grows high. Recall that ridge regression can deal with the correlation between the predictors very well because it shrinks different principal components based on the variance of each principal component, the smaller the variation leads to the larger the shrinkage. In this way the ridge estimate shrinks the coefficients of the co-expressing predictors towards similar values and the high-variance correlation structures (which are the high-variance principal components) between the predictors are retained because a smaller shrinkage is applied. Coming back to the high-dimensional problem where $p >> N$. Despite the fact that the spurious predictors are actually not related to the response, there might exist some serious correlations between the spurious predictors themselves which as a result can represent some large variance principal

components. Let's illustrate this problem of ridge regression in a high-dimensional problem more clearly through a simple example from [6]. Figure 3.1 shows some simple fictitious microarray data for our illustration purpose only. This figure is a heat map for the observed gene expressions with each gene representing a row and the data from one patient on one microarray representing a column (which means columns are samples and rows are genes). The gene expression (or the value of the data) is represented by the color from blue (low) to yellow (high). Moreover, there are totally 20 patients divided into four equally sized blocks. The second and fourth block of patients are sick with a certain disease while the others are healthy. The first two blocks of patients have blood type A while the others have blood type B. For the genes marked as A the second half set of 10 patients have a significantly higher expression than the first set of ten patients, resulting in the largest variation. For the genes marked as B, the second and fourth part of patients have higher gene expressions than the first and third part of the patients (this variation is not as large as the genes in A, cause the color difference is not as large as in A). Other non-marked genes don't show any obvious variations. In the bottom of the figure we showed the first two principal components of the expressing data, with u_1 consisting of the genes in A and u_2 consisting of the genes in B. When ridge regression is applied to this problem, the principal component u_2 will be shrunken more as compared to the principal component u_1 because u_1 has a larger variance than u_2. In this situation, we could have poor prediction performance if the genes in A are actually spurious, which is possible when the variation in the genes from A represent some biological process (blood type in this case) that is not related to the disease of our concern (ie the genes in A are not actually correlated with our response $\mathbf{Y}$).

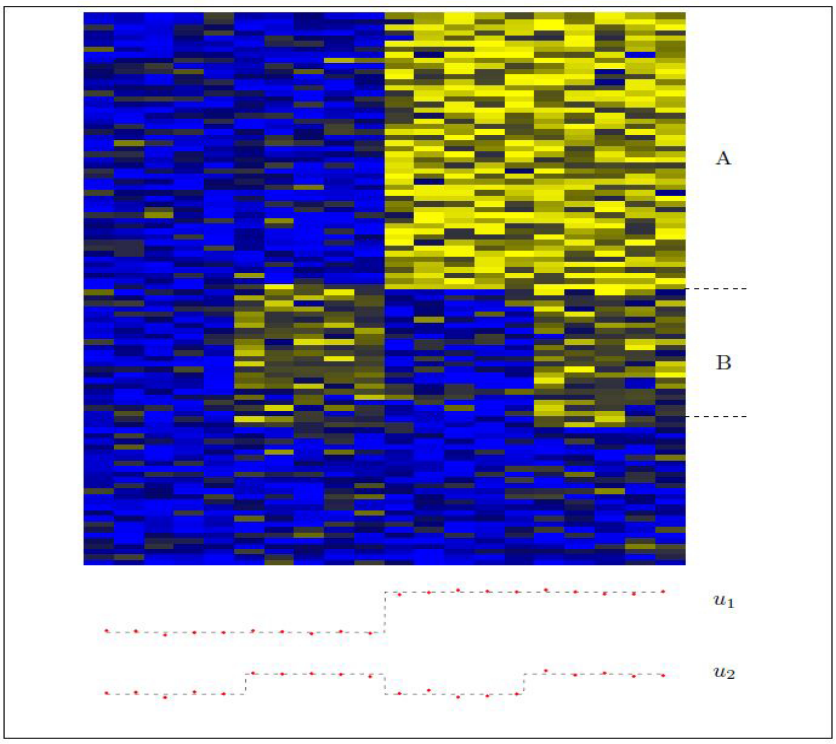

Fig. 3.1: The fictitious data for microarray data

In a word, in a high-dimensional linear regression problem, ridge regression can't produce an interpretable model since the regression coefficients for all the predictors will be non-zero. Moreover, when some spurious predictors are correlated and exhibit high variance, the prediction performance (the smaller the expected prediction error, the better the prediction performance) of ridge regression could also be poor. This concludes the two major limitations of ridge regression in a high-dimensional linear regression problem.

As for the LASSO, we have already offered some insights into the problems of LASSO when $p >> N$. We have explained that the prediction error increases rapidly when the variance of the additive measurement noise starts to increase. Apart from the high-variance and instability of LASSO in high-dimensional problems, we want to know if LASSO can produce a true or consistent sparsity pattern. To clarify the problem and make things precise, we offer a brief mathematical def-

inition for variable selection consistency. Let A (a subset of $\{1, 2, ..., p\}$) be the index set of true predictors whose regression coefficients are not zero. A variable selection technique in linear regression is said to be consistent if

$$P(\left\{ j : \hat{\beta}_j \neq 0 \right\} = A) \longrightarrow 1 \; as \;\; N \longrightarrow \infty \tag{3.3}$$

In our high-dimensional linear regression problems where $p >> N$, p is allowed to grow with N in the above definition of variable selection consistency. Due to the importance of LASSO in high-dimensional linear regression problems, many researches have studied its asymptotic behavior and consistency. In [7], a sharp threshold for LASSO was proposed to indicate the condition that LASSO has to satisfy in order to be consistent. Let m denote the true number of predictors that are related to the response, σ^2 denote the noise variance, N,p,k still follow the same definition as before. Then the LASSO will perform a consistent variable selection of the coefficients only if the following relationship is satisfied,

$$\frac{N}{2m \log p - m} > 1 + \frac{\sigma^2}{k^2 m} \tag{3.4}$$

which means that only when the above inequality holds will LASSO be able to recover the true sparsity pattern with a probability converging to one (when N and p $\rightarrow \infty$). The development of this inequality is extremely complicated. Since it's used only for illustration here, it will not be explored in details. Interested readers can refer to [7] for details. It can be seen that this condition

(3.4) can be easily violated when p and σ^2 are large, in which case LASSO can select spurious predictors and discard relevant predictors resulting in a model with a wrong set of predictors. Apart from this, it has also been mentioned in the original paper by Tibshirani[3] that LASSO also can't deal with correlated groups of predictors very well. LASSO will tend to select only one predictor from a correlated group and not care about which one is selected. This means that the correlated structures and relations between these predictors are completely ignored by LASSO. In conclusion, in a high-dimensional linear regression problem with a significant measurement noise, the prediction result of the LASSO is unstable and the prediction error could be very large. In addition, variable selection has a large probability of being inconsistent and one predictor out of a group of correlated predictors could be selected.

Due to these limitations of LASSO and ridge regression, we are motivated to seek a linear regression technique that can offer us interpretable linear models in high-dimensional problems, while being more robust to the measurement noise.

4. THE SUPERVISED RIDGE REGRESSION

4.1 Introduction

In this chapter we will propose a regression technique called "supervised ridge regression" (SRR) for high dimensional problems. The purpose of introducing SRR in high-dimensional problems is to accommodate for the limitations of both LASSO and ridge regression while preserving both of their advantages. The idea of SRR is actually quite straight forward. Instead of doing regression directly on the huge high-dimensional design matrix $\mathbf{X}$, we first perform a cleaning operation on the predictors to filter out those predictors whose estimated uni-variant correlation with the observed response $\mathbf{Y}$ is small. We then perform ridge regression on the resulting reduced data matrix. Using the example from figure (3.1), the genes marked as A would have been filtered out during the cleaning step, so they now would not be counted in the regression model anymore.

4.2 Definition of Supervised Ridge Regression

We will directly use the notations and assumptions from (3.2) with $\mathbf{X}$ and $\mathbf{Y}$ representing the $N \times p$ observed design matrix and the $N \times 1$ observed response and ϵ representing the additive Gaussian measurement noise with mean zero and variance σ^2. Also all the data used are assumed to have been centered and

standardized. First of all we will give a brief description for the proposed SRR:

Supervised Ridge Regression

1. Compute the uni-variant (marginal) regression coefficients for each predictor X_j with respect to the observed response $\mathbf{Y}$.

2. Compose a reduced data matrix made of by only those predictors whose uni-variant regression coefficient computed in step 1 exceeds a threshold θ. The value of this θ can be estimated by cross-validation (this will be explained in details later).

3. Perform ridge regression on the reduced data matrix to get the estimate for the regression parameters that can be used to make the prediction.

4. Set the coefficients for all the predictors that are not used in step 3 to be zero.

From the above description, we can see that the idea of SRR is actually quite simple to understand. We call our method "supervised" ridge regression because before we actually fit the data we offered some additional information to the ridge estimator about what are probably the right set of predictors to operate on. Due to this reason we call the steps of getting the reduced data matrix the "supervision steps", which as a result produces the name of "supervised ridge regression" for our technique. We now give a more detailed description of this method. For step 1, we denote the uni-variant regression coefficient of predictor X_j with the response $\mathbf{Y}$ as s_j, and the value of s_j can be calculated using the following equation.

$$s_j = \frac{X_j^T \mathbf{Y}}{||X_j||}, \ where \ ||X_j|| = \sqrt{X_j^T X_j}$$

(4.1)

This formula is valid because we are considering a Gaussian linear model, for some more generalized model, we may use the partial log-likelihood relating the data for X_j and $\mathbf{Y}$ to compute this uni-variant regression coefficient. Nevertheless, using the s_j in (4.1) already suffices for the problem of our concern (which is described clearly in 3.2).

We use the following procedure to estimate the value of θ using cross-validation.

1. Using all of the N data samples at hand and then compute the uni-variant regression coefficients $|s_j|$s in (4.1). We then find the mean and standard deviation of $|s_j|$s (here we take the absolute values to accommodate for the possible situations where there are clusters of variables that influence the function in opposite ways), for which we will use $\tilde{\theta}$ and $\tilde{\sigma}$ to represent respectively.

2. For k-fold cross-validation, we will divide the N samples into k equally sized subsets. Each time when we try to compute the cross-validation error for a specific value of θ, we will use $k-1$ subsets as training samples and the one set left as the validation set out of which we can compute the prediction error from the trained model. We will do this k times by leaving out each one of the k subsets and the cross-validation error will be the average value of the k prediction errors.

3. Compute the cross-validation error for values of θ within the range of $\tilde{\theta}-2\tilde{\sigma}$ to $\tilde{\theta}+2\tilde{\sigma}$, we don't have to compute every value of θ within this range, we can manually adjust the step size. It is already accurate enough if we use small step sizes like $0.1\tilde{\sigma}$. The desired value of θ is the θ that gives us the smallest k-fold cross-validation error.

Let A_θ be the collection of indices such that $|s_j| > \theta$. One thing to notice

here is that, it should turn out that we have the same index set A_θ for a range of values of θ, then what is the exact value of θ within that optimal range would not really matter. Later in the simulation section we will demonstrate in details about how to find the reduced data matrix and we will plot the cross-validation error against the number of predictors we choose for A_θ.

We can now continue with the ridge regression step. Let us use $\mathbf{X}_\theta$ to denote the matrix consisting of the columns of $\mathbf{X}$ corresponding to the index set A_θ. Then we will compute the regression coefficients for the predictors in the set A_θ by ridge regression.

$$\hat{\beta}^* = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \sum_{j \in A_\theta} x_{ij}\beta_j)^2 + k \sum_{j \in A_\theta} \beta_j^2 \right\} \tag{4.2}$$

Then the final result from SRR for the complete set of regression coefficients is as follows:

$$\hat{\beta}_j^{SRR} = \hat{\beta}_j^*, if\ j \in A_\theta \quad \hat{\beta}_j^{SRR} = 0, if\ j \notin A_\theta \tag{4.3}$$

After we have obtained the estimated coefficients $\hat{\beta}^{SRR}$, for a given input test data x^* the predicted response is then $\hat{y}^* = (x^*)^T \hat{\beta}^{SRR}$. Up to this point we have clearly defined the supervised ridge regression and how to do the regression and prediction using SRR.

4.3 An Underlying Latent Model

In this section we will describe a latent variable model that relates the predictors and the responses through a couple of latent quantities to support our supervised ridge regression method. For our response $\mathbf{Y}$ we assume that it is related to a set of latent variables $\mathbf{U}_1, \mathbf{U}_2, ...\mathbf{U}_M$ through the following linear model:

$$\mathbf{Y} = \gamma_0 + \sum_{m=1}^{M} \gamma_m \mathbf{U}_m + \epsilon \tag{4.4}$$

Also, for the predictors we assume that spurious predictors are not related to the latent quantities $\mathbf{U}_m$ while the true predictors are related. Let Q be the index set (which is a subset of $\{1, 2, 3....p\}$) for the true predictors. Then for $j \in Q$ the relationship between the latent quantities and the predictors $\mathbf{X}_j$ is as follows:

$$\mathbf{X}_j = \alpha_{0j} + \sum_{m=1}^{M} \alpha_{1jm} \mathbf{U}_m + \eta_j, j \in Q \tag{4.5}$$

For those X_j where $j \notin Q$, they are independent of all the latent quantities $\mathbf{U}_m$. Bringing this latent variable model into the microarray data testing problem we talked about in the previous chapter, we can actually think of the latent quantities $\mathbf{U}_m$ as some specific biological processes or data representation of some cell types that affect the occurrence of the cancer of interest. The subset Q will represent the set of genes that are related to the forming of these cell types. ϵ and η_js are additive Gaussian noise that are independent of all the other random quantities. Moreover we also assume that $\mathbf{U}_m$s are orthogonal (linearly indepen-

dent) to each other and $var(\mathbf{U}_1) > var(\mathbf{U}_2) > var(\mathbf{U}_3) > .... > var(\mathbf{U}_m)$. The variance here is the N sample scalar variance of each latent vector just like in (2.26) These are reasonable assumptions for practical problems, but they need not be always true. After we have introduced this model, we can now regard the supervised ridge regression as a method for fitting this model.

1. The first supervision step in SRR gives an estimation of the set of true predictors by $\hat{Q} = A_\theta$

2. With the above estimated $\hat{Q}$, we get the reduced data matrix $\mathbf{X}_\theta$. Then we can perform a SVD on this reduced data matrix and get its principal components $\mathbf{U}_{\theta,j}$, which are then used to estimate the latent quantities $\mathbf{U}_m$.

3. Finally, the ridge regression that fits (4.2) gives an estimate for (4.4).

The first step above is very natural, since from (4.4) and (4.5) we can see that for each true predictor $\mathbf{X}_j \in Q$, its correlation with $\mathbf{Y}$ should be larger than those spurious predictors not in the set Q. Step 2 is also natural since using principal components to approximate latent quantities is the most adequate approach when no additional information is available. For the last step, in order to deal with the possible correlation between the true predictors we use ridge regression (in page 19 and 20 we have explained that ridge fits the model using principal components of the design matrix as basis vectors) to fit the coefficients. As a result we can see that the proposed SRR is a regression technique that can fit the above latent model. Though this model is somehow artificial, it still offers some useful insights into the internal tasks that the supervised ridge regression accomplishes while dealing high-dimensional linear regression problems.

4.4 Ridge LASSO and Supervised Ridge Regression

In this section, we will compare the SRR with our two traditional linear regression techniques ridge regression and LASSO, which will then reveal the value for our new SRR in high-dimensional linear regression problems.

4.4.1 LASSO vs SRR

First of all let's recall what we have mentioned in 3.3 and 2.4.3 about LASSO's limitations when applied to a high-dimensional linear regression problem. The main limitations of LASSO can be summarized as the two points:

1. The LASSO estimate is very sensitive to the variance of the additive Gaussian noise. When the noise variance increases the LASSO estimate becomes more unstable with very random variable selection behavior. As a result, the prediction performance degrades rapidly.

2. LASSO can't deal with correlated group of predictors and it only tends to select one predictor out of a group of correlated predictors, while the better way is to select the whole group.

Let's look at the first problem of LASSO and how the supervised ridge regression can avoid it. In 2.4.3, we have provided an explanation for why LASSO is so sensitive to noise. Using the coordinate descent algorithm, in each iteration we are actually performing uni-variant regression for a specific predictor with the current residue. As such, the additive Gaussian noise attached to the residue is considered in the calculation for every single iteration. Due to this reason the effect of the noise is accumulated and magnified, leading to the instability and variability of the LASSO estimate. However in our method of SRR, we only have one iteration of uni-variant screening (the supervision step), where we calculate

the uni-variant regression coefficient between the response $\mathbf{Y}$ and each of the predictors $\mathbf{X}_j$, and the effect of the noise is only counted in this step and not iteratively calculated as in LASSO. As a result our SRR is not sensitive to the effect of the additive noise and the prediction error only increases slowly when the noise increases, which is much better than the rapid increase of prediction error in LASSO. We can actually think of the first supervision step of the SRR to be very similar to the first iteration of coordinate descent for LASSO. So the first step of supervision actually adopts partially the principle of LASSO to screen out those predictors that are obviously spurious. For the second weakness of selecting only one out of a group of correlated predictors, SRR uses the ridge regression to fit the reduced data matrix and the ridge regression will shrink groups of correlated predictors together. The SRR will only drop the predictors that are spurious in the first step of supervision and will tend to select and shrink correlated true predictors together. This means that the SRR will not drop any one of a group of correlated predictor as long as this group of predictors are true. Of course if a group of correlated predictors are spurious the SRR would have dropped the whole group during the supervision step. From the above discussion, we can see that the SRR is much more robust to the additive Gaussian noise than LASSO and can deal with the groups of correlated true predictors very well. The SRR retains the most important advantage of LASSO is preserved, which is the ability to produce a sparse and interpretable linear model.

4.4.2 Ridge regression vs SRR

Now let's compare the ridge regression with the supervised ridge regression. It seems that the ridge regression should be quite close to SRR. However, it is the

supervision step that makes SRR different and standing out in a high-dimensional linear regression problem. Let's first recall the two limitations of ridge regression in high-dimensional problems described in 3.3:

1. Ridge regression will select every single predictor into the model no-matter how large the dimension p is, so in a high-dimensional problem, the interpretability of a ridge estimate is unacceptably low.

2. When there are groups of correlated spurious predictors that have a large variance as one of the principal components for the design matrix $\mathbf{X}$, the ridge estimator will tend to apply only a very small amount of shrinkage to these predictors (In the microarray example, the set of genes marked as A would be such predictors as described). As a result these set of spurious predictors will have significant effect on the ridge estimate, which will then result a poor prediction performance.

For the first interpretability problem of ridge regression, the supervision step of SRR solves it automatically by only choosing those predictors whose uni-variant regression coefficient with the response exceeds θ. As a result most of the spurious predictors would have been removed from the model during this supervision step and we can arrive at a reasonably interpretable model. For the second problem of ridge, in which there exist groups of correlated spurious predictors we will use the microarray data testing problem shown in figure 3.1 as an example. For the original ridge regression, we have already mentioned that it will treat the set of genes in A as very important predictors since they represent the first principal component and the least amount of shrinkage will be applied to them, while more shrinkage would be applied to the genes in B as they only represent the second principal component. Unfortunately the genes in A are actually not

related to our desired target disease at all, so the prediction performance of ridge regression would be quite poor. In high-dimensional problems there could exist many groups of correlated spurious predictors like A, which would then become a serious problem for ridge regression. In SRR, the supervision step will help us get rid of the genes in A before we pass the regression problem to ridge regression. As a result the genes in B would form the first principal component for the reduced data matrix and receive the most attention by ridge regression. In conclusion, the supervision step in SRR help us clean those obviously spurious predictors before the data is fitted by ridge regression, and the limitation caused by correlated groups of spurious predictors are then naturally alleviated.

Up to this point, we can see that the SRR does preserve both of the advantages of LASSO and ridge by producing a sparse linear model that can deal with the correlations between the predictors very well. The SRR is much more robust to the additive Gaussian noise than LASSO and also unlike ridge SRR would not get affected by groups of correlated spurious predictors. In the next section we will demonstrate and test the function of SRR through some simulation results on a high-dimensional Gaussian linear model, and the result will be compared with both ridge and LASSO to testify the arguments provided above.

5. TESTING AND SIMULATION

In this chapter, we will do experiments on several models in order to test the performance of the proposed supervised ridge regression compared with the original ridge regression and LASSO. We will compare the performance of the three methods based on two critical aspects: 1. the expected prediction error 2. the number of predictors they selected into the model. We will start with a typical high-dimensional Gaussian linear model to illustrate the general behavior followed by a series of modified experiments to test the behavior of SRR under different situations.

5.1 A Simulation Example

In this first section we will give a simple illustration for the actual effect of SRR when applied to a typical high-dimensional Gaussian linear model. The result will be compared with both ridge and LASSO running on the same model, so that the arguments provided in the previous section could then be tested. First of all, let's introduce the exact high dimensional Gaussian linear model where we generate our data from. The mathematical formulation is actually the same as what we have already defined in 3.2:

$$y = x_1\beta_1 + x_2\beta_2 + ... + x_{800}\beta_{800} + \epsilon \tag{5.1}$$

So here the dimension we use is $p = 800$ and $x_1, x_2, ...x_{800}$ come from a multivariate Gaussian distribution (each individual predictor has mean 0 and variance 1) with pairwise correlation specified as follows:

x_1 to x_5 pairwise correlation 0.8, x_{20} to x_{30} pairwise correlation 0.8, x_{45} to x_{50} pairwise correlation 0.9. x_{51} to x_{100} pairwise correlation 0.9. All the other predictors are pairwise independent.

Also in our data model, most of the predictors are spurious, we have $\beta_{51} = \beta_{52}, ..., = \beta_{800} = 0$, and the first 50 predictors are true while their corresponding regression coefficients β_1 to β_{50} are fixed constants with values between 0 and 1 (In this way the correlation structure mentioned above guarantees that we have both correlated groups of spurious and true predictors). The additive measurement noise ϵ also comes from a uni-variant Gaussian distribution with mean 0 and variance σ^2 (This is the parameter that we control and the expected prediction error will be plotted against this variance of the noise). To satisfy the needs for a high-dimensional linear regression problem we will take 300 samples (which is much smaller than 800) from the above model, we use a 6-fold cross validation to compute the average square error (cross-validation error) which means that we will divide the 300 samples into 6 subsets with each containing 50 samples and we will use each subset as the validation set while the others as the training set to compute the prediction error (which is the average value of $(y - X\hat{\beta})^2$, here the $\hat{\beta}$ is the fitted coefficient vector for a certain method) . The average square error will then be the average value of the six prediction errors computed previously. Moreover the values of θ (the thresholding parameter for the supervision step of SRR) should also be chosen to be the one that minimizes the cross-validation error (see details in section (4.2)). As we have already pointed out in 4.2 that

many values for θ could correspond to the same number of predictors chosen by SRR, we will plot the cross-validation error (expected prediction error) against the number of predictors chosen in the supervision step (we will plot the instance when the noise variance is 14, which is a significant noise level under this case). The actual number of predictors chosen should be the one that minimizes the prediction error. For convenience's sake we will refer to this graph as the EPE-V-THETA graph.

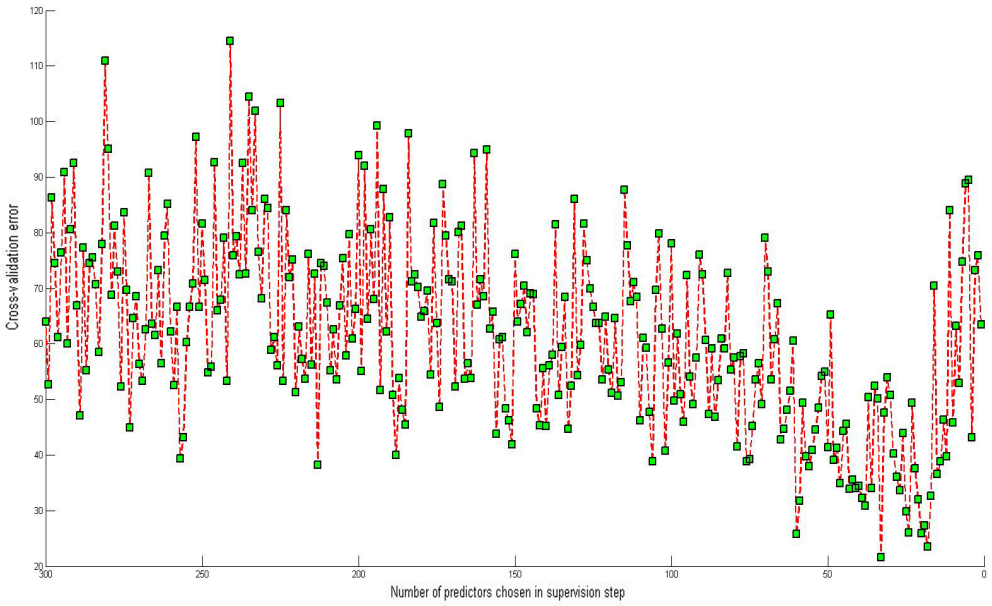

Fig. 5.1: EPE-V-THETA

From the figure above, we can see that the cross-validation error reaches the lowest value when we choose the 38 predictors whose uni-variant correlation with the response are larger than those that are not chosen. Any θ which results in exactly 38 predictors satisfying $|s_j| > \theta$ is optimal. As a result the 38 predictors will be chosen to form the reduced data matrix and perform the following-up ridge regression.

Next we shall plot the expected square prediction error against the variance

of the additive noise and show how this noise will affect the prediction of different methods, and we will call this graph the EPE-v-N graph for convenience's sake. The following figure shows what we have described:

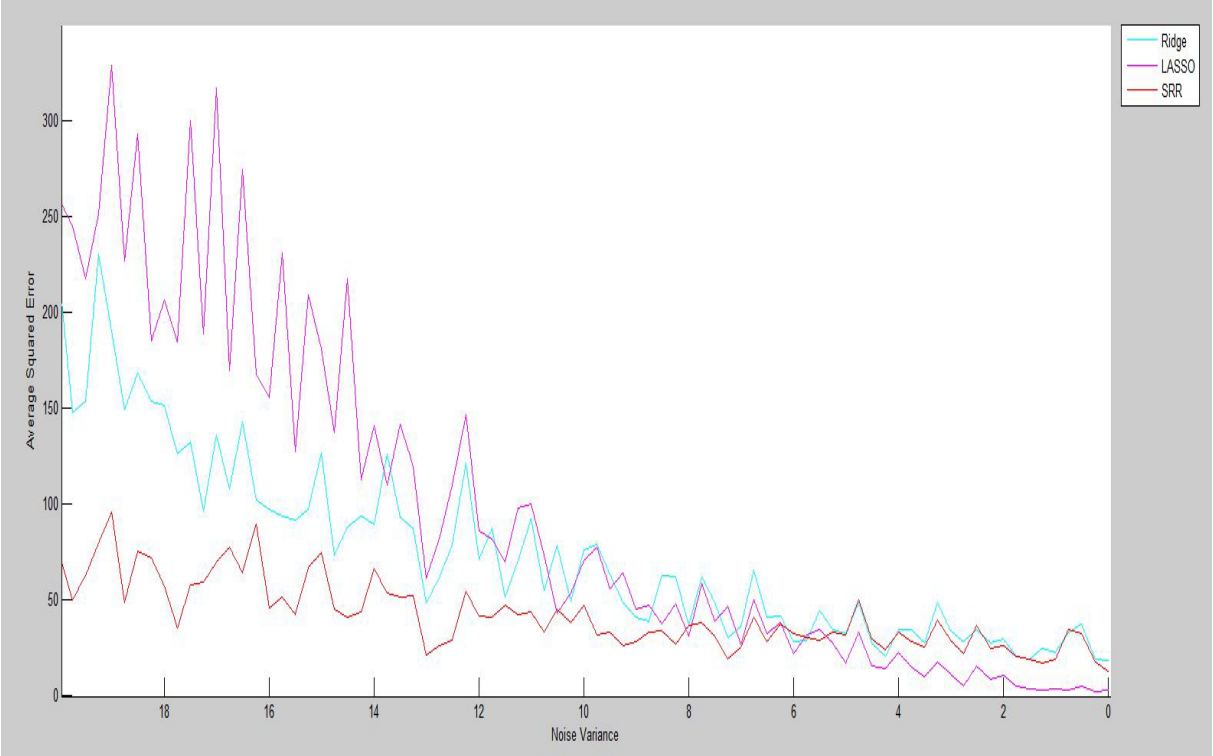

Fig. 5.2: EPE-V-N for SRR, LASSO and Ridge

Here we plotted the EPE-v-N graphs for all three different methods on the same picture with purple standing for LASSO, teal standing for ridge and red standing for SRR. We can see that when the variance of the additive noise is small, the prediction errors of all three methods are small. LASSO seems to deliver quite satisfactory prediction performance and SRR only slightly outer-performs ridge when the noise is small. The difference between these three methods starts to reveal itself when the noise level increases. LASSO as what we have explained in previous chapters, starts to become more and more unstable and the prediction error increases rapidly when the noise variance increases. On the other hand the prediction error of ridge regression also increases (since ridge regression doesn't

choose predictors, larger noise will make ridge regression more vulnerable to the effect of spurious predictors) but in a less aggressive way than LASSO, which then results in ridge's better performance over LASSO when the noise becomes large. Finally, we can see that the SRR's prediction error increases very slowly when the noise variance increases and an obvious improvement over both LASSO and ridge can be observed. The above simulation result demonstrates our claim about SRR's robustness to noise in high-dimensional linear regressions, the prediction performance of SRR also as expected dominates both ridge and LASSO.

In addition, SRR not only has better prediction performance over ridge and LASSO, it also does better in terms of producing a sparse interpretable model when the noise is large. The improvement of interpretability over ridge regression is obvious since we have already mentioned early on that ridge will select all of the 800 predictors into the resulting model. This means that the regression coefficients of ridge for all of the 800 predictors will be non-zero, which is actually a model without any reasonable interpretability. Let's compare SRR and LASSO in terms of how many predictors they select into the model and how many of the selected predictors come from the true set of the first 50 true predictors. Taking the instance where the noise variance is 14 for illustration, SRR selects 38 predictors (through the supervision steps) out of the 800 predictors, all of the selected predictors comes actually from first 50 true predictors. Even though there are still quite a few true predictors not chosen, this result is already acceptable enough as compared to what LASSO offers. Under this same noise level the LASSO selects as many as 139 predictors and only 18 of those selected predictors are actually true. This poor selection performance of LASSO can be easily explained in this situation. Since there are a lot of correlated predictors in the first 50 true predic-

tors and LASSO only tends to select one out of a group of correlated predictors, so only a small part of the true predictors will be actually selected by LASSO. Moreover, due to the significance of the noise and LASSO's sensitivity to it in high-dimensional problems, it tends to select a lot of spurious predictors into the model (recall in 3.3 LASSO can easily become inconsistent when noise is large). SRR on the other hand preserves groups of correlated true predictors and filters out most of the spurious predictors during the supervision step, so the variable selection quality of SRR also dominates both ridge and LASSO when the additive measurement noise is large.

Up to this point we have demonstrated through the above simulation example about our motivations and rationales for the supervised ridge regression. We have shown that the SRR preserves the advantages for both LASSO and ridge by producing a sparse interpretable model. The most important improvement for SRR over LASSO and ridge is its robustness against the variance of the additive noise, the prediction error for SRR also increases with the enlarging of the noise but comparably in a much slower speed than ridge and LASSO. In later sections we will perform a number of further tests to find possible limitations of SRR.

5.2 More Experiments

So far, we have illustrated the satisfying performance of SRR under a few assumptions for the high-dimensional Gaussian linear model (for example, we assume that the spurious and true predictors are completely independent with respect to each other). This illustrative example alone is still not sufficient for us to draw convincing arguments. So in this section we will focus on each critical assumptions and conditions we set for SRR and try to test the significance of each of the

assumptions for the performance of SRR.

5.2.1 Correlated Spurious and True Predictors

We have been assuming that the spurious predictors and true predictors are ideally separated (completely uncorrelated). In the supervision step for SRR we are trying to distinguish spurious and true predictors based on their uni-variant correlation with the response. As a result, the ideal-separation assumption of spurious and true predictors should improve the significance of the supervision step in SRR. So in this modification, we design an experiment where we add a significant pairwise correlation between a small number of spurious predictors and true predictors to test the sensitivity of SRR's variable selection and prediction performance with respect to the correlation between spurious and true predictors.

More precisely, we add a pairwise correlation of 0.9 from x_{40} to x_{80} while all the other settings remain the same with respect to the original model. Under this modification, we expect more predictors to be chosen by SRR, since inevitably some of the spurious predictors (those correlated with true predictors) will be chosen into the model because they might also have a significant correlation with the response. First of all, to see how this change can affect the variable selection performance of SRR, we will plot the EPE-V-THETA graph under a specific noise level (we will still choose a relatively large noise level of 14 in order to be consistent with the previous case).

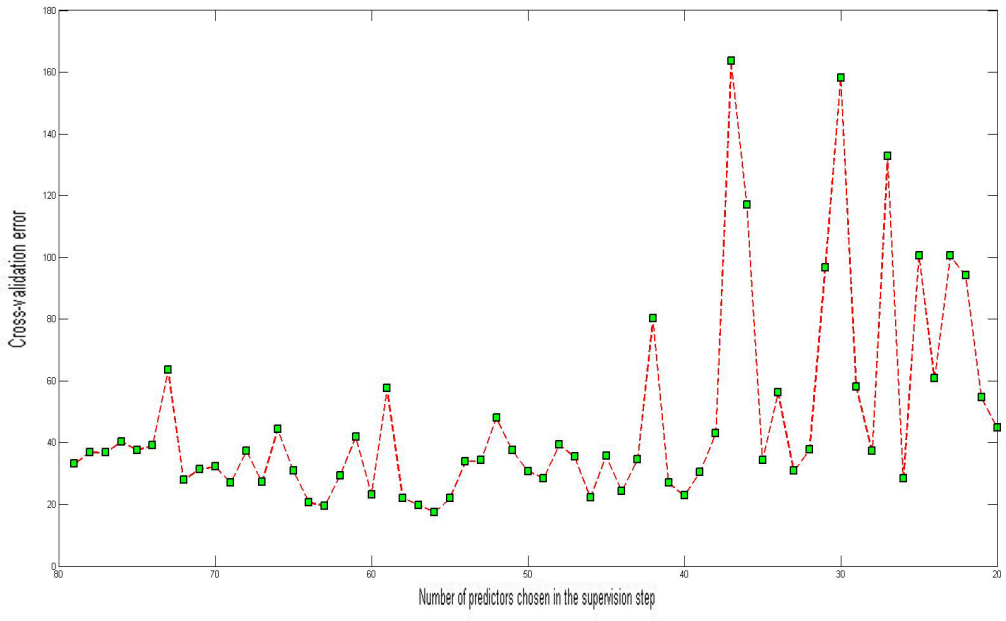

Fig. 5.3: EPE-V-THETA2 in modified case 1

In this case, we choose 57 predictors in the first supervision step that gives us the smallest cross-validation error. Out of the 57 predictors chosen, 37 of them are true predictors while others are spurious. This performance is no surprising to what we have expected. Due to the correlation between the two kinds of predictors and the existence of the additive noise, the uni-variant correlation between some of the spurious predictors and the response may be close to or even larger than some true predictors. As a result the SRR will select some of the spurious predictors into the model. Nevertheless, SRR still gives a better variable selection performance than the LASSO which selects up to 134 predictors in this situation.

Next, it is also important to investigate how the correlations between spurious and true predictors will affect the prediction performance of SRR. In order to do this, we will plot the EPE-V-N graph for this correlated true and spurious predictor case and see how the expected prediction error for the three methods

vary with the noise variance:

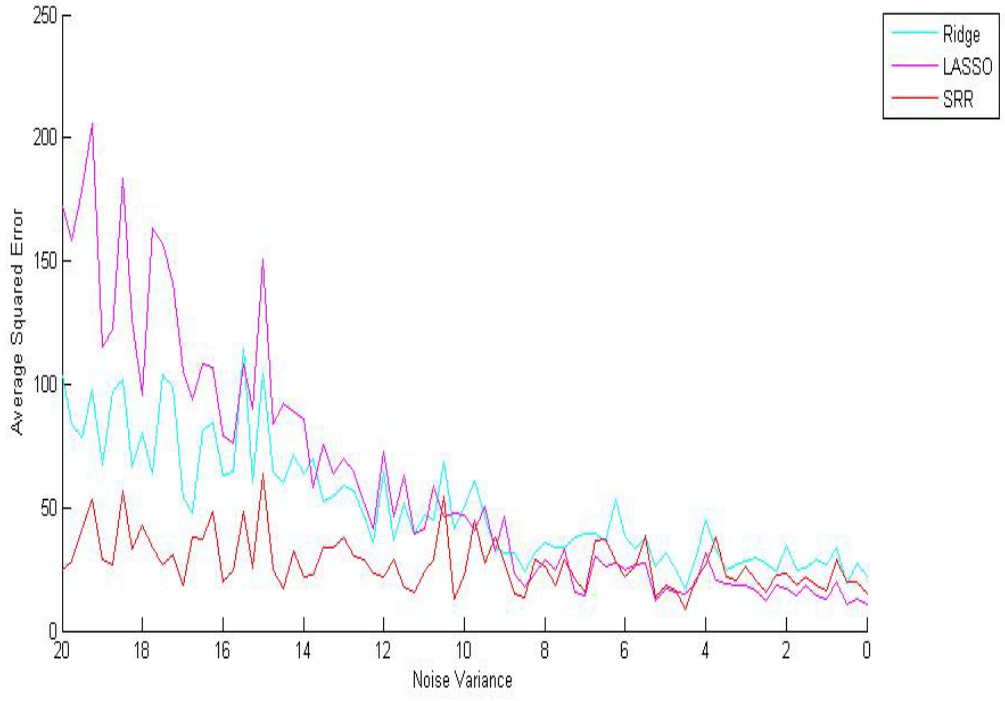

Fig. 5.4: EPE-V-N for SRR, LASSO and Ridge in modified case 1

From the above figure, we can see that the prediction performance of SRR still starts to dominate both ridge and LASSO when the noise variance becomes larger and larger and the general behavior of all of the three methods are not significantly different from the original case (the first illustrative example). More precisely, the prediction performance for SRR has been degraded very slightly as compared to the original case while the LASSO and ridge somehow have even smaller prediction error. As a result we can see that in the original model SRR starts to dominate when the noise variance exceeds 9 while in this case it starts to dominate when the noise variance exceeds 11. In a word, both the prediction and variable selection accuracy of SRR are affected by the correlations between spurious and true predictors. The effect on the variable selection accuracy is more significant, the more the correlations are, the more the number of spurious

predictors will be chosen into the model regardless of the noise level (ie, when compared to the original case, under the same noise level this modified case will choose more predictors into the model). To demonstrate this point more clearly, we now try to look at an extreme case where a large of number spurious predictors are correlated with the true predictors. We add a pairwise correlation of 0.9 from x_{40} to x_{300}. Now, the resulting EPE-V-THETA graph is as follows:

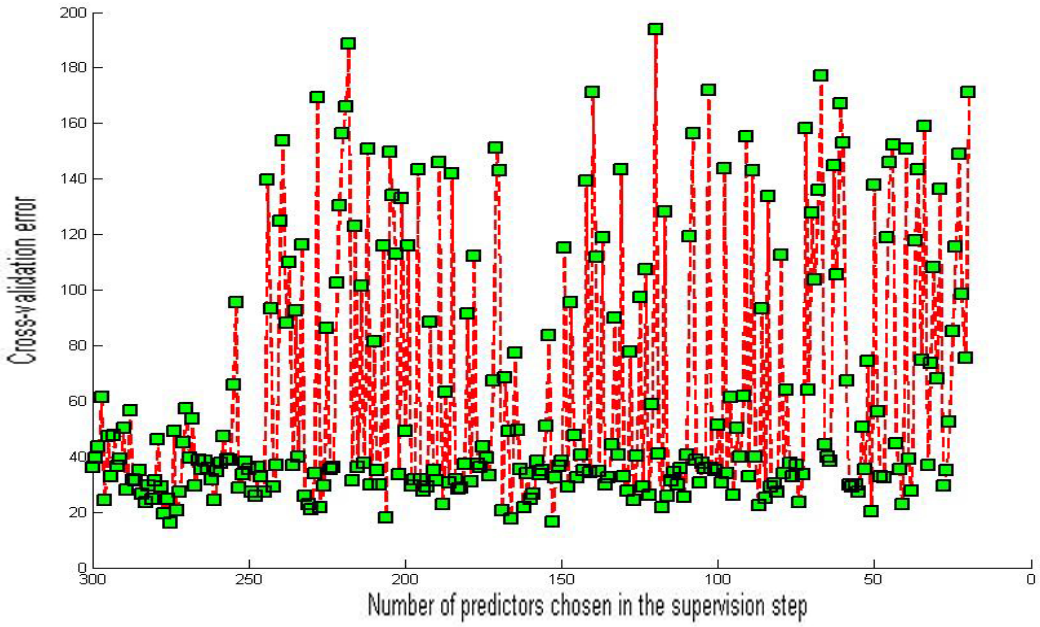

Fig. 5.5: EPE-V-THETA in modified case $1^{'}$

Due to this strong correlation between true and spurious predictors, we can see that our cross-validation error stabilizes to a low level only after we have chosen more than 250 predictors, which actually results in a lot of mistakes in terms of variable selection accuracy. On the other hand, when we have already chosen 250 predictors into the model the cross-validation error is only around 40, which is very close to the prediction error in the original case when the noise variance is 14. So we can see that the correlations between spurious and true predictors have only a very slight effect on the prediction accuracy of SRR. Nevertheless, SRR

performs better in cases where spurious and true predictors are not correlated. Let's look at another variation of the original model where the number of samples is smaller than the number of true predictors.

5.2.2 Insufficient Amount of Data Samples

Let's recall in our original definition of a high-dimensional linear model in (3.1), we assumed that the number of data samples is much smaller than the total number of predictors while being larger than the number of true predictors. This original assumption was made mainly to ensure that we have sufficient amount of data to reasonably recover the true regression model. In this modified experiment we will test the robustness of SRR under the situation where the amount of data is not sufficient ($N < m$). The modified model will preserve all the settings of the original model while the only difference is that the sample size N is 50 instead of the original 300 (The number of true predictors in the original model is 50, and some part of the 50 samples will be used as the validation set so in the training set we actually have $N < m$). We will use a 5-fold cross-validation to estimate the threshold parameter θ and the prediction error. So the training sample size is only 40. To evaluate the performance change of SRR under this situation we will use the same approach as before, i.e., we will first test the change on variable selection accuracy and then test the change on the prediction performance. So as the first step, the EPE-V-THETA graph for SRR at the noise level 14 will be plotted to see how many predictors the supervision step would tend to choose when there are not enough samples:

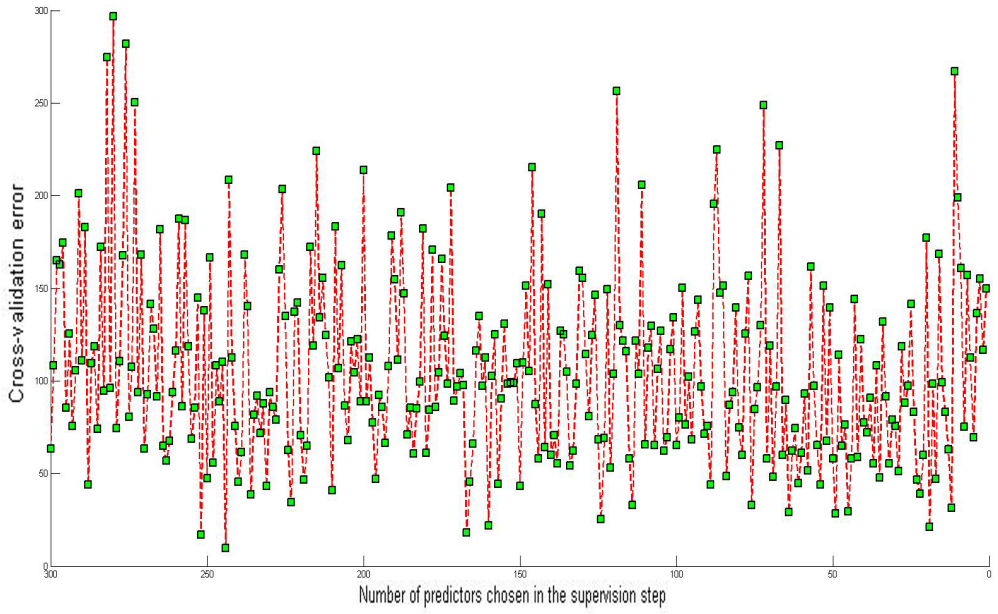

Fig. 5.6: EPE-V-THETA in modified case 2

From the above graph we can see that unlike the previous few cases, there is no obvious stable minimum for any range of predictors chosen and it becomes very difficult to determine exactly how many predictors should be chosen in the supervision step. This nondeterministic behavior is reasonable since the lack of information will result in spurious correlations between the response and spurious predictors thus making it very difficult to distinguish between true and spurious predictors. As a result the behavior of SRR in this situation is very random and problematic. Of course, due to this lack of sample points, ridge and LASSO also behave very poorly. Let's again plot the EPE-V-N graph, which will demonstrate more clearly how undesirably the performance for all of the three methods could be under this situation.

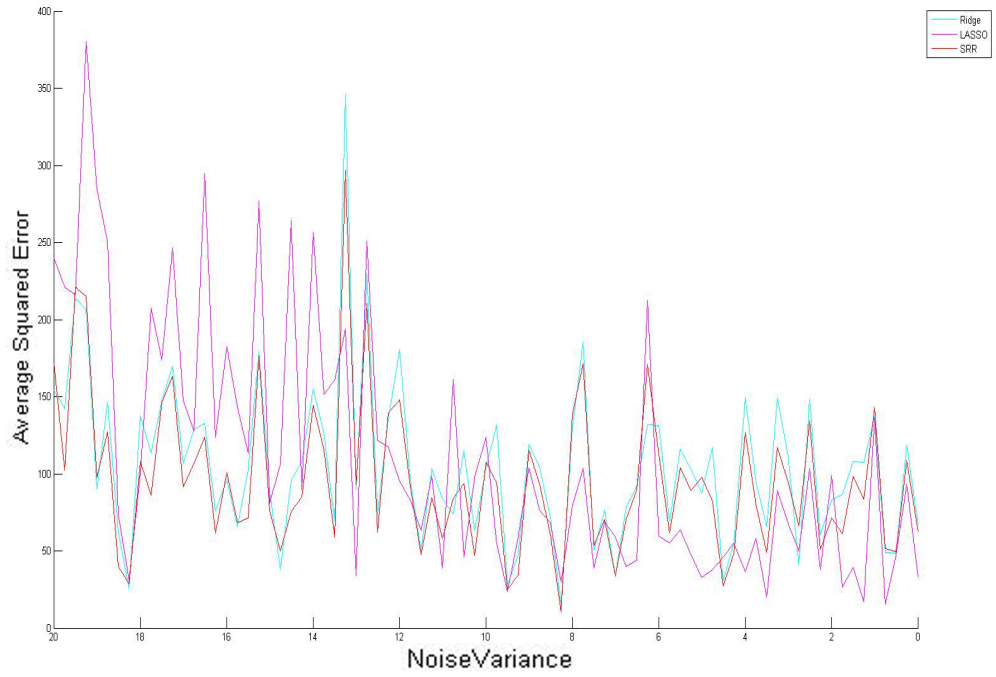

Fig. 5.7: EPE-V-N for SRR, LASSO and Ridge in modified case 2

Here we can see that the expected prediction errors for all three methods are very large even when the noise is small. The variable selection accuracy for LASSO is also very low. For instance, when noise variance is 14, LASSO selects 34 predictors out of which only 2 are actually from the true set of 50 predictors. As for SRR, out of the 22 chosen predictors only 8 of them are actually true predictors. The other 14 chosen spurious predictors have some spurious high correlation with the response due to the lack of data samples. These kinds of poor behavior for all of the three methods are easy to understand since we don't have enough information to recover a reasonably true model thus it is quite acceptable for any method to behave poorly under this situation (Since we are trying to do regression and predict the future based on the past data using regression. Without enough information for the past it is difficult to build a reasonably good prediction model). In a word, from what we have observed in the above test,

when the number of samples is smaller than the number of true predictors, the set of predictors SRR chooses in the supervision step tends to be very random and moreover the prediction error is also very large.

5.2.3 Low Dimensional Problem

Recall that the typical type of problem for which we propose the supervised ridge regression is based on the high-dimensional assumption $p >> N > m$, and SRR has been demonstrated to be able to dominate both ridge regression and LASSO in this situation when the noise is large. It is interesting to see whether SRR can retain its significantly better performance over both ridge regression and LASSO under a usual low dimensional problem. To perform this experiment we make a simple modification to the original model. Also it is important that p is larger than m, in order to keep the significance of the supervision step of SRR since it would become meaningless to try to filter out any predictor when all of them are actually true. Thus, the actual relationship between m, p, and N is $N > p > m$. Particularly, in this experiment we let $p = 100$, which is smaller than $N = 300$ and greater than $m = 50$, and all the other settings will remain the same with respect to the original model. Same as before, we will first plot the EPE-V-THETA graph and see how many predictors will SRR choose during the supervision step (In order to be consistent with the previous few cases, we also only show the particular case where the noise variance is equal to 14):

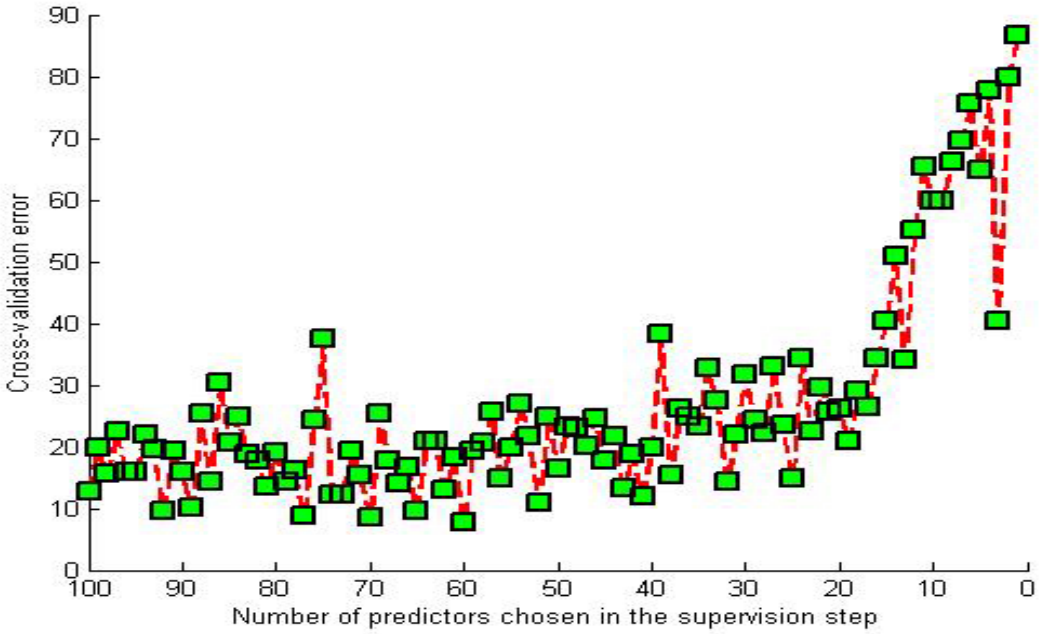

Fig. 5.8: EPE-V-THETA in modified case 3

We can see that the cross-validation error rapidly stabilizes to a very low level, in this case 64 predictors were selected in the supervision step. We can see that even if all the 100 predictors were selected into the model (which would just correspond to the ridge regression) the prediction error is still small, which actually means that under this low dimensional situation the effect of the supervision step is not as effective as it is in the high dimensional problem. To look at the prediction performance under this low dimensional situation, let's plot the EPE-V-N graph for LASSO, ridge regression and SRR:

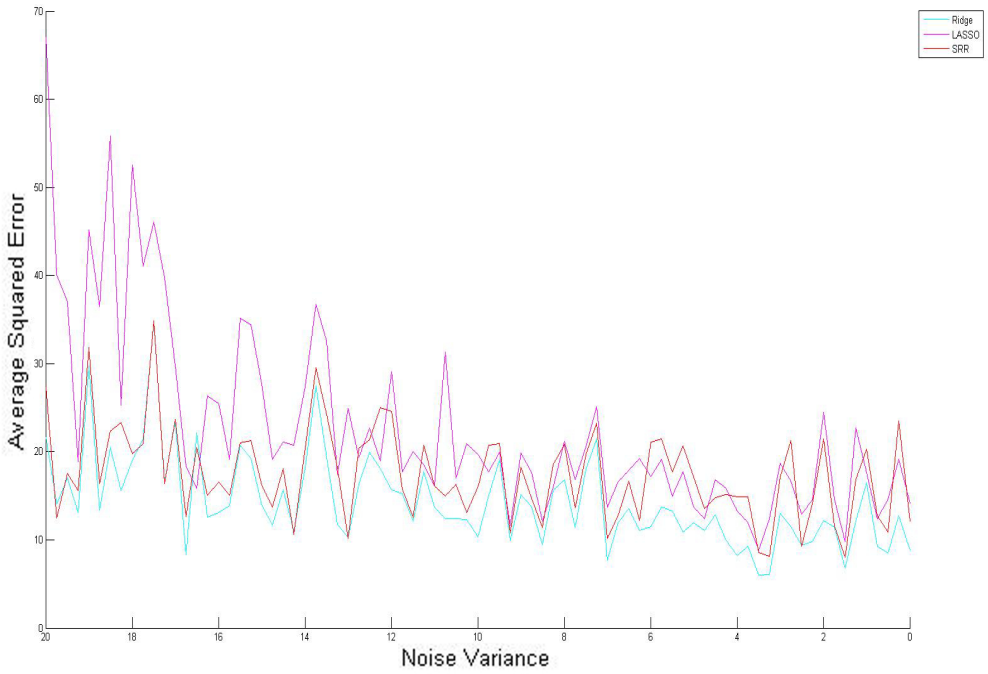

Fig. 5.9: EPE-V-N for SRR, LASSO and Ridge in modified case 3

From the above graph, we can see clearly that SRR in this low dimensional problem does not significantly dominate the performance of either ridge regression or LASSO. In terms of variable selection, for the case where noise variance is 14 LASSO selects 36 predictors in total and 24 of them are true while SRR selects 64 predictors and 44 of them are true. This is still a reasonable improvement but it is already far less significant as compared to what happened in the high-dimensional case. For the prediction error, we can see that the ridge regression somehow has an even slightly better performance. From the observations above, we can see that we do need a high dimensional situation for the advertised effects of the supervised ridge regression to take place. Nevertheless, we propose the SRR specifically for high-dimensional problems mainly because the effect of the supervision step will become significant in this situation and SRR is more robust to the additive noise than LASSO and ridge regression when dimension is high.

As a result, it is not a surprise that the performance of SRR is not too different from ridge regression and LASSO when the dimension is low.

6. CONCLUSIONS AND DISCUSSIONS

6.1 Conclusions

In this section, we will present a brief summary of the properties of the SRR introduced in this thesis. The proposed SRR applies ridge regression on a reduced data matrix $\mathbf{X}_\theta$ that is determined from a supervision step which chooses predictors based on their uni-variant correlation with the response. We have demonstrated that the SRR is able to compensate for both the limitations of ridge and LASSO. Moreover, the improvement is more significant when the noise variance is larger. The advantages of SRR over both of the methods include better prediction performance and better variable selection accuracy. To evaluate the performance of SRR, we did a series of experiments based on different model assumptions and conditions. Here we will summarize the conclusions drawing from the observations of those experiments.

1. When the spurious and true predictors are not ideally separated, the correlation between these two kinds of predictors will result in spurious high correlations between the response and some spurious predictors. Such correlation will affect the selection accuracy in the supervision step because those spurious predictors could be chosen into the model. On the other hand, the prediction performance of SRR is only degraded slightly due to this correlation between spurious and true predictors. Therefore even though many spurious predictors might be chosen into the regression model doing regression and prediction includ-

ing those spurious predictors will not result in a much larger prediction error.

2. When the number of data samples are not sufficient (ie, when $p >> m > N$), the variable selection step for SRR (namely, the supervision step) behaves in a very random and nondeterministic fashion, and many spurious predictors can be chosen into the model while only a small number of true predictors are actually chosen. Also, the prediction performance of SRR under this situation is very poor and not any significant improvement over ridge regression or LASSO is observed. Thus, to recover a reasonably good model (fair interpretability and prediction accuracy) using SRR, we need to at least make sure that the number of samples is no less than the number of true predictors.

3. In a low-dimensional problem (namely the relationship between N, m, p is $N > p > m$), the variable selection steps for both LASSO and SRR have very good accuracy and SRR only out-performs LASSO a little bit when the noise variance is relatively large. This improvement is much less significant as compared to what has been observed in a high-dimensional problem. Moreover, the prediction performance of the SRR is not too different from both ridge regression and LASSO. Under our particular test example, ridge regression has an even slightly smaller prediction error. Thus, the SRR is especially useful for high-dimensional problems.

When we compared the mechanisms of variable selection for LASSO and SRR we have found that the first supervision step of SRR is quite similar to the first iteration of the coordinate descent algorithm for LASSO (which is actually one of the reasons why SRR is more robust to the noise since the noise is only computed once). Thus the SRR can be regarded as an implicit combination of part of LASSO with ridge regression, which turns out to be able to beat both methods

when the additive noise is large and the dimension is high. There are also some other ways to combine ridge and LASSO, one very popular combination is called the elastic-net which combines the L_1 and L_2 penalties together, interested readers can refer to [8] for more information.

Moreover, despite all of the reasoning and rationales we provided to demonstrate the functionality of the supervised ridge regression, we didn't show sufficient mathematical analysis for the asymptotical behavior and the conditions of the variable selection consistency for our technique. Detailed mathematical proofs and analysis are also an important direction for further research work.

6.2 References and Related Works

[1] Arthur E. Hoerl, Robert W.Kennard, "Ridge Regression:Biased Estimation for Nonorthogonal Problems", University of Delaware and E.I.du Pont de Nemours, Co. 1994

[2] Alan J. Miller, "Selection of Subsets of Regression Variables", J.R.Statist.Soc A, 147,part 3,pp. 389-425. 1984

[3] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso", J.R.Staistist.Soc B, Vol 58, Issue 1, 267-288. 1996

[4] B.Efron, T.Hastie, I.Johnstone and R.Tibshiran, "Least Angle Regression", The Annals of Statistics, Vol.32, No.2, 407-499. 2004

[5] J.Friedman, T.Hasitie, and R.Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent", J Stat Softw,33(1):1-22. 2010

[6] B.Efron, T.Hastie, D.Paul, and R.Tibshirani, "Prediction by Supervised Principal Components", JASA, Vol. 101, No. 473. 2006

[7] Martin J.Wainwright, "Sharp thresholds for high-dimensional and noisy

sparsity recovery using l1-constrained quadratic programming", J. IEEE Transactions on Information Theory, Vol 55 Issue 5. 2009

[8] Zhou, Hui, T.Hastie and Trevor, "Regularization and Variable Selection via the Elastic Net", J.R.Statist.Soc B,pp. 310-320. 2005

[9] F.Bunea, Marten H.Wegkamp and A.August, "Consistent variable selection in high-dimensional regression via multiple testing", J.Statistical Planning and Inference DOI:10.1016, 2005

[10] J.Huang, S.Ma and Z.Cun-hui "Adaptive LASSO for sparse high-dimensional regression models", University of Iowa department of Statistics and Actuarial Science. 2006

[11] L.Runze and J.Q.Fan, "Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties", J.A.Statist.Soc vol 96, Issue 456. 2001

[12] K.Knight and W.Fu, "Asymptotics for lasso-type estimators", Ann. Statist. 28 1356-1378. 2000

[13] David L.Donoho and J.Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing", Phil.Trans.R.Soc.A 367, 4273-4293, 2009