8-2007

Model Selection Techniques for Kernel-Based Regression Analysis Using Information Complexity Measure and Genetic Algorithms

Rui Zhang
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Rui Zhang entitled "Model Selection Techniques for Kernel-Based Regression Analysis Using Information Complexity Measure and Genetic Algorithms." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Business Administration.

Hamparsum Bozdogan, Major Professor

We have read this dissertation and recommend its acceptance:

Mary Leitnaker, Myong K. Jeong, Russell Zaretzki

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Rui Zhang entitled "Model Selection Techniques for Kernel-based Regression Analysis Using Information Complexity Measure and Genetic Algorithms." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Business Administration.

Hamparsum Bozdogan

Major Professor

We have read this dissertation
and recommend its acceptance:

Mary Leitnaker

Myong K. Jeong

Russell Zaretzki

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of
Graduate School

(Original signatures are on file with official student records.)

Model Selection Techniques for Kernel-Based Regression Analysis Using Information Complexity Measure and Genetic Algorithms

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Rui Zhang

August 2007

Dedication

This dissertation is dedicated to my wife Yan Liu and my parents who have supported and encouraged me to persevere through challenges.

Acknowledgments

I am deeply indebted to all those who helped in completing this work.

I would like to thank my advisor Dr. Bozdogan for introducing me to the field of information complexity measure, to various areas of statistical modeling, and the problem of this thesis in kernel methods.

I would further like to thank the other members of my committee: Dr. Leitnaker, Dr. Zaretzki and Dr. Jeong for helpful discussions and useful criticism.

Finally, I would like to express my appreciation to my family whose suggestions and encouragement made this work possible.

Abstract

In statistical modeling, an overparameterized model leads to poor generalization on unseen data points. This issue requires a model selection technique that appropriately chooses the form, the parameters of the proposed model and the independent variables retained for the modeling. Model selection is particularly important for linear and nonlinear statistical models, which can be easily overfitted.

Recently, support vector machines (SVMs), also known as kernel-based methods, have drawn much attention as the next generation of nonlinear modeling techniques. The model selection issues for SVMs include the selection of the kernel, the corresponding parameters and the optimal subset of independent variables. In the current literature, k-fold cross-validation is the widely utilized model selection method for SVMs by the machine learning researchers. However, cross-validation is computationally intensive since one has to fit the model k times.

This dissertation introduces the use of a model selection criterion based on information complexity (ICOMP) measure for kernel-based regression analysis and its applications. ICOMP penalizes both the lack-of-fit and the complexity of the model to choose the optimal model with good generalization properties. ICOMP provides a simple index for each model and does not require any validation data. It is computationally efficient and it has been successfully applied to various linear model selection problems. In this dissertation, we introduce ICOMP to the nonlinear kernel-based modeling areas. Specifically, this dissertation proposes ICOMP and its various forms in the area of kernel ridge regression; kernel partial least squares regression; kernel principal component analysis; kernel principal component

regression; relevance vector regression; relevance vector logistic regression and classification problems. The model selection tasks achieved by our proposed criterion include choosing the form of the kernel function, the parameters of the kernel function, the ridge parameter, the number of latent variables, the number of principal components and the optimal subset of input variables in a simultaneous fashion for intelligent data mining.

The performance of the proposed model selection method is tested on simulation benchmark data sets as well as real data sets. The predictive performance of the proposed model selection criteria are comparable to and even better than cross-validation, which is too costly to compute and not efficient.

This dissertation combines the Genetic Algorithm with ICOMP in variable subsetting, which significantly decreases the computational time as compared to the exhaustive search of all possible subsets. GA procedure is shown to be robust and performs well in our repeated simulation examples.

Therefore, this dissertation provides researchers an alternative computationally efficient model selection approach for data analysis using kernel methods.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

This research proposes a novel model selection criterion for support vector machines (SVMs) models (Vapnik, 1995). SVMs, also known as the kernel-based methods (Shawe-Taylor and Cristianini, 2004), are a set of nonlinear statistical learning techniques that have drawn much attention since the mid 1990s. SVMs were developed at AT&T Bell Laboratories by Vapnik and his co-workers (Vapnik, 1995). Kernel-based methods (KMs) have been widely used in different nonlinear modeling areas including regression (Rosipal and Trejo, 2001; Tipping, 2001), principal component analysis (Schölkopf et al., 1998), canonical correlation analysis (Akaho, 2001; Shawe-Taylor and Cristianini, 2004), discriminant analysis (Mika et al., 1999), clustering and classification.

The basic idea of KMs is to nonlinearly map the independent variables to a high-dimensional or even infinite-dimensional Hilbert space (Hilbert, 1927). This transformed space is generally called feature space in machine learning language. After the transformation, the traditional linear learning techniques are applied to the feature space. Such nonlinear transformation is conducted through the 'kernel tricks' (Aizermann et al., 1964). Using the kernel matrix, it is not necessary to know the explicit form of the feature space. One just needs to select the appropriate kernel function and 'tune' its parameters to prevent

the overfitting problem in KMs. KMs have become the next generation of nonlinear machine learning techniques that took the place of neural networks (NNs) due to the following advantages:

1. The kernel-based models are simple. In NNs, one has to conduct preliminary trials to design the number of layers, the number of neurons in each layer and the optimization methods.

2. Training NNs is slow since it requires the gradient-search type nonlinear optimization. Many KMs do not require iterative procedures.

3. KMs are more efficient when the number of independent variables is bigger than the number of training observations.

4. Training NNs models may suffer from falling into the local minimum which depends on the initialization methods. KMs do not suffer from this limitation.

1.2 Statement of Problem

Model selection is one of the open questions of KMs. First, one must use the appropriate kernel function. For instance, the Gaussian Radial Basis Function (RBF) kernel is the most popular kernel that can provide a complicated nonlinearity while the polynomial kernel is simple but can only provide limited nonlinearity. Second, the scale parameter of the kernel function needs to be "tuned" to achieve good generalization. For the Gaussian RBF kernel, a large scale parameter will lead to underfitting and a small scale parameter will lead to overfitting. Increasing the order of the polynomial kernel provides higher nonlinearity.

In addition, the parameters of the modeling methods must be appropriately selected to prevent overfitting. Such model selection issues that will be discussed in this dissertation include choosing the ridge parameter, choosing the number of latent variables and choosing the number of principal components.

When multiple independent variables are involved, statisticians would be interested in knowing which variables are critical and which variables are nuisance. This will lead to an important model selection topic - subset variable selection. The KMs literature in this area is limited since the machine learning researchers are generally focused on tuning the kernel parameters. Our numerical experiments indicate that including nuisance variables may lead to a poor predictive ability for future observations. From a statistician's point of view, selecting the appropriate subset of variables also provides a good interpretation of the fitted model.

Currently in the literature, error of the validation data is used to evaluate the generalization of the fitted model. When the number of observations is large, one can use the hold-out sample to evaluate the fitted model. For relative small data sets, the k-fold cross-validation is superior (Goutte, 1997). It estimates the generalization error based on re-sampling. In k-fold cross-validation, one divide the data into k groups of equal size. The model is fit k times. In each fit, one group is excluded during the training and the fitted model is used to compute the error of the omitted group. If k is equal to the sample size, this is called leave-one-out cross-validation (LOOCV). A value of 10 for k is popular for estimating generalization error. The drawback of k-fold cross-validation is in its computational intensity, especially for the LOOCV approach. LOOCV also suffers from the continuity, which means a small variance in the data can lead to a large change in the selected model (Breiman, 1996).

An intuitive thinking based on the above shortcomings is that it is computationally efficient to utilize a model selection criterion which does not require extra validation data. One of such criteria is Akaike information criterion (AIC) (Akaike, 1973). AIC is consisted of two parts, a function of the maximum likelihood to measure the lack-of-fit of the model and a penalty term to measure the complexity of the model. In AIC, the penalty term is two times the number of estimated parameter. The optimal model is the minimizer of AIC that provides good generalization as well as good fit. However, it is questioned (Rissanen, 1976; Bozdogan, 1988) that counting the number of parameters in AIC and its variants do

not provide sufficient penalty. Inspired by AIC, Bozdogan (1988) proposed the information complexity (ICOMP) measure. Instead of penalizing the number of estimated parameters, ICOMP penalizes the interdependency among the estimated parameters. This criterion provides a more judicious penalty term than AIC and other AIC-type criteria. It has been successfully utilized in different linear modeling applications. This dissertation extends the use of ICOMP to the kernel-based methods based on the fact that linear model methods are utilized in the feature space.

1.3 Contributions

This dissertation contributes to the existing machine learning literature by utilizing ICOMP and its variants as the alternative model selection criteria for various kernel-based regression analysis problems without using validation data.

This dissertation provides ICOMP forms to be used for the kernel ridge regression, kernel least squares regression, kernel principal component analysis, kernel principal component regression, relevance vector regression and relevance vector logistic regression respectively. The proposed model selection method is used to choose the optimal parameters of a kernel function and compare between different kernel functions for the above regression analysis applications.

Given a fixed kernel function, this dissertation uses ICOMP to choose the optimal ridge parameter of the kernel ridge regression, the optimal number of latent variables of kernel least squares regressions or the optimal number of kernel principal components.

This dissertation also utilizes ICOMP as the criterion to choose the optimal subset of independent variables in the kernel-based methods listed above. When the number of total independent variables are large, genetic algorithm is used to search the optimal subset efficiently using a function of ICOMP as the measure of fitness.

Last, this dissertation derives the point and interval estimates of the weighted kernel ridge regression to solve the heteroscedasticity problem.

1.4 Organization of Dissertation

The rest of the dissertation is organized as follows. Chapter 2 is an introduction to kernel-based methods. Chapter 3 describes the information measure approach to model selection based on the work of Bozdogan(1988; 1990; 1998; 2004a). The forms of ICOMP for different kernel-based methods will be detailed in the subsequent individual chapters. Chapter 4 briefly describes genetic algorithm that is used for subset model selection. Chapter 5 to Chapter 8 present the ICOMP-based model selection approaches for kernel ridge regression, kernel partial least squares regressions, kernel principal component analysis and regression and relevance vector machines respectively. Chapter 9 concludes the dissertation and suggests possible future work and improvements in this area of research.

Chapter 2

Overview of Kernel-based Methods

2.1 Introduction of Kernel-based Methods

Kernel-based methods (KMs) are a set of nonlinear statistical modeling techniques that have received much attention recently from the machine learning literature. KMs first appeared and became popular in the form of support vector machines (SVMs), originally proposed by Vapnik (1995). The basic idea of kernel-based algorithms is to first nonlinearly map the original data space $\mathbf{X} \subseteq \mathbf{R}^p$, called input space, to a high-dimensional feature space $\Phi(\mathbf{X}) \subseteq \mathbf{R}^k$. Then, the traditional linear learning algorithms are applied on the feature space.

Instead of giving the exact form of the high-dimensional $\Phi(\mathbf{X})$ in the feature space directly, KMs use the kernel trick, first introduced by Aizermann, Bravermann and Rozoener (1964) on the method of potential functions to avoid the "curse of high dimensionality". The kernel trick uses kernel functions to perform the nonlinear transformation and the explicit dimension and form of the feature space is unnecessary to be known. A kernel is a function k that for any two observations $\mathbf{x}$ and $\mathbf{z}$ of $\mathbf{X}$, satisfies

$$k(\mathbf{x}, \mathbf{z}) =< \Phi(\mathbf{x}), \Phi(\mathbf{z}) > \tag{2.1}$$

where $\Phi(\cdot)$ represents the mapping from the input space $\mathbf{X}$ to the feature space. According to the geometric meaning of inner product, a kernel function is actually calculating the squared distance of two observations in the feature space. Therefore, the kernel function is a measure of similarity in the feature space.

A kernel matrix $\mathbf{K}$, also called gram matrix, derived from the p-dimensional training data $\mathbf{X} \subseteq \mathbf{R}^p$ with n observations is a $n \times n$ finitely positive semi-definite matrix whose ij^{th} element k_{ij} is the inner product of the i^{th} feature space observation $\Phi(\mathbf{x_i})$ and the j^{th} feature space observation $\Phi(\mathbf{x_j})$. That is,

$$k_{ij} =< \Phi(\mathbf{x_i}), \Phi(\mathbf{x_j}) >= k(\mathbf{x_i}, \mathbf{x_j}). \tag{2.2}$$

The kernel matrices in KMs are equivalent to the covariance matrices in the original data space. They contains all the information needed for modeling purposes. They also can be thought of as the training data matrix, like $\mathbf{X}$ from the input space, when the linear learning is conducted in the feature space. A separable and complete inner product space is called a Hilbert space (Hilbert, 1927), a generalization of the vector space that is not restricted to finite dimensions. Since the kernel function $k(\mathbf{x}, \mathbf{z})$ defined in (2.1) is finitely positive semi-definite (Shawe-Taylor and Cristianini, 2004), the corresponding feature space is referred as Reproducing Kernel Hilbert Space (RKHS).

We further define an $m \times n$ matrix $\mathbf{K_{new}}$ which involves both the training data $\mathbf{X}$ and the m new observations $\mathbf{X_{new}}$. The ij^{th} element of $\mathbf{K_{new}}$ is the inner product of the feature space mapping the i^{th} observation of $\mathbf{X_{new}}$ and the feature space mapping of the j^{th} observation of $\mathbf{X}$. That is,

$$k_{new(ij)} =< \Phi(\mathbf{x_{new(i)}}), \Phi(\mathbf{x_j}) >= k(\mathbf{x_{new(i)}}, \mathbf{x_j}). \tag{2.3}$$

2.2 Selected Kernel Functions

Different applications may require different kernel functions for the appropriate similarity measure. Once the kernel function is decided, one must choose the appropriate parameters to achieve the good generalization. Mercer's theorem (Mercer, 1909) is usually used to construct a feature space for a valid kernel (Shawe-Taylor and Cristianini, 2004).

In this section, we list some widely used kernel functions that will be compared in this research. Assume $\mathbf{x} \in \mathbf{R}^p$ and $\mathbf{z} \in \mathbf{R}^p$ are both p-dimensional observation vectors.

1. Polynomial Kernel Function is defined by

$$k(\mathbf{x}, \mathbf{z}) = (< \mathbf{x}, \mathbf{z} > + a)^b, \tag{2.4}$$

   where a and b are the bias parameter and the polynomial order respectively. Increasing the order b will increase the nonlinearity (or decrease the smoothness) of the resulting model. For a p-dimensional input space, the dimension of the feature space for the polynomial kernel is

$$\begin{pmatrix} p+b \\ b \end{pmatrix}.$$

   A polynomial kernel function with $a = 0$ and $b = 1$ is called the linear kernel function, which results pure linear models or working with the original data.

2. Gaussian Radial Basis Function (RBF) is defined by

$$k(\mathbf{x}, \mathbf{z}) = \exp\left[-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2a^2}\right], \tag{2.5}$$

   where a is the bandwidth (or scale) parameter that controls the smoothness of the nonlinear mapping.

3. Exponential RBF Kernel is defined by

$$k(\mathbf{x}, \mathbf{z}) = \exp\left[-\frac{\|\mathbf{x} - \mathbf{z}\|}{2a^2}\right], \tag{2.6}$$

where a is the scale parameter.

4. Linear Spline is defined by

$$k(\mathbf{x}, \mathbf{z}) = \prod_{i=1}^{p}\left[1 + \mathbf{xz} + \frac{1}{2}\mathbf{xz}\min(\mathbf{x}, \mathbf{z}) - \frac{1}{6}\min(\mathbf{x}, \mathbf{z})^3\right], \tag{2.7}$$

where p is the dimension of $\mathbf{x}$ or $\mathbf{z}$. Both $\mathbf{xz}$ and $\min(\mathbf{x}, \mathbf{z})$ are the pairwise operation instead of the matrix operation.

5. Cauchy Kernel is defined by

$$k(\mathbf{x}, \mathbf{z}) = \frac{1}{1 + \frac{\|\mathbf{x}^a - \mathbf{z}^a\|^2}{a}}, \tag{2.8}$$

where a is the scale parameter.

6. Sigmoid (Multi-Layer Perceptron) is defined by

$$k(\mathbf{x}, \mathbf{z}) = tanh(a < \mathbf{x}, \mathbf{z} > +b), \tag{2.9}$$

where a is the scale parameter and b is the bias parameter.

7. Thin-plate Spline is defined by

$$k(\mathbf{x}, \mathbf{z}) = \frac{1}{2}\eta\log(\eta + b), \tag{2.10}$$

where $\eta = a\|\mathbf{x} - \mathbf{z}\|^2$ and

$$b = \begin{cases} 0, & \eta \neq 0\ ; \\ 1, & \text{else.} \end{cases}$$

8. Cubic kernel is defined by

$$k(\mathbf{x}, \mathbf{z}) = a^{\frac{3}{2}} \|\mathbf{x} - \mathbf{z}\|^3, \tag{2.11}$$

where a is the scale parameter.

9. Neighborhood Indicator (Bubble) kernel is defined by

$$k(\mathbf{x}, \mathbf{z}) = a\|\mathbf{x} - \mathbf{z}\|^2 < 1, \tag{2.12}$$

where a is the scale parameter. This kernel gives a logical value which indicates if two observations are neighbors (1) or not (0).

10. B-spline is defined by

$$k(\mathbf{x}, \mathbf{z}) = B_{2N+1}(\mathbf{x} - \mathbf{z}). \tag{2.13}$$

11. ANOVA Spline is defined by

$$k(\mathbf{x}, \mathbf{z}) = \prod_{i=1}^{p} 1 + x_i z_i + x_i z_i \min(x_i, z_i) - \frac{x_i + z_i}{2} \left[\min(x_i, z_i)^2 + \frac{1}{3}\min(x_i, z_i)\right]^3, \tag{2.14}$$

where p is the dimension of the observations.

12. ANOVA B-Spine is defined by

$$k(\mathbf{x}, \mathbf{z}) = \prod_{i=1}^{p} (1 + a_i), \tag{2.15}$$

where

$$a_i = \sum_{j=0}^{2(N+1)} (-1)^j \begin{pmatrix} 2(N+1) \\ j \end{pmatrix} [\max(0, x_i - z_i + N + 1 - j)]^{(2N+1)}. \tag{2.16}$$

The plots of these kernel functions are shown in Figure 2.1.

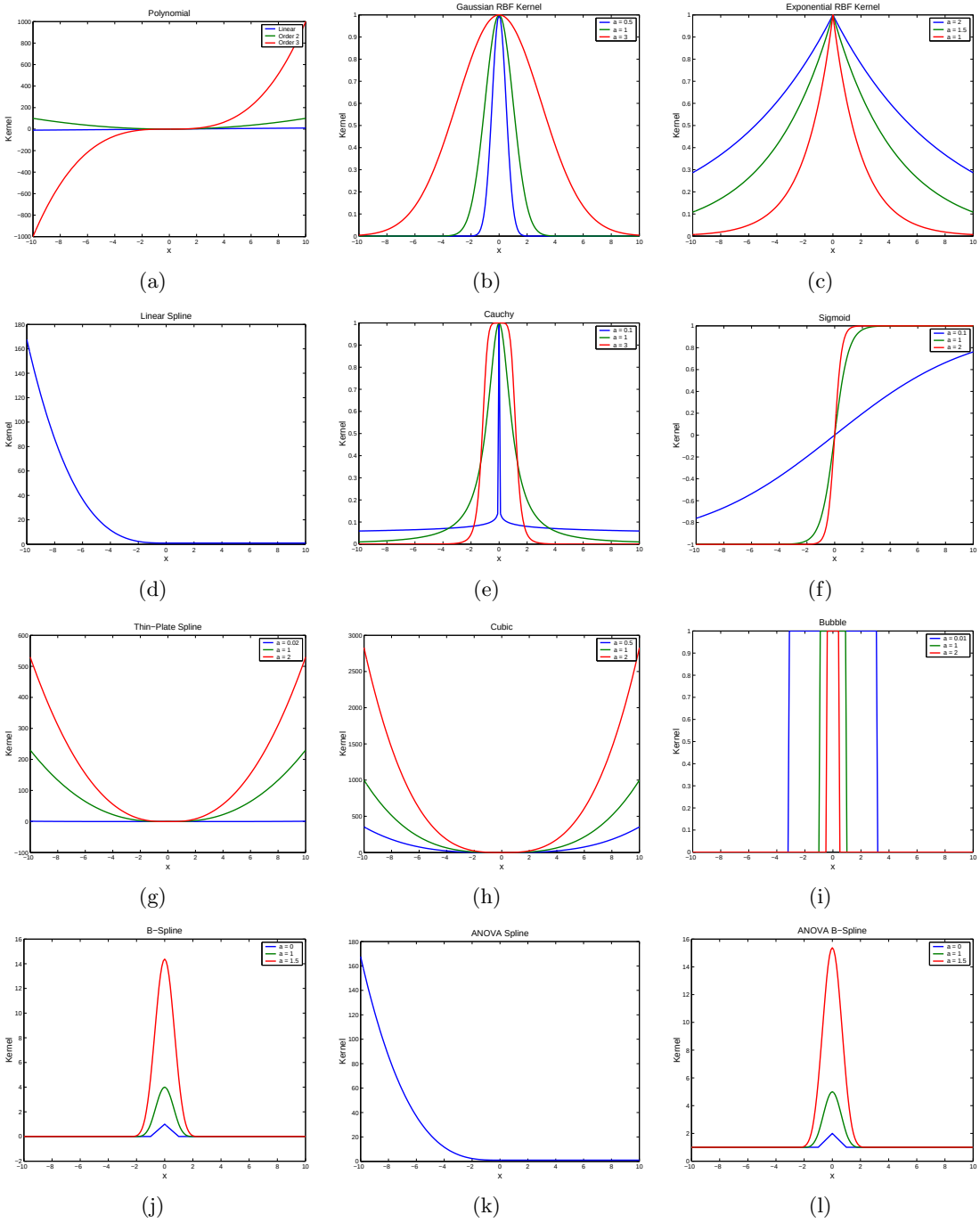


Figure 2.1: Plots of Kernel Functions (a) Polynomial (b) Guassian RBF (c) Exponential RBF (d) Linear Spline (e) Cauchy (f) Sigmoid (g) Thin-plate Spline (h) Cubic (i) Bubble (j) B-Spline (k) ANOVA Spline (l) ANOVA B-Spline

2.3 Some Implementation Issues in the Feature Space

2.3.1 Centering in the Feature Space

Before specifying the specific kernel functions, since the exact form of $\Phi(\mathbf{X})$ is not known, centering can not be performed directly on the data matrix of the feature space $\mathcal{F}$. However, only the kernel matrix is needed for the training purpose and the kernel matrix $\mathbf{K}_{(c)}$ based on the centered $\Phi(\mathbf{X})$ can be obtained as a function of $\mathbf{K}$. $\mathbf{K}_{(c)}$ in terms of $\Phi(\mathbf{X})$ is shown as:

$$\begin{aligned}
\mathbf{K}_{(c)} &= \left[\Phi(\mathbf{X}) - \overline{\Phi(\mathbf{X})}\right]\left[\Phi(\mathbf{X}) - \overline{\Phi(\mathbf{X})}\right]^T \\
&= \left[\Phi(\mathbf{X}) - \frac{1}{n}\mathbf{J}\Phi(\mathbf{X})\right]\left[\Phi(\mathbf{X}) - \frac{1}{n}\mathbf{J}\Phi(\mathbf{X})\right]^T \\
&= \Phi(\mathbf{X})\Phi(\mathbf{X})^T - \frac{1}{n}\Phi(\mathbf{X})\Phi(\mathbf{X})^T\mathbf{J} - \frac{1}{n}\mathbf{J}\Phi(\mathbf{X})\Phi(\mathbf{X})^T + \frac{1}{n^2}\mathbf{J}\Phi(\mathbf{X})\Phi(\mathbf{X})^T\mathbf{J} \\
&= \mathbf{K} - \frac{1}{n}\mathbf{K}\mathbf{J} - \frac{1}{n}\mathbf{J}\mathbf{K} + \frac{1}{n^2}\mathbf{J}\mathbf{K}\mathbf{J}, \qquad (2.17)
\end{aligned}$$

where $\mathbf{J}$ is the $n \times n$ square matrix of 1s. If the independent variables of the training data are centered, the new observations of independent variables $\Phi(\mathbf{X_{new}})$ needs to be centered using the mean vector of the training data $\Phi(\mathbf{X})$ before being used for the prediction. Similarly, $\mathbf{K_{new}}$ needs to be transformed to $\mathbf{K_{new(c)}}$, which is based on the centered training data and new data in the feature space as shown below:

$$\begin{aligned}
\mathbf{K_{new(c)}} &= \left[\Phi(\mathbf{X_{new}}) - \overline{\Phi(\mathbf{X})}\right]\left[\Phi(\mathbf{X}) - \overline{\Phi(\mathbf{X})}\right]^T \\
&= \left[\Phi(\mathbf{X_{new}}) - \frac{1}{n}\mathbf{J_{new}}\Phi(\mathbf{X})\right]\left[\Phi(\mathbf{X}) - \frac{1}{n}\mathbf{J}^T\Phi(\mathbf{X})\right]^T \\
&= \Phi(\mathbf{X_{new}})\Phi(\mathbf{X})^T - \frac{1}{n}\mathbf{J_{new}}\Phi(\mathbf{X})\Phi(\mathbf{X})^T \\
&\quad -\frac{1}{n}\Phi(\mathbf{X_{new}})\Phi(\mathbf{X})^T\mathbf{J}^T + \frac{1}{n^2}\mathbf{J_{new}}\Phi(\mathbf{X})\Phi(\mathbf{X})^T\mathbf{J}^T \\
&= \mathbf{K_{new}} - \frac{1}{n}\mathbf{J_{new}}\mathbf{K} - \frac{1}{n}\mathbf{K_{new}}\mathbf{J} + \frac{1}{n^2}\mathbf{J_{new}}\mathbf{K}\mathbf{J}, \qquad (2.18)
\end{aligned}$$

where $\mathbf{J_{new}}$ is a $m \times n$ matrix of 1s.

2.3.2 Adding a Constant Term to the Regression Problems

If the variables are not centered, the following two methods can be used:

1. Add 1 to each element of the original kernel matrix. This is equivalent as adding a column of 1s to $\Phi(\mathbf{X})$ in the feature space and defining the kernel matrix as:

$$k_{ij} =< [1 \quad \Phi(\mathbf{x})], [1 \quad \Phi(\mathbf{z})] > . \tag{2.19}$$

2. Add a column of 1s to the kernel matrix.

Chapter 3

Information Complexity Criteria

3.1 Introduction

The information complexity (ICOMP) measure is a statistical model evaluation criterion originally proposed by Bozdogan (1988). The development of ICOMP was inspired by Akaike's information criterion (AIC) (Akaike, 1973, 1974), which evaluates both the goodness of a model's fit to the sample data and the complexity of the model. In general AIC is defined by

$$\text{AIC} = -2\log L(\hat{\boldsymbol{\theta}}) + 2p, \tag{3.1}$$

where $L(\hat{\boldsymbol{\theta}})$ is the maximized likelihood function in which $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate of the parameter (or parameters) $\boldsymbol{\theta}$, and p is the number of independent parameters of the model. In AIC, the compromise takes place between $-2\log L(\hat{\boldsymbol{\theta}})$, the measure of the lack-of-fit, and $2p$, the complexity of the model. The optimal model is chosen as the minimizer of AIC. AIC is an unbiased estimator of minus twice the expected log likelihood (Akaike, 1987; Bozdogan, 2000). ICOMP uses the same lack-of-fit measurement as AIC does, but the complexity measure of a model is based on a generalization of the covariance complexity index originally introduced by van Emden (1971). It measures the degree of the interaction or the dependency between the components of a model (Bozdogan, 2004a). The general

form of ICOMP for a specific model is defined by

$$\text{ICOMP} = -2\text{log}L(\hat{\boldsymbol{\theta}}) + 2\text{C}[\widehat{Cov(\hat{\boldsymbol{\theta}})}], \tag{3.2}$$

where C[·] represents a real-valued measure of complexity of the model, and $\widehat{Cov(\hat{\theta})}$ represents the estimated covariance matrix of the parameter vector of the model.

To derive the complexity, we consider a continuous p-variate distribution with the joint density function $f(\mathbf{X}) = f(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_p})$ and marginal density functions $f_{\mathbf{i}}(\mathbf{x_i}), \mathbf{i} = \mathbf{1}, \mathbf{2}, ..., \mathbf{p}$. The informational measure of dependence between p random variables is defined by (Bozdogan, 2004a)

$$\begin{aligned} I(\mathbf{X}) &= I(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_p}) \\ &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}) \text{log} \frac{f(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p})}{f_{\mathbf{1}}(\mathbf{x_1}) f_{\mathbf{2}}(\mathbf{x_2}) \cdots f_{\mathbf{p}}(\mathbf{x_p})} d\mathbf{x_1} \cdots d\mathbf{x_p}, \end{aligned} \tag{3.3}$$

where $I(\mathbf{X}) \geq 0$ is known as the Kullback-Leibler (K-L) (1951) information against independence. The K-L information quantifies the meaning of "information" related to Fisher's concept of sufficient statistics. It's earlier roots can be traced back to the famous Boltzmann's concept of entropy (Boltzmann, 1877) in thermodynamics. $I(\mathbf{X}) = 0$, when $f(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}) = f_{\mathbf{1}}(\mathbf{x_1}) f_{\mathbf{2}}(\mathbf{x_2}) \cdots f_{\mathbf{p}}(\mathbf{x_p})$, if and only if the p random variables are mutually independent. If $I(\mathbf{X}) > 0$, this implies at least one variable is correlated with one or multiple of the other variables. Equation (3.3) can be written in terms of Shannon's (1948) entropy given by

$$I(\mathbf{X}) \equiv I(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}) = \sum_{\text{i=1}}^{p} H(\mathbf{x_i}) - H(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}), \tag{3.4}$$

where $H(\mathbf{x_i})$ is the marginal entropy and $H(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p})$ is the global or join entropy. Watanabe (1985) calls (3.4) the strength of structure and a measure of interdependence. The desirable model should have relatively smaller complexity or information entropy such

that the variables or the components of the model are less dependent to each other thus have less redundant information.

Assuming the p variables follow a multivariate normal distribution, i.e. $\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with the corresponding density function

$$f(\mathbf{X}) = (2\pi)^{-\frac{p}{2}} |\boldsymbol{\Sigma}| \exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathbf{T}}\boldsymbol{\Sigma}^{-\mathbf{1}}(\mathbf{x}-\boldsymbol{\mu})\} \tag{3.5}$$

Van Emden (1971) first defined the information complexity of a covariance matrix $\boldsymbol{\Sigma}$ for the multivariate normal distribution as:

$$\begin{aligned} I(\mathbf{X}) &= \sum_{\mathrm{i}=1}^{p} H(\mathbf{x_i}) - H(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}) \\ &= \sum_{i=1}^{p} \left[\frac{1}{2}\log(2\pi) + \frac{1}{2}\log(\sigma_{ii}) + \frac{1}{2}\right] - \frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{p}{2}, \end{aligned} \tag{3.6}$$

which can be simplified to:

$$\mathrm{C}_0(\boldsymbol{\Sigma}) = \frac{1}{2}\sum_{i=1}^{p}\log(\sigma_{ii}) - \frac{1}{2}\log|\boldsymbol{\Sigma}|, \tag{3.7}$$

where $\sigma_{ii} \equiv \sigma_i^2$ is the i^{th} diagonal element of $\boldsymbol{\Sigma}$, that is, the variance of the i^{th} random variable. $\mathrm{C}_0(\boldsymbol{\Sigma}) = 0$ when all variables are linearly independent. $\mathrm{C}_0(\boldsymbol{\Sigma})$ is infinite ($|\boldsymbol{\Sigma}|$=0) if $\boldsymbol{\Sigma}$ is not of full rank, that is, at least one variable is the linear combination of the others. The two drawbacks of $\mathrm{C}_0(\boldsymbol{\Sigma})$ pointed out by van Emden (1971) are:

- $\mathrm{C}_0(\boldsymbol{\Sigma})$ depends on the marginal and the joint distributions of the random variables and
- $\mathrm{C}_0(\boldsymbol{\Sigma})$ is coordinate dependent because $\sum_{i=1}^{p}\log(\sigma_{ii})$ would change under orthonormal transformations.

Because of this, Bozdogan (1990) proposed a maximal covariance complexity measure of a multivariate normal distribution:

$$\mathrm{C}_1(\mathbf{\Sigma}) = \frac{p}{2}\log\left[\frac{tr(\mathbf{\Sigma})}{p}\right] - \frac{1}{2}\log|\mathbf{\Sigma}| = \frac{1}{2}\log\frac{\left(\frac{tr(\mathbf{\Sigma})}{p}\right)^p}{|\mathbf{\Sigma}|}. \tag{3.8}$$

Note that $\mathrm{C}_1(\mathbf{\Sigma})$ is an upper bound to $\mathrm{C}_0(\mathbf{\Sigma})$ and is independent of the coordinate system by considering the contribution of covariances as well as variances in $\mathbf{\Sigma}$. $C_1(\mathbf{\Sigma})$ combines the geometric mean of the average total variation and the generalized variance into one index. In general, large complexity indicates a high dependency among the variables and a low complexity value indicates less dependency among the variables. If we let $\lambda_1, \lambda_2, \ldots, \lambda_p$ be the eigenvalues of $\mathbf{\Sigma}$, then the complexity of $\mathbf{\Sigma}$ can be written as

$$\mathrm{C}_1(\mathbf{\Sigma}) = \frac{p}{2}\log\left(\frac{\bar{\lambda}_a}{\bar{\lambda}_g}\right), \tag{3.9}$$

where $\bar{\lambda}_a = \sum_{i=1}^{p}\lambda_i/p$ is the arithmetic mean of the eigenvalues of $\mathbf{\Sigma}$ and $\bar{\lambda}_g = (\prod_{i=1}^{p}\lambda_i)^{\frac{1}{p}}$ is the geometric mean of the eigenvalues.

Under some regularity conditions, the maximum likelihood estimator (MLE) is asymptotically (for large samples) normally distributed. The mean is equal to $\boldsymbol{\theta}$ and the covariance matrix is equal to the inverse of Fisher information (FIM). Note that the $(i,j)^{th}$ element of the Fisher information matrix is defined by

$$\mathcal{F}(\boldsymbol{\theta})_{ij} = -E\left[\frac{\partial^2 L}{\partial\theta_i\theta_j}\right]. \tag{3.10}$$

Therefore, $\mathcal{F}^{-1}$ is another way to calculate the covariance matrix of the estimated parameters used in the information measure.

3.2 ICOMP Criteria for Multiple Regression Models

Bozdogan (1998) proposed a general form of ICOMP based on the inverse Fisher information (IFIM) for a statistical model given by

$$\mathrm{ICOMP(IFIM)} = -2\log L(\hat{\boldsymbol{\theta}}) + 2\mathrm{C}_1(\widehat{\mathcal{F}}^{-1}(\hat{\boldsymbol{\theta}})), \tag{3.11}$$

where $\hat{\boldsymbol{\theta}}$ represents the MLE of the model and $\widehat{\mathcal{F}}^{-1}(\hat{\boldsymbol{\theta}})$ represents the estimated IFIM of the parameter manifold of the model. We consider a multiple regression model given by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{3.12}$$

where

$$\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^{\mathbf{2}}\mathbf{I}).$$

Under the normality assumption, the probability density of the reponse is defined as:

$$f(y_i|\mathbf{x}_i, \boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left[-\frac{(y_i - \mathbf{x}_i\boldsymbol{\beta})^2}{2\sigma^2}\right]. \tag{3.13}$$

The log-likelihood function of a random sample of size n is:

$$\begin{aligned} l(\boldsymbol{\beta}, \sigma^2) &= \log L(\boldsymbol{\beta}, \sigma^2) \\ &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\sigma^2 - \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{2\sigma^2} \end{aligned} \tag{3.14}$$

The maximum likelihood estimates of $\boldsymbol{\beta}$ and σ^2 are given by:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{y} \tag{3.15}$$

$$\hat{\sigma}^2 = \frac{\text{Sum of Squared Residuals}}{n} = \frac{(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})}{n} \tag{3.16}$$

The covariances of the estimated regression coefficients are

$$\begin{aligned} \text{Cov}(\hat{\boldsymbol{\beta}}) &= \text{Cov}((\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{y}) \\ &= (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\text{Cov}(\mathbf{y})\mathbf{X}(\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}} \\ &= \sigma^2(\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{X}(\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-\mathbf{1}} \\ &= \sigma^2(\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}, \end{aligned} \tag{3.17}$$

where σ^2 can be estimated by (3.16).

Using the estimated covariance of the regression coefficients, ICOMP for a multiple regression model can be defined as:

$$\begin{aligned}
\text{ICOMP}(\widehat{\text{Cov}}(\hat{\boldsymbol{\beta}}))_{Reg} &= -2\log L(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}}) + 2C_1(\widehat{\text{Cov}}(\hat{\boldsymbol{\beta}})) \\
&= n\log(2\pi) + n\log(\hat{\sigma}^2) + n \\
&\quad +2\left[\frac{p}{2}\log\left(\frac{tr(\widehat{\text{Cov}}(\hat{\boldsymbol{\beta}}))}{p}\right) - \frac{1}{2}\log\left|\widehat{\text{Cov}}(\hat{\boldsymbol{\beta}})\right|\right] \\
&= n\log(2\pi) + n\log(\hat{\sigma}^2) + n + p\cdot\log\left(\frac{\overline{\lambda}_a}{\overline{\lambda}_g}\right) \qquad (3.18)
\end{aligned}$$

Under the large sample assumption, we may use IFIM to estimate the covariance of the estimated model parameters given by

$$\widehat{\text{Cov}}(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2) = \widehat{F}^{-1} = \begin{bmatrix} \widehat{\text{Cov}}(\hat{\boldsymbol{\beta}}) & \widehat{\text{Cov}}(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2) \\ \widehat{\text{Cov}}(\hat{\sigma}^2, \hat{\boldsymbol{\beta}}) & \widehat{\text{Var}}(\hat{\sigma}^2) \end{bmatrix}. \qquad (3.19)$$

To derive the above equation, the first and the second partial derivatives of the log-likelihood function defined in (3.14) are as follows.

$$\frac{\partial l(\boldsymbol{\beta}, \sigma^2)}{\partial \boldsymbol{\beta}} = \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathbf{T}}\mathbf{X}}{\sigma^2} \qquad (3.20)$$

$$\frac{\partial l(\boldsymbol{\beta}, \sigma^2)}{\partial (\sigma^2)} = -\frac{n}{2\sigma^2} + \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{2\sigma^4} \qquad (3.21)$$

$$\frac{\partial^2 l(\boldsymbol{\beta}, \sigma^2)}{\partial \boldsymbol{\beta}^2} = -\frac{\mathbf{X}^{\mathbf{T}}\mathbf{X}}{\sigma^2} \qquad (3.22)$$

$$\frac{\partial^2 l(\boldsymbol{\beta}, \sigma^2)}{\partial (\sigma^2)^2} = \frac{n}{2\sigma^4} - \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{\sigma^6} \qquad (3.23)$$

$$\frac{\partial^2 l(\boldsymbol{\beta}, \sigma^2)}{\partial (\sigma^2)\partial \boldsymbol{\beta}} = \left[\frac{\partial^2 l(\boldsymbol{\beta}, \sigma^2)}{\partial \boldsymbol{\beta}\partial (\sigma^2)}\right]^T = -\frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathbf{T}}\mathbf{X}}{\sigma^4} \qquad (3.24)$$

The corresponding negative expected values of the second derivatives are:

$$
\begin{aligned}
-\mathrm{E}\left[\frac{\partial^2 l(\boldsymbol{\beta},\sigma^2)}{\partial \boldsymbol{\beta}^2}\right] &= \frac{\mathbf{X^T X}}{\sigma^2} \\
-\mathrm{E}\left[\frac{\partial^2 l(\boldsymbol{\beta},\sigma^2)}{\partial (\sigma^2)^2}\right] &= -\frac{n}{2\sigma^4} + \frac{\mathbf{E}\left[(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^{\mathbf{T}}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right]}{\sigma^6} \\
&= -\frac{n}{2\sigma^4} + \frac{n\sigma^2}{\sigma^6} \\
&= \frac{n}{2\sigma^4} \\
-\mathrm{E}\left[\frac{\partial^2 l(\boldsymbol{\beta},\sigma^2)}{\partial (\sigma^2)\partial \boldsymbol{\beta}}\right] &= -\mathrm{E}\left[\frac{\partial^2 l(\boldsymbol{\beta},\sigma^2)}{\partial \boldsymbol{\beta}\partial (\sigma^2)}\right]^T = -\frac{\mathrm{E}\left[(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^{\mathbf{T}}\right]\mathbf{X}}{\sigma^4} = \mathbf{0}_{1\times p}
\end{aligned}
$$

Therefore, the Fisher information matrix for the regression is given by

$$
\mathcal{F} = \begin{bmatrix} \frac{\mathbf{X^T X}}{\sigma^2} & \mathbf{0}_{p\times 1} \\ \mathbf{0}_{1\times p} & \frac{n}{2\sigma^4} \end{bmatrix}. \tag{3.25}
$$

The estimated inverse Fisher information (IFIM) in (3.19) is defined as:

$$
\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}},\hat{\sigma}^2) = \widehat{\mathcal{F}}^{-1} = \begin{bmatrix} \hat{\sigma}^2(\mathbf{X^T X})^{-1} & \mathbf{0}_{p\times 1} \\ \mathbf{0}_{1\times p} & \frac{2\hat{\sigma}^4}{n} \end{bmatrix}, \tag{3.26}
$$

where $\hat{\sigma}^2$ is given in (3.16).

Using IFIM, the ICOMP for a multiple regression model has the form of:

$$
\begin{aligned}
ICOMP(IFIM)_{Reg} &= -2\log L(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}}) + 2C_1\left(\widehat{\boldsymbol{\Sigma}}(\hat{\boldsymbol{\theta}})\right) \\
&= n\log(2\pi) + n\log(\hat{\sigma}^2) + n + 2C_1\left(\hat{\mathcal{F}}^{-1}(\hat{\boldsymbol{\theta}})\right),
\end{aligned} \tag{3.27}
$$

where

$$
\begin{aligned}
C_1\left(\hat{\mathcal{F}}^{-1}(\hat{\boldsymbol{\theta}})\right) &= \frac{1}{2}(p+1)\log\left[\frac{tr\left(\hat{\sigma}^2(\mathbf{X^T X})^{-1}\right) + \frac{2\hat{\sigma}^4}{n}}{p+1}\right] \\
&\quad -\frac{1}{2}\log\left|\hat{\sigma}^2(\mathbf{X^T X})^{-1}\right| - \frac{1}{2}\log\left(\frac{2\hat{\sigma}^4}{n}\right),
\end{aligned} \tag{3.28}
$$

and where $p+1$ is the total number of the estimated independent model parameters.

Van Emden (1971) suggested a second measure of complexity of a covariance matrix based on the Frobenious norm given by

$$C_F(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})) = \frac{1}{s}\|\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})\|^2 - \left(\frac{tr(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))}{s}\right)^2, \tag{3.29}$$

where $\|\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})\|^2 = tr(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})^T\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))$, the square of the Frobenious norm of $\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})$, and s is the number of estimated independent model parameters, that is, the rank of $\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})$. $C_F(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))$ is a non-negative index with $C_F(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})) = 0$ when all the eigenvalues are the same. In terms of the eigenvalues, $C_F(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))$ can be written as

$$C_F(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})) = \frac{1}{s}\sum_{i=1}^{s}(\lambda_i - \overline{\lambda}_a)^2. \tag{3.30}$$

Bozdogan (Bozdogan, 2003; Bao and Bozdogan, 2004) related $C_1(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))$ to $C_F(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))$ and provided a new complexity measure called $C_{1F}(\cdot)$ given by

$$\begin{aligned} C_{1F}(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})) &= \frac{s}{4}\frac{C_F(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))}{\left(\frac{tr(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))}{s}\right)^2} \\ &= \frac{s}{4}\frac{\frac{1}{s}tr(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})^T\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}})) - \left(\frac{tr(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))}{s}\right)^2}{\left(\frac{tr(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))}{s}\right)^2} \qquad (3.31) \\ &= \frac{1}{4\overline{\lambda}_a^2}\sum_{i=1}^{s}(\lambda_i - \overline{\lambda}_a)^2. \qquad (3.32) \end{aligned}$$

$C_{1F}(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))$ is scale-invariant and is a second order equivalent measure of complexity to the original $C_1(\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\theta}}))$ measure. Because it can be expressed by eigenvalues of the covariance matrix, it is ideal for the statistical models with orthogonal (uncorrelated) variables which are eigen-problems, including partial least squares regression, principal component analysis, principal component regression, Fisher's discriminant analysis and canonical correlation analysis.

3.3 ICOMP Criteria for Multivariate Regression Models

The above results on ICOMP in the usual multiple regression case can be easily extended to the multivariate regression. Consider the multivariate regression model given by

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E} \tag{3.33}$$

where the $n \times q$ matrix $\mathbf{Y}$ represents q response variables, the $n \times p$ matrix $\mathbf{X}$ represents the p-dimensional independent variables with n sample observations, the $p \times q$ matrix $\mathbf{B}$ stands for the regression coefficient, and the $n \times q$ random errors $\mathbf{E}$ have the mean vector of zero and the constant covariance $\mathbf{\Sigma}$. Assuming that the random error $\mathbf{e_i}(1 \times q$ vector) follows a multivariate normal distribution with the mean vector of zero, the probability density function of $\mathbf{e_i}$ is given by

$$f(\mathbf{e_i}) = \frac{1}{2\pi^{\frac{q}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{e_i}\mathbf{\Sigma^{-1}}\mathbf{e_i^T}\right). \tag{3.34}$$

AIC for the multivariate regression model is defined by

$$\text{AIC(Multivar Reg)} = nq\log(2\pi) + n\log|\hat{\mathbf{\Sigma}}| + nq + 2\left[pq + \frac{q(q+1)}{2}\right]. \tag{3.35}$$

The corresponding ICOMP(IFIM) derived by Bozdogan (1998) has the form:

$$\text{ICOMP(IFIM)}_{\text{Multivar Reg}} = nq\log(2\pi) + n\log|\hat{\mathbf{\Sigma}}| + nq + 2C_1(\hat{\mathcal{F}}^{-1}(\hat{\boldsymbol{\theta}})), \tag{3.36}$$

where the maximum likelihood estimate of the error covariance matrix is given by

$$\hat{\mathbf{\Sigma}} = \frac{(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})^{\mathbf{T}}(\mathbf{Y} - \mathbf{X}\hat{\mathbf{B}})}{n}, \tag{3.37}$$

and the estimated IFIM, $\hat{\mathcal{F}}^{-1}(\hat{\boldsymbol{\theta}})$, is given by

$$\hat{\mathcal{F}}^{-1}(\hat{\boldsymbol{\theta}}) = \begin{bmatrix} \hat{\boldsymbol{\Sigma}} \otimes (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1} & \mathbf{0}_{pq \times \frac{1}{2}q(q+1)} \\ \mathbf{0}_{\frac{1}{2}q(q+1) \times pq} & \frac{2}{n}\mathbf{D}_{\mathbf{q}}^{+}(\hat{\boldsymbol{\Sigma}} \otimes \hat{\boldsymbol{\Sigma}})\mathbf{D}_{\mathbf{q}}^{+\mathbf{T}}. \end{bmatrix}, \tag{3.38}$$

In (3.36), $\mathbf{D}_{\mathbf{q}}^{+}$ is the Moore-Penrose of the $q^2 \times \frac{q(q+1)}{2}$ duplication matrix $\mathbf{D}_{\mathbf{q}}$ (Magnus and Neudecker, 1988), and $\otimes$ represents the Kronecker tensor product. Given the above definition of $\hat{\mathcal{F}}^{-1}(\hat{\boldsymbol{\theta}})$, the complexity then becomes

$$\begin{aligned} C_1(\hat{\mathcal{F}}^{-1}(\hat{\boldsymbol{\theta}})) &= \frac{q(q+p)}{2}\log\left[\frac{tr(\hat{\boldsymbol{\Sigma}})tr(\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1} + \frac{1}{2n}\left[tr(\hat{\boldsymbol{\Sigma}}^2) + tr^2(\hat{\boldsymbol{\Sigma}}) + 2\sum_{i=1}^{q}\hat{\sigma}_{ii}^2\right]}{q(q+p)}\right] \\ &\quad -\frac{1}{2}(p+q+1)\log|\hat{\boldsymbol{\Sigma}}| - \frac{q}{2}\log\left|(\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\right| - \frac{q}{2}\log(2). \end{aligned} \tag{3.39}$$

In PCA, PCR, PLS, FDA and CCA, the modeling components are orthogonal and $C1(\cdot)$ goes to negative infinity. We use $C_{1F}(\cdot)$ to compute the complexity as we do for the univariate models. In this case, ICOMP is given by

$$\text{ICOMP}(\widehat{\text{Cov}}(\hat{\boldsymbol{\theta}}))_{\text{Multivar reg}} = nq\log(2\pi) + n\log|\hat{\boldsymbol{\Sigma}}| + nq + 2C_{1F}(\widehat{\text{Cov}}(\hat{\boldsymbol{\theta}})), \tag{3.40}$$

where $C_{1F}(\widehat{\text{Cov}}(\hat{\boldsymbol{\theta}}))$ is given in (3.32).

We assume Gaussian noise through out this dissertation in kernel-based methods. It is also possible that the noise distribution may not be normal. Our future work will be focused on applying the misspecification form of ICOMP (Bozdogan, 2004a,b, 2005) to explore its ability to guard against the non-normal noise in modeling of kernelized data.

3.4 ICOMP Criteria for Kernel Methods

It has been shown that ICOMP criteria have been successfully applied in many linear regression applications including ridge regression, multiple and multivariate regression, partial least squares regression, mixture models, principal component analysis, and Fisher's discriminant analysis. The purpose of this research is to extend these successful applications

of ICOMP to the kernel-based nonlinear modeling techniques, since after the "kernel trick" the modeling techniques become simply linear procedures, and we can carry out the model selection enterprise on the kernelized data as we would have done on the original data.

Chapter 4

Genetic Algorithms

4.1 Introduction

Genetic Algorithms (GAs), are searching techniques inspired by the natural selection of evolutionary biology. It was originally developed by John Holland, his colleagues and students in the early 1970s. GAs can be applied to find the optimal solutions to optimization problems when the exhaustive searching of all the possible solutions is impossible or not efficient to implement. GAs starts with a random sample (the first generation) of all the solutions. A fitness function is used to evaluate each solution and the better fitted solutions are retained to generate the next generation. In GAs, each solution is expressed by a binary string called chromosome. This will allow inheritance, mutation and recombination (also called crossover) to prevent from the local minimum. The retained good solutions then will mate and generate the next generation of solutions. This procedure is repeated until a certain convergence criterion is reached or the optimal solutions are found.

Efficiency is the major advantage of using GAs. The purpose of this chapter is not a thorough coverage of Genetic Algorithms but a brief introduction of the GA procedure which will be used for searching optimal subset models in this research. The interested readers may find the comprehensive coverage of GAs given in (Goldberg, 1989) and (Michalewicz, 1992).

In the regression analysis, choosing the optimal subset models for the best generalization and interpretation is often of interest. The traditional variable selection approaches include Backward and Forward stepwise selection. The well known shortcomings of these pocedures are that the order which a variable "enters" or "leaves" the model will affect the variable selection results (Boyce et al., 1974; Wilkinson, 1989). Also, there is no theoretical justification of such procedures. In addition, the probability (threshold) for entering and leaving the model is an arbitrary choice. Stepwise searching can hardly find the global best subset model or even the best subset of a particular size. This has been criticized in (Hocking, 1976, 1983; Mose, 1986; Mantel, 1970).

If practically feasible, all-possible-subset selection would be the best approach to subset selection. However, the "curse of dimensionality" (Bellman, 1961) makes it practically impossible to evaluate all the subsets. For instance, for 30 independent variables, we will have $2^{30} - 1 = 1,073,741,823$ subsets models to evaluate. In this research, we bring in the capability of GA to reduce the computational burden tremendously but still find the optimal subset models.

4.2 The Procedure of GAs

4.2.1 Representation

GAs perform selection, partition, combination and modification of the chromosomes, expressed as binary strings. Therefore, the first step is to represent the domain of a numerical optimization problem using the corresponding chromosomes. For instance, suppose x is the independent variable and the optimization is performed in the x domain of $[-1, 1]$. If the precision requires 100000 equal width intervals within the range, which means 100001 values, this will requires 17 bits since

$$65536 = 2^{16} < 100001 < 2^{17} = 131072$$

If "0000000000000001" represents "-1", then "11000011010100001" is the expression of "1" the 100001^{st} value. If there are multiple independent variables, one may connect the binary strings of the individual independent variables to a single long binary string.

In the subset selection problem of a regression analysis, representation is even simpler. Suppose there are 20 independent variables in the saturated model. We use a binary string of 20 bits to code the subset model. If a variable is included, the corresponding bit is "1", otherwise, "0". For instance,

$$10110000000000000000$$

stands for the subset model $[x_1, x_3, x_4]$. We lay out the bits from left to right to be consistent with the way that we write the regression formula. This is the inverse order of the tradition expression of a binary number. The following discussion is focused on the subset model selection using GA.

The subset models are evaluated by a fitness function. In this research, ICOMP is the subset model selection criterion. Higher values are better for the fitness but the lower values are better for ICOMP. The fitness function is defined as follows:

$$\Delta\text{ICOMP}_i = \max(\text{ICOMP}) - \text{ICOMP}_i \tag{4.1}$$

$$\text{Fitness}_i = \Delta\text{ICOMP}_i / \overline{\Delta\text{ICOMP}_i} \tag{4.2}$$

Our GA procedure starts with a random sample of subsets. The subsets with relative high fitness will be selected to reproduce the springs to the next generation. We applied a natural selection mechanism. We sort the fitness of the models in ascending order such that the model with the lowest fitness is ranked as 1 while the highest fitness is ranked as m. We then create a "weighted roulette wheel" with m bins, one for each subset, where the bin width for the subset with rank i is

$$\frac{i}{m(m+1)/2} \tag{4.3}$$

We then randomly shoot the bins by drawing a uniform random numbers from [0,1]. The chosen model is included in a mating pool for the reproduction. The fitter model has wider bin thus better chance to be selected.

4.2.2 Reproduction

After coding all the subsets to the binary chromosomes, we start with taking a random sample from all the subsets as the first generation. The population size n_{pop} (also the sample size for the first generation) of each generation is an arbitrary choice as long as it converges. We used 20 to 50 in our numerical experiments. The subsets in the population are evaluated using the fitness function (4.2).

Only the subsets with relative high fitness will be selected to the mating pool to reproduce the next generation. In our research, "relative high" is defined as follows. Let

$$\zeta_i = \frac{\text{Fitness}_i}{Mean(\text{Fitness})} \tag{4.4}$$

We select the subsets whose fitness ratio $\zeta_i > 0.5$ to the mating pool. The subsets in the mating pool are randomly paired to produce a pair of springs. Mating is performed as a crossover process. There are different ways to crossover the parents genes to the next generation. We utilized the following three types.

Single point crossover - One breakup point is randomly selected. Each of the two children's chromosomes is consisted of the first part of one parent and the second part of another parent. For instance, if the 4^{th} bit is chosen:

$ParentA$	$ParentB$
$1101 \cdot 01$	$0011 \cdot 00$
$Child_1$	$Child_2$
110100	001101

Two point crossover - Two breakup points are randomly decided for both parents. A child's chromosome is consisted of the first part and the last part of one parent's chromosome and the middle part of another parent's chromosome. For instance, if the breakup points are after the second and the fourth bits:

$ParentA$	$ParentB$
$11 \cdot 01 \cdot 01$	$00 \cdot 11 \cdot 00$
$Child_1$	$Child_2$
111101	000100

Uniform crossover - Bits are randomly copied from the first or the second parent. For instance:

$ParentA$	$ParentB$
11001011	11011101
$Child_1$	
11011111	

By doing crossover, we keep the variables (genes) in the "Good" subsets to the future generations. A probability of the crossover is defined. Generally, this probability is relative large, for instance, 0.7. If the crossover does not happen, the parents will become the offsprings for the next generation without any change. One may also force to keep the best subset to the next generation without any crossover.

4.2.3 Mutation

Mutation is to randomly change a bit of the chromosome from 0 to 1 or from 1 to 0. A low probability $P_{mutation}$ is assigned to the happening of mutation. Mutation allows the searching jumping out of the current searching area to avoid sticking with the local minimum.

Table 4.1: GA Parameters

n_{gen}	Number of Generations
n_{pop}	Population size of each generation
P_{cross}	Probability of crossover
$P_{mutation}$	Probability of notation
Elitism	Keep the best subset or not
Type of crossover	Single/Two/Uniform

4.2.4 Other Configuration

The above steps are repeated to reproduce future generation until a certain stop condition is satisfied. The stop condition could be the maximum number of iterations, a certain fitness value is reached, all the chromosomes are the same. The some searching parameter must be configured as shown in Table 4.1 before starting the iteration.

4.3 A Numerical Demonstration

In this section, we use a simple two-dimensional numerical example to demonstrate searching optimal solutions using GA. The target function is a mixture of 3 bivariate Gaussian probability density (Figure 4.1):

$$z = \frac{1}{3}\left[f(x, y|\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}) + f(x, y|\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2}) + f(x, y|\boldsymbol{\mu_3}, \boldsymbol{\Sigma_3})\right] + 5 \tag{4.5}$$

where

$$\boldsymbol{\mu_1} = \begin{bmatrix} 0 & 0 \end{bmatrix} \quad \boldsymbol{\mu_2} = \begin{bmatrix} 2 & -1.5 \end{bmatrix} \quad \boldsymbol{\mu_3} = \begin{bmatrix} -2 & -3 \end{bmatrix}$$

$$\boldsymbol{\Sigma_1} = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \quad \boldsymbol{\Sigma_2} = \begin{bmatrix} 5 & 1 \\ 1 & 2 \end{bmatrix} \quad \boldsymbol{\Sigma_3} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$$

Argument x and y are both defined in the range [-4, 4] with the interval of 0.1. There are 81 values for each argument thus 6561 different combinations of x and y. The target function

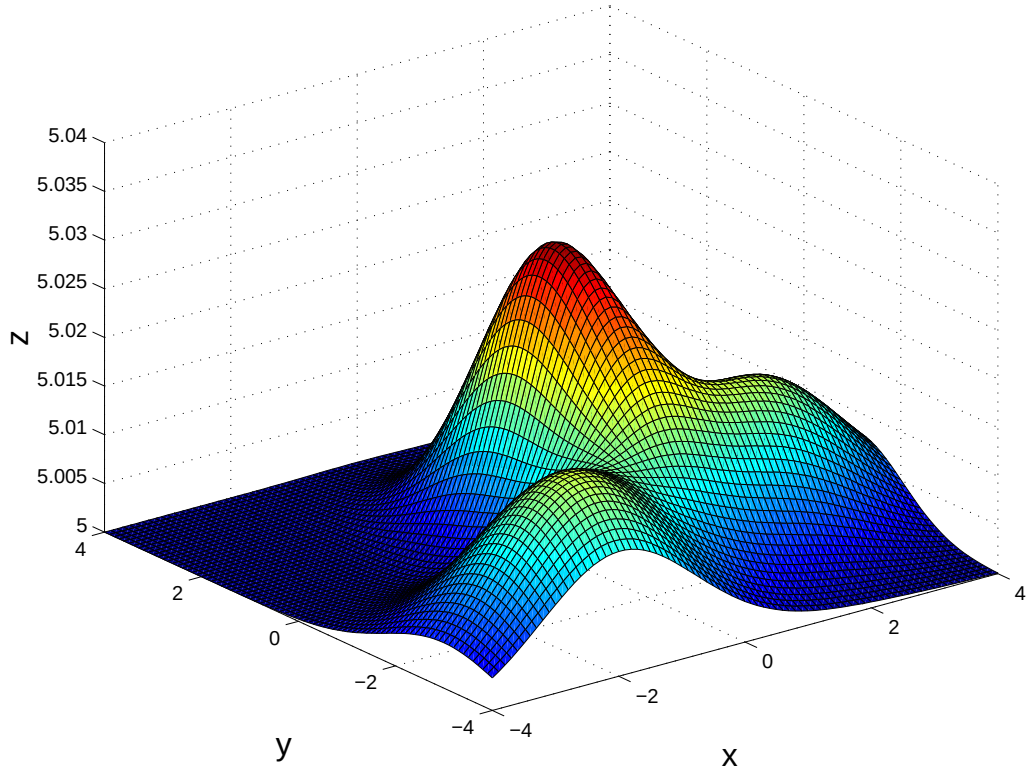


Figure 4.1: Surface Plot of the Target Function

z is maximized at 5.03216 when $x = 0$ and $y = -0.2$. There are also two local maximums (Figure 4.1).

Suppose we want to find the global maximum of z using GA. The function defined by (4.5) is the fitness. The GA parameters used in this example is listed in Table 4.2. GA starts from a random sample of 50 points as the initial population (Figure 4.2(a)). The fittest point (z = 5.0178) in this population is around a local maximum. After 5 generations, the fittest point (z = 5.0282) jumps to the ridge with the global maximum (Figure 4.2(b)). The population are concentrated in a small area around the global maximum after some generations (Figure 4.2(c)). At the 175^{th} generation, all the population have the same chromosome (Figure 4.2(d)), that is the same point. This point (x=0.1, y = -0.2) gives a z value of 5.0320, which is very close to the true maximum.

Table 4.2: GA Example Parameters

n_{gen}	200
n_{pop}	50
P_{cross}	0.9
$P_{mutation}$	0.01
Elitism	YES
Type of crossover	Single

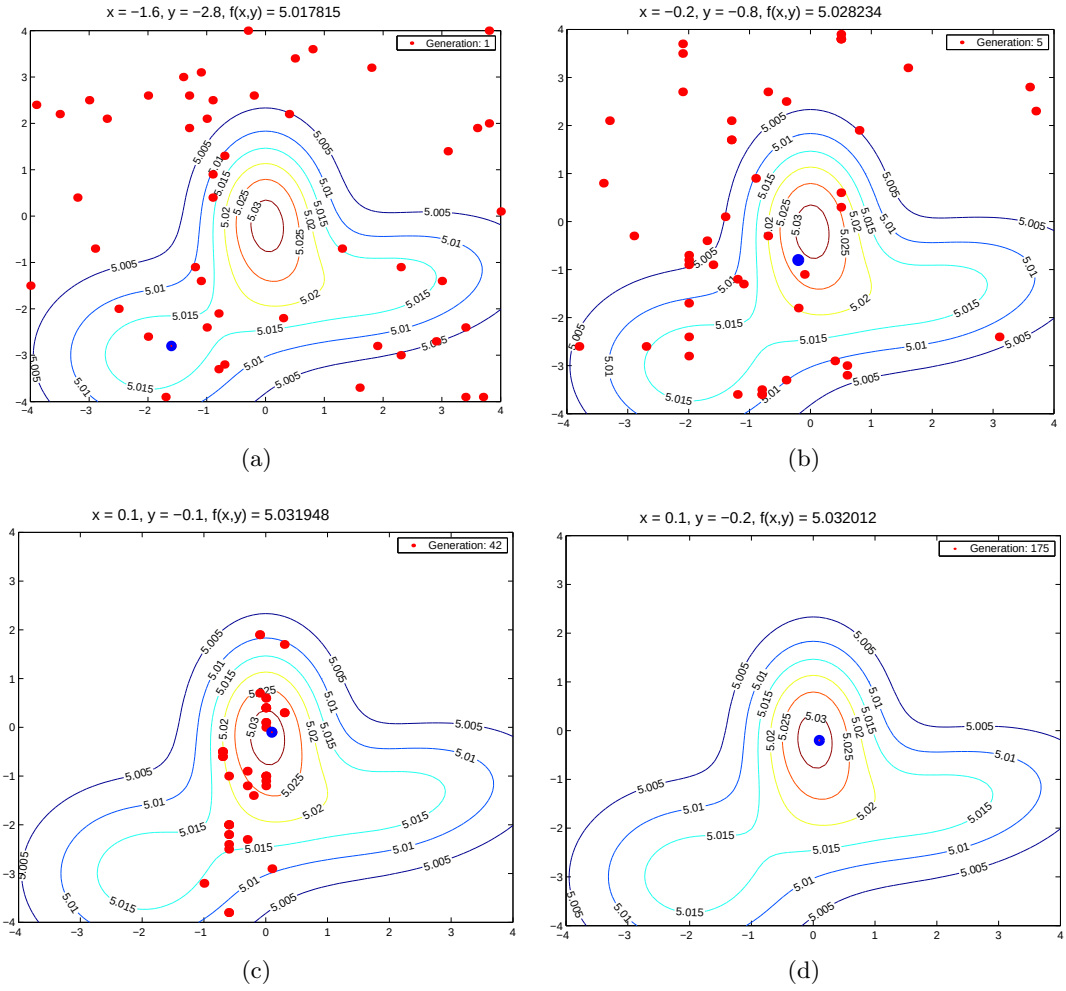


Figure 4.2: GA Searching Example. (a) Initial Generation (b) 5^{th} Generation (c) 42^{nd} Generation (d) 175^{th} Generation

Chapter 5

Kernel Ridge Regression

5.1 Introduction of Ridge Regression

Ridge regression, also known as Tikhanov regularization, is one of the most commonly used regularization methods for ill-posed problems. It was originally proposed by Hoerl and Kennard (1970b; 1970a). Consider the multiple regression model defined by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{5.1}$$

where the $n \times 1$ vector $\mathbf{y}$ represents a dependent variable with n observations, the $n \times p$ matrix $\mathbf{X}$ represents the p independent variables, $\boldsymbol{\beta}$ is a vector of p regression coefficients and $\boldsymbol{\epsilon}$ is the independent and identically distributed (i.i.d.) random error with mean of zero and constant variance σ^2. The ridge estimator of the regression coefficients is the minimizer of

$$f(\boldsymbol{\beta_r}) = \lambda \boldsymbol{\beta_r}^{\mathbf{T}} \boldsymbol{\beta_r} + (\mathbf{y} - \mathbf{X}\boldsymbol{\beta_r})^{\mathbf{T}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta_r}) \tag{5.2}$$

The first derivative of $f(\boldsymbol{\beta_r})$ with respect to $\boldsymbol{\beta_r}$ is

$$\frac{\mathbf{df}(\boldsymbol{\beta_r})}{\mathbf{d}\boldsymbol{\beta_r}} = \mathbf{2}\lambda \boldsymbol{\beta_r}^{\mathbf{T}} - \mathbf{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta_r})^{\mathbf{T}} \mathbf{X}. \tag{5.3}$$

The estimated ridge coefficients $\hat{\boldsymbol{\beta}}_{\boldsymbol{r}}$ is the value of $\boldsymbol{\beta}_{\boldsymbol{r}}$ that makes the above derivative equal to zero, i.e. the solution to

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\hat{\boldsymbol{\beta}}_{\boldsymbol{r}} = \mathbf{X}^T\mathbf{y}, \tag{5.4}$$

i.e.

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{r}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{5.5}$$

where $\lambda \geq 0$ is a predetermined constant which controlls the bias. When $\lambda = 0$, the ridge estimator is the least-squares estimator. The covariance of the ridge estimator is defined by

$$\begin{aligned} \mathrm{Cov}(\hat{\boldsymbol{\beta}}_{\boldsymbol{r}}) &= \mathrm{Cov}[(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathrm{VAR}(\mathbf{y})\mathbf{X}(\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I})^{-\mathbf{1}} \\ &= \sigma^2(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I})^{-\mathbf{1}}. \end{aligned} \tag{5.6}$$

The bias in $\hat{\boldsymbol{\beta}}_r$ increases with λ. However, the covariances decreases as λ increases.

There is an alternative expression for the ridge estimator which serves an important role in kernel ridge regression described in the next section. Equation (5.4) can be rewritten as

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{r}} = \mathbf{X}^T\frac{1}{\lambda}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\boldsymbol{r}}) = \mathbf{X}^T\mathbf{W}, \tag{5.7}$$

where $\mathbf{W} = \frac{1}{\lambda}(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\boldsymbol{r}})$. $\mathbf{W}$ is solved using the following steps:

$$\begin{aligned} & \mathbf{W}\lambda = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\boldsymbol{r}} = \mathbf{y} - \mathbf{X}\mathbf{X}^{\mathbf{T}}\mathbf{W} \\ \Rightarrow\ & (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})\mathbf{W} = \mathbf{y} \\ \Rightarrow\ & \mathbf{W} = (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{y} \end{aligned} \tag{5.8}$$

Substituting the result in Equation (5.8) back to (5.7), the alternative expression of the ridge estimator is defined as:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{r}} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{y}. \tag{5.9}$$

The above expression is called the dual representation (Shawe-Taylor and Cristianini, 2004).

5.2 Kernel Ridge Regression

In kernel ridge regression (KRR), the original p-dimensional data matrix $\mathbf{X}$ is nonlinearly transformed to $\Phi(\mathbf{X})$ in the feature space. The multiple linear regression model using $\Phi(\mathbf{X})$ as the independent variables can be defined by

$$\mathbf{y} = \Phi(\mathbf{X})\boldsymbol{\beta_r} + \boldsymbol{\varepsilon}. \tag{5.10}$$

The ridge estimator of the regression coefficients is the solution to

$$\left[\Phi(\mathbf{X})^T\Phi(\mathbf{X}) + \lambda\mathbf{I}\right]\hat{\boldsymbol{\beta}}_{\boldsymbol{r}} = \Phi(\mathbf{X})^T\mathbf{y}.$$

Using the result in Equation (5.5), the kernel ridge estimator is expressed as

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{r}} = \left[\Phi(\mathbf{X})^T\Phi(\mathbf{X}) + \lambda\mathbf{I}\right]^{-1}\Phi(\mathbf{X})^T\mathbf{y} \tag{5.11}$$

Since the explicit form of $\Phi(\mathbf{X})$ is unknown in kernel methods, the above expression can not lead to an explicit estimation of $\mathbf{y}$, the response. However, using the dual representation from Equation (5.9), the kernel ridge estimator (Shawe-Taylor and Cristianini, 2004) is defined by

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{r}} = \Phi(\mathbf{X})^T\left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y}, \tag{5.12}$$

where $\mathbf{K} = \Phi(\mathbf{X})\Phi(\mathbf{X})^{\mathbf{T}}$ is the kernel matrix. Although the explicit expression of $\hat{\boldsymbol{\beta}}_{\boldsymbol{r}}$ is not available, the prediction of the observation can be obtained through kernel functions. The point estimate of the training observations can be obtained by

$$\hat{\mathbf{y}} = \Phi(\mathbf{X})\hat{\boldsymbol{\beta}}_{\boldsymbol{r}}\Phi(\mathbf{X})\Phi(\mathbf{X})^T\left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y} \quad = \mathbf{K}\left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y}.$$

Similarly, the point estimate of the testing observations or new observations can be obtained by

$$\begin{aligned}\hat{\mathbf{y}}_{\text{New}} &= \Phi(\mathbf{X}_{\text{New}})\hat{\boldsymbol{\beta}}_{\boldsymbol{r}} \\ &= \Phi(\mathbf{X}_{\text{New}})\Phi(\mathbf{X})^T \left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y} \\ &= \mathbf{K}_{\text{New}} \left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y}. \end{aligned} \tag{5.13}$$

The covariance of $\hat{\boldsymbol{\beta}}_{\boldsymbol{r}}$ defined in (5.12) is given by

$$\begin{aligned}\text{Cov}(\hat{\boldsymbol{\beta}}_{\boldsymbol{r}}) &= \text{Cov}(\Phi(\mathbf{X})^T \left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y}) \\ &= \Phi(\mathbf{X})^T \left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1} \text{VAR}(\mathbf{y}) \left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1} \Phi(\mathbf{X}) \\ &= \sigma^2\Phi(\mathbf{X})^T \left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1} \left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1} \Phi(\mathbf{X}). \end{aligned} \tag{5.14}$$

Since $\Phi(\mathbf{X})$ has no explicit form, the exact form of $\text{Cov}(\hat{\boldsymbol{\beta}}_{\boldsymbol{r}})$ is unknown. This problem brings troubles to the model selection techniques proposed later as $\text{Cov}(\hat{\boldsymbol{\beta}}_{\boldsymbol{r}})$ is the important part in the complexity measure. To solve this problem we look at the kernel ridge regression model in a different way in this dissertation. We treat $\mathbf{K}$ and $\mathbf{K_{New}}$ as the training data and new observations of the independent variables respectively. The kernel ridge regression model can be redefined by

$$\mathbf{y} = \mathbf{K}\boldsymbol{\beta}_r^* + \varepsilon. \tag{5.15}$$

The ridge estimator defined in (5.12) is modified to

$$\hat{\boldsymbol{\beta}}_r^* = \left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y}. \tag{5.16}$$

The predicted response given the training data $\mathbf{X}$ is

$$\hat{\mathbf{y}} = \mathbf{K}\hat{\boldsymbol{\beta}}_r^* = \mathbf{K}\left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y}. \tag{5.17}$$

The predicted response given the new observations $\mathbf{X_{New}}$ is

$$\hat{\mathbf{y}} = \mathbf{K_{New}}\hat{\boldsymbol{\beta}}_r^* = \mathbf{K_{New}}\left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y}. \tag{5.18}$$

We see that this alternative expression does not change the model assumptions or the estimated kernel ridge regression model. The benefit of using this expression is that the covariance of the ridge estimator $\hat{\boldsymbol{\beta}}_r^*$ has the explicit form defined by

$$\begin{aligned} \mathrm{Cov}(\hat{\boldsymbol{\beta}}_{\boldsymbol{r}}^{*}) &= \mathrm{Cov}(\left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathbf{y}) \\ &= \left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1}\mathrm{VAR}(\mathbf{y})\left[\mathbf{K} + \lambda\mathbf{I}\right]^{-1} \\ &= \sigma^2\left[\mathbf{K} + \lambda\mathbf{I}\right]^{-2}. \end{aligned} \tag{5.19}$$

5.3 Weighted Kernel Ridge Regression

One of the assumptions of the ridge regression is constance variance. That is, there is $Var(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$. Sometimes, this assumption is violated such that $\mathrm{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{A}$, where $\mathbf{A} \neq \mathbf{I}$ is an $n \times n$ matrix. This is also called heteroscedasticity. A special scenario is that $\mathbf{A}$ is a diagonal matrix with unequal diagonal elements in which we assume the random errors are uncorrelated but with unequal variances. $\mathbf{A}$ is a non-singular positive definite matrix since $\sigma^2\mathbf{A}$ is the covariance matrix of the random errors. We use $\mathbf{C}$ to denote the squared root of $\mathbf{A}$. Then, $\mathbf{C}$ is a nonsingular symmetric matrix such that $\mathbf{C^T C} = \mathbf{CC} = \mathbf{A}$. Left-Multiplying the both sides of

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

by $\mathbf{C^{-1}}$, a new regression model can be defined by

$$\mathbf{y}^* = \mathbf{X}^*\boldsymbol{\beta} + \boldsymbol{\varepsilon}^*, \tag{5.20}$$

where $\mathbf{y}^* = \mathbf{C^{-1}y}$, $\mathbf{X}^* = \mathbf{C^{-1}X}$ and $\boldsymbol{\varepsilon}^* = \mathbf{C^{-1}}\boldsymbol{\varepsilon}$.

There are $E(\boldsymbol{\varepsilon}^*) = \mathbf{C^{-1}}E(\boldsymbol{\varepsilon}^*) = 0$ and $Var(\boldsymbol{\varepsilon}^*) = \sigma^2\mathbf{I}$ since

$$\begin{aligned} Var(\boldsymbol{\varepsilon}^*) &= [\boldsymbol{\varepsilon}^* - E(\boldsymbol{\varepsilon}^*)][\boldsymbol{\varepsilon}^* - E(\boldsymbol{\varepsilon}^*)]' \\ &= E(\boldsymbol{\varepsilon}^*\boldsymbol{\varepsilon}^{*T}) \\ &= E(\mathbf{C^{-1}}\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^{\mathbf{T}}\mathbf{C^{-1}}) \\ &= \mathbf{C^{-1}}\mathbf{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^{\mathbf{T}})\mathbf{C^{-1}} \\ &= \sigma^2\mathbf{C^{-1}}\mathbf{A}\mathbf{C^{-1}} \\ &= \sigma^2\mathbf{C^{-1}}\mathbf{C}\mathbf{C}\mathbf{C^{-1}} \\ &= \sigma^2\mathbf{I}. \end{aligned}$$

The generalized least squares estimator of $\boldsymbol{\beta}$ is

$$\begin{aligned} \widehat{\boldsymbol{\beta}} &= (\mathbf{X}^{\mathbf{T}}\mathbf{C^{-1}}\mathbf{C^{-1}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{C^{-1}}\mathbf{C^{-1}}\mathbf{y} \\ &= (\mathbf{X}^{\mathbf{T}}\mathbf{A^{-1}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{A^{-1}}\mathbf{y} \end{aligned}$$

and the predicted $\mathbf{y}^*$ is

$$\hat{\mathbf{y}}^* = \mathbf{X}^*\widehat{\boldsymbol{\beta}} = \mathbf{C^{-1}}\mathbf{X}(\mathbf{X}^{\mathbf{T}}\mathbf{A^{-1}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{A^{-1}}\mathbf{y}.$$

Therefore, the original response $\mathbf{y}$ can be estimated by

$$\hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{y}}^* = \mathbf{X}(\mathbf{X}^{\mathbf{T}}\mathbf{A^{-1}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{A^{-1}}\mathbf{y}.$$

The covariance of $\widehat{\boldsymbol{\beta}}$ is

$$Cov(\widehat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^{*\mathbf{T}}\mathbf{X}^*)^{-\mathbf{1}} = \sigma^2(\mathbf{X}'\mathbf{A^{-1}}\mathbf{X})^{-\mathbf{1}}.$$

When the errors are uncorrelated, matrix $\mathbf{A}$ is a diagonal matrix. If we use the weight matrix $\mathbf{W} = \mathbf{A^{-1}}$, then

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{w}} = (\mathbf{X^T W X})^{-1}\mathbf{X^T W y}$$

is generally called the weighted least squares estimator (Montgomery, 2001).

Similarly, the weighted ridge estimator of the linear regression coefficients is given by

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{wr}} = (\mathbf{X^T W X} + \lambda\mathbf{I})^{-1}\mathbf{X^T W y}. \tag{5.21}$$

This dissertation extends the weighted ridge estimator to the feature space when heteroscedasticity exists. Given a kernel ridge regression model in the feature space defined by

$$\mathbf{y} = \boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

with $E(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{A}$, the transformation is given by

$$\mathbf{C^{-1}y} = \mathbf{C^{-1}}\boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Regress $\mathbf{C^{-1}y}$ on $\mathbf{C^{-1}}\boldsymbol{\Phi}(\mathbf{X})$, the generalized kernel ridge estimator of $\boldsymbol{\beta_r}$ is

$$\begin{aligned}\widehat{\boldsymbol{\beta}} &= \boldsymbol{\Phi}(\mathbf{X})^{\mathbf{T}}\mathbf{C^{-1}}(\mathbf{C^{-1}}\boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Phi}(\mathbf{X})^{\mathbf{T}}\mathbf{C^{-1}} + \lambda\mathbf{I})^{-1}\mathbf{C^{-1}y} \\ &= \boldsymbol{\Phi}(\mathbf{X})^{\mathbf{T}}\mathbf{C^{-1}}(\mathbf{C^{-1}KC^{-1}} + \lambda\mathbf{I})^{-1}\mathbf{C^{-1}y}.\end{aligned}$$

The estimated $\mathbf{C^{-1}y}$ is defined by

$$\begin{aligned}\widehat{\mathbf{C^{-1}y}} &= \mathbf{C^{-1}}\boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Phi}(\mathbf{X})^{\mathbf{T}}\mathbf{C^{-1}}(\mathbf{C^{-1}KC^{-1}} + \lambda\mathbf{I})^{-1}\mathbf{C^{-1}y} \\ &= \mathbf{C^{-1}KC^{-1}}(\mathbf{C^{-1}KC^{-1}} + \lambda\mathbf{I})^{-1}\mathbf{C^{-1}y}.\end{aligned}$$

Solving $\hat{\mathbf{y}}$ of the above equation, there is

$$\hat{\mathbf{y}} = \mathbf{KC^{-1}}(\mathbf{C^{-1}KC^{-1}} + \lambda\mathbf{I})^{-1}\mathbf{C^{-1}y}.$$

Let the weight matrix $\mathbf{W} = \mathbf{A^{-1}}$, there is $\mathbf{C^{-1}} = \mathbf{A^{-\frac{1}{2}}} = \mathbf{W^{\frac{1}{2}}}$. The weighted kernel ridge estimate has the form of

$$\hat{\mathbf{y}} = \mathbf{K}\mathbf{W}^{\frac{1}{2}}(\mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}} + \lambda\mathbf{I})^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{y}. \tag{5.22}$$

The new observations can be predicted as:

$$\hat{\mathbf{y}}_{\mathbf{new}} = \mathbf{K}_{\mathbf{new}}\mathbf{W}^{\frac{1}{2}}(\mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}} + \lambda\mathbf{I})^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{y}. \tag{5.23}$$

The covariance matrix of the estimated coefficients has to be defined in a different way since $\mathbf{\Phi}(\mathbf{X})$ is unknown. Using the kernel ridge regression model redefined in (5.15) and the covariance matrix form given by (5.19), the weighted kernel ridge estimator is redefined by

$$\hat{\boldsymbol{\beta}}^{*}_{\boldsymbol{wr}} = \mathbf{W}^{\frac{1}{2}}\left[\mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}} + \lambda\mathbf{I}\right]^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{y}. \tag{5.24}$$

The covariance of $\hat{\boldsymbol{\beta}}^{*}_{\mathbf{wr}}$ has the explicit form defined as:

$$Cov(\widehat{\boldsymbol{\beta}}^{*}_{wr}) = \sigma^2\left\{\mathbf{W}^{\frac{1}{2}}\left[\mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}} + \lambda\mathbf{I}\right]^{-1}\mathbf{W}^{\frac{1}{2}}\right\}^{\mathbf{2}}, \tag{5.25}$$

where the estimated σ^2 is given by

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \hat{\mathbf{y}})^{\mathbf{T}}(\mathbf{y} - \hat{\mathbf{y}})}{n - \mathrm{trace}(\mathbf{H})},$$

and where $\mathbf{H} = \mathbf{K}\mathbf{W}^{\frac{1}{2}}(\mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}} + \lambda\mathbf{I})^{-1}\mathbf{W}^{\frac{1}{2}}$ is the hat matrix of the weighted kernel ridge regression.

The prior information about the weights has to be known or the weights can be estimated as a function of the residuals or regressors (Montgomery, 2001). For instance, the moving averages of the squared residuals of the un-weighted kernel ridge regression (Silverman, 1985) can be used to estimate the weights. The results of our numerical experiments indicate that the optimal ridge parameter of the weighted kernel ridge regression is generally different

from the chosen one for the regular kernel ridge regression. Therefore, it is required to search the optimal ridge parameter for the weighted kernel ridge regression.

5.4 Interval Estimates for Kernel Ridge Regression

This dissertation also derives the interval estimates for kernel ridge regression following the interval estimates of a linear ridge regression problem (Montgomery, 2001). To give a confidence interval estimate of $\mu_y|_{x_0}$, we define the standard error of $\hat{\mu}_y|_{x_0}$ as:

$$\begin{aligned} se(\hat{\mu}_y|_{x_0}) &= \sqrt{Var(\hat{y}|_{x_0})} \\ &= \sqrt{\sigma^2 \mathbf{k}(\mathbf{x_0}, \mathbf{X}) \mathbf{W}^{\frac{1}{2}} (\mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}} + \lambda \mathbf{I})^{-\mathbf{2}} \mathbf{W}^{\frac{1}{2}} \mathbf{k}(\mathbf{X}, \mathbf{x_0})}. \end{aligned} \tag{5.26}$$

Under the large sample assumption and applying the Central Limit Theorem (CLT), we have

$$\hat{\mu}_y|_{x_0} \sim N\left(\mu_y|_{x_0}, se^2(\hat{\mu}_y|_{x_0})\right).$$

Consequently, a $100(1-\alpha)\%$ confidence interval on $\mu_y|_{x_0}$ is defined by

$$\hat{\mu}_y|_{\mathbf{x_0}} \pm z_{\alpha/2} se(\hat{\mu}_y|_{\mathbf{x_0}}), \tag{5.27}$$

where $\hat{\mu}_y|_{\mathbf{x_0}} = \hat{y}|_{\mathbf{x_0}}$ and $se(\hat{\mu}_y|_{\mathbf{x_0}})$ is defined in (5.26).

Similarly, to compute the prediction interval for a future observation $y_0|_{\mathbf{x_0}}$, we use the standard error of $\tau = y_0|_{\mathbf{x_0}} - \hat{y}_0|_{\mathbf{x_0}}$ to find the margin of error (Montgomery, 2001). Since $y_0|_{\mathbf{x_0}}$ is independent of $\hat{y}_0|_{\mathbf{x_0}}$, this standard error can be defined as:

$$\begin{aligned} se(\tau) &= \sqrt{Var(y_0|_{\mathbf{x_0}}) + Var(\hat{y}_0|_{\mathbf{x_0}})} \\ &= \sqrt{\sigma^{\mathbf{2}} \frac{\mathbf{1}}{\mathrm{w}_0} + \sigma^{\mathbf{2}} \mathbf{k}(\mathbf{x_0}, \mathbf{X}) \mathbf{W}^{\frac{1}{2}} (\mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}} + \lambda \mathbf{I})^{-\mathbf{2}} \mathbf{W}^{\frac{1}{2}} \mathbf{k}(\mathbf{X}, \mathbf{x_0})} \\ &= \sqrt{\sigma^{\mathbf{2}} \left[\frac{\mathbf{1}}{\mathrm{w}_0} + \mathbf{k}(\mathbf{x_0}, \mathbf{X}) \mathbf{W}^{\frac{1}{2}} (\mathbf{W}^{\frac{1}{2}} \mathbf{K} \mathbf{W}^{\frac{1}{2}} + \lambda \mathbf{I})^{-\mathbf{2}} \mathbf{W}^{\frac{1}{2}} \mathbf{k}(\mathbf{X}, \mathbf{x_0}) \right]}, \end{aligned} \tag{5.28}$$

where w_0 is the weight corresponding to $y_0|_{\mathbf{x_0}}$. If $\mathbf{x_0}$ is not an observation from the training data, one must estimate w_0. The estimated w_0 may not be reliable given an $\mathbf{x_0}$ that is far beyond the span of the training data. Since the point estimate of $y|_{x_0}$ is also $\hat{y}|_{x_0}$, a $100(1-\alpha)\%$ prediction interval on $y|_{x_0}$ is defined as:

$$\hat{y}|_{x_0} \pm z_{\alpha/2} se(\tau). \tag{5.29}$$

5.5 Choosing Optimal Kernel Ridge Parameter

The ridge parameter λ prevents from collinearity problems and controls the generalization of the ridge regression model. In the literature of the linear ridge regression, there is massive and controversial discussion regarding the techniques of choosing the ridge parameter. There is no single ridge parameter estimator which is proven to be the best overall (Montgomery, 2001).

Some techniques (Engle et al., 2000; Morozov, 1984) require the knowledge of the noise level, which is not possible in many cases. The most widely-used visual tool in machine learning is the L-curve (Hansen, 1998) in which the residual norm is plotted against the norm of the estimated ridge regression coefficients at the different values of the ridge constant. The optimal ridge constant is found at the corner of the "L" curve where both the residual norm and the coefficient norm are relative small. The advantage of this method is that the researchers do not need the knowledge about the noise level. However, the L-curve method has been proven to be nonconvergent (Vogel, 1996; Leonov and Yagola, 1997). Either the corner of the L-curve could be difficult to locate or there exits multiple corners. A similar visual way suggested by Hoerl and Kennard is to inspect the ridge trace (Marquardt, 1975). The ridge trace is a plot of the estimated coefficients $\hat{\beta}_r^i$ versus λ. For the regression applications with collinearity problems, the magnitudes of the least squares estimators are relative big. The magnitudes of the estimated coefficients decrease and tend to stabilize with the increasing of λ. The objective of observing ridge trace is to select a reasonably

small λ value at which the coefficients $\hat{\beta}_r^i$ are stable. However, it is subjective to locate the "stable" point. These two methods can not be applied in a model selection procedure since neither of them is an automatic ridge parameter finder.

Hoerl, Kennard, and Baldwin (Hoerl et al., 1975) proposed an choice of λ defined by

$$\lambda_{HKB} = \frac{p\hat{\sigma}^2}{\hat{\boldsymbol{\beta}}^T \hat{\boldsymbol{\beta}}}, \tag{5.30}$$

where p is the number of estimated regression coefficients, $\hat{\boldsymbol{\beta}}$ is the no-constant-term ordinary least-squares (OLS) estimator, and $\hat{\sigma}^2$ is the estimated response variance in a OLS regression. Hoerl and Kennard (Hoerl and Kennard, 1976) further proposed an iterative estimation based on (5.30). Several authors interpreted that the ridge regression is closely related to Bayesian estimation. The method of ordinary least-squares (OLS) can be viewed as a Bayes estimator using an unbounded uniform prior distribution when estimating the coefficients. The ridge estimator is the result based on a prior distribution. Theil and Goldberger(1961) have introduced a procedure called mixed estimation which can be numerically equivalent as the ridge regression. Following Sclove(Sclove, 1973), suppose the prior distribution of $\boldsymbol{\beta_r}$ is

$$\boldsymbol{\beta_r} \sim N(\mathbf{0}, \sigma^{\mathbf{2}}\mathbf{I}) \tag{5.31}$$

and independent from the random error $\boldsymbol{\varepsilon}$. The ridge constant can be estimated as

$$\lambda_{Bayes} = \frac{\hat{\sigma}^2}{\hat{\sigma}^2_{\beta_r}}, \tag{5.32}$$

where σ^2 is the error variance estimated by OLS and $\hat{\sigma}^2_{\beta_r}$ is defined by

$$\hat{\sigma}^2_{\beta_r} = \frac{\mathbf{y^T y} - n\hat{\sigma}^2}{tr(\mathbf{X^T X})}.$$

However, the above two analytic methods require the estimate of the error variance using the OLS solution. In the kernel case, the so called OLS solution leads to the zero lack-of-fit. Therefore, we are not able to imbed these two methods to the kernel ridge regression.

Some techniques perform grid searching in the range of $[0, 1]$. Generalized cross-validation (GCV), proposed by Wahba (Wahba et al., 1979), is a widely used stochastic method to search the optimal ridge constant. Using GCV, the optimal ridge constant is the minimizer of the statistic defined as:

$$GCV = \frac{\sum_{i=1}^{n} e_{i,k}^2}{\{n - [1 + tr(\mathbf{H}_r)]\}^2}, \tag{5.33}$$

where $\mathbf{H}_r = \mathbf{X}(\mathbf{X^T X} + \lambda \mathbf{I})^{-1}\mathbf{X^T}$ is equivalent to the hat matrix in ordinary least squares. In kernel ridge regression the equivalent hat matrix is $\mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1}$. The assumption of using GCV is the white (independent) Gaussian noise. Also base on the white Gaussian noise assumption, Mallows (1973) proposed the C_L statistic defined by

$$C_L = \frac{SS_{Res}(\lambda)}{\hat{\sigma}^2} - n + 2 + 2tr(\mathbf{H_r}(\lambda)). \tag{5.34}$$

The optimal λ value is the minimizer of C_L. Leave-one-out cross validation (LOOCV) is a widely used criterion to search the optimal ridge parameter since it has a close form (Wahba, 1990) which avoids the heavy computation. The squared error is defined by

$$\frac{\|[\mathbf{diag}(\mathbf{I} - \mathbf{H})]^{-1}(\mathbf{I} - \mathbf{H})\mathbf{y}\|^2}{n}, \tag{5.35}$$

where n is the number of training observations, $\mathbf{H} = \mathbf{K}(\mathbf{K} + \lambda \mathbf{I})^{-1}$ is the hat matrix of KRR, and $diag(\mathbf{I} - \mathbf{H})$ is a diagonal matrix whose diagonal elements are the same as those of $\mathbf{I} - \mathbf{H}$. Recently, Bozdogan's information measure approach has been successfully used for power plant data to choose the optimal ridge constant of a linear ridge regression model (Urmanov, 2002). We derive its ICOMP extension to the feature space in the next section and compare its performance with GCV and LOOCV using the benchmark data sets.

5.6 Model Selection Using ICOMP

For a kernel ridge regression model, the form of the kernel function, the parameters of the kernel function and the ridge parameter affect the model generalization as well as

the goodness-of-fit. This dissertation uses ICOMP as the criterion to conduct the model selection for KRR applications. In addition, this dissertation uses ICOMP to choose the best subset of independent variables given the selected kernel function. The subset selection can further decrease the model complexity, increase the model generalization and provide better interpretation to the model applied. In this section details the ICOMP form which scores the candidate models.

Recall that the general form of ICOMP(C_{1F}) or ICOMP(C_1) for the multiple regression model is defined by

$$\text{ICOMP}(\widehat{\text{Cov}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}})})_{Reg} = -2\log L(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}}) + 2C_{1F}(\widehat{\text{Cov}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}})})$$

or

$$\text{ICOMP}(\widehat{\text{Cov}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}})})_{Reg} = -2\log L(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}}) + 2C_{1}(\widehat{\text{Cov}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}})}).$$

In the linear ridge regression model, a penalized log-likelihood (PLL) function defined by

$$PLL(\boldsymbol{\beta}, \sigma^2) = L(\boldsymbol{\beta}, \sigma^2) - \lambda \cdot \text{Penalty}(\boldsymbol{\beta}) \tag{5.36}$$

is applied for the maximum likelihood estimate. The quadratic penalty term, first proposed in (Good and Gaskins, 1971), is defined as:

$$\text{Penalty}(\boldsymbol{\beta}) = \frac{\boldsymbol{\beta}^{\boldsymbol{T}}\boldsymbol{\beta}}{2\sigma^2}. \tag{5.37}$$

Under the normal random error assumption, the penalized log-likelihood function of a linear ridge regression model is defined by

$$PLL(\boldsymbol{\beta}, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathbf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{2\sigma^2} - \lambda\frac{\boldsymbol{\beta}^{\boldsymbol{T}}\boldsymbol{\beta}}{2\sigma^2}. \tag{5.38}$$

Let the first derivatives of $PLL(\boldsymbol{\beta}, \sigma^2)$, with respect to $\boldsymbol{\beta}$ and σ^2 respectively, be equal to zeros, the maximum likelihood estimates (MLEs) of the parameters are defined as:

$$\begin{aligned} \widehat{\boldsymbol{\beta}} &= (\mathbf{X^T X} + \lambda \mathbf{I})^{-1} \mathbf{X^T y} \\ \widehat{\sigma^2} &= \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathbf{T}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{\beta}}{n}. \end{aligned}$$

The intuitive thinking is to embed $PLL(\boldsymbol{\beta}, \sigma^2)$ to the feature space for the kernel ridge regression replacing $\mathbf{X}$ by $\Phi(\mathbf{X})$. However, a little modification is needed since $\Phi(\mathbf{X})$ is implicit. We apply the alternative kernel ridge regression defined as $\mathbf{y} = \mathbf{K}\boldsymbol{\beta} + \varepsilon$ in (5.15), where the kernel matrix $\mathbf{K}$ is treated as the regressors. The penalized log-likelihood function of the kernel ridge regression is proposed as

$$PLL(\boldsymbol{\beta}, \sigma^2)_{KRR} = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{(\mathbf{y} - \mathbf{K}\boldsymbol{\beta})^{\mathbf{T}} (\mathbf{y} - \mathbf{K}\boldsymbol{\beta})}{2\sigma^2} - \lambda \frac{\boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{K} \boldsymbol{\beta}}{2\sigma^2}, \quad (5.39)$$

where $\frac{\boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{K} \boldsymbol{\beta}}{2\sigma^2}$ is the penalty term for KRR. The first derivatives of $PLL(\boldsymbol{\beta}, \sigma^2)_{KRR}$ are defined by

$$\begin{aligned} \frac{\partial PLL}{\partial \boldsymbol{\beta}} &= \frac{(\mathbf{y} - \mathbf{K}\boldsymbol{\beta})^T \mathbf{K}}{\sigma^2} - \frac{\lambda \boldsymbol{\beta}^T \mathbf{K}}{\sigma^2} \\ \frac{\partial PLL}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{(\mathbf{y} - \mathbf{K}\boldsymbol{\beta})^{\mathbf{T}} (\mathbf{y} - \mathbf{K}\boldsymbol{\beta})}{2\sigma^4} + \frac{\lambda \boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{K} \boldsymbol{\beta}}{2\sigma^4}. \end{aligned}$$

Let $\frac{\partial PLL}{\partial \boldsymbol{\beta}} = \frac{\partial PLL}{\partial \sigma^2} = 0$, the MLEs of the parameters are given by

$$\widehat{\boldsymbol{\beta}} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y} \quad (5.40)$$

$$\widehat{\sigma^2} = \frac{(\mathbf{y} - \mathbf{K}\boldsymbol{\beta})^{\mathbf{T}} (\mathbf{y} - \mathbf{K}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{K} \boldsymbol{\beta}}{n} \quad (5.41)$$

The above MLE $\widehat{\boldsymbol{\beta}}$ is the same as the least squares estimate. The second derivatives of PLL are defined as

$$\frac{\partial^2 PLL}{\partial \boldsymbol{\beta}^{\mathbf{2}}} = -\frac{\mathbf{K^2} + \lambda \mathbf{K}}{\sigma^2} \tag{5.42}$$

$$\frac{\partial^2 PLL}{\partial (\sigma^2)^2} = \frac{n}{2\sigma^4} - \frac{(\mathbf{y} - \mathbf{K}\boldsymbol{\beta})^{\mathbf{T}}(\mathbf{y} - \mathbf{K}\boldsymbol{\beta})}{\sigma^6} - \frac{\lambda \boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{K} \boldsymbol{\beta}}{\sigma^6} \tag{5.43}$$

$$\frac{\partial^2 PLL}{\partial \boldsymbol{\beta} \partial \sigma^2} = \left[\frac{\partial^2 PLL}{\partial \sigma^2 \partial \boldsymbol{\beta}^{\boldsymbol{T}}}\right]^T = \frac{-(\mathbf{y} - \mathbf{K}\boldsymbol{\beta})^{\mathbf{T}}\mathbf{K} + \lambda \boldsymbol{\beta}^{\boldsymbol{T}} \mathbf{K}}{\sigma^4}. \tag{5.44}$$

The $\widehat{\boldsymbol{\beta}}$ and $\widehat{\sigma^2}$ are MLEs indeed since

$$\frac{\partial^2 PLL}{\partial \boldsymbol{\beta}^{\mathbf{2}}} < 0 \qquad \text{and} \qquad \frac{\partial^2 PLL}{\partial (\sigma^2)^2} < 0.$$

The Fisher's Information Matrix (FIM) is defined as

$$\begin{aligned} \mathcal{F} &= \begin{bmatrix} -E\left(\frac{\partial^2 PLL}{\partial \boldsymbol{\beta}^{\mathbf{2}}}\right) & -E\left(\frac{\partial^2 PLL}{\partial \sigma^2 \partial \boldsymbol{\beta}^{\boldsymbol{T}}}\right) \\ -E\left(\frac{\partial^2 PLL}{\partial \boldsymbol{\beta} \partial \sigma^2}\right) & -E\left(\frac{\partial^2 PLL}{\partial (\sigma^2)^2}\right) \end{bmatrix}_{\widehat{\boldsymbol{\beta}}, \widehat{\sigma^2}} \\ &= \begin{bmatrix} \frac{K^2 + \lambda K}{\hat{\sigma}^2} & \frac{\lambda K \boldsymbol{\beta}}{\hat{\sigma}^4} \\ \frac{\lambda \boldsymbol{\beta}^{\boldsymbol{T}} K}{\hat{\sigma}^4} & \frac{n}{2\hat{\sigma}^4} + \frac{\lambda \boldsymbol{\beta}^{\boldsymbol{T}} K \boldsymbol{\beta}}{\hat{\sigma}^6} \end{bmatrix}. \end{aligned} \tag{5.45}$$

Given the above derivation, assuming normal random error, $ICOMP(C_{1F})$ for the univariate KRR can be defined as

$$\begin{aligned} \text{ICOMP}(\widehat{\text{Cov}(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\sigma}^{\mathbf{2}}})})_{KRR} &= -2PLL(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\sigma}^{\mathbf{2}}})_{KRR} + 2C_{1F}(\widehat{\text{Cov}(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\sigma}^{\mathbf{2}}})}) \\ &= n\log(2\pi) + n\log(\widehat{\sigma^2}) + n + 2C_{1F}(\widehat{\text{Cov}(\widehat{\boldsymbol{\beta}}, \widehat{\sigma^2})}) \end{aligned} \tag{5.46}$$

where $\widehat{\boldsymbol{\beta}}$ and $\widehat{\sigma^2}$ are defined in (5.40) and (5.41) respectively. Under the large sample assumption, $\widehat{\text{Cov}(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\sigma}^{\mathbf{2}}})}$ can be calculated using the inverse fisher information matrix (IFIM) $\mathcal{F}^{-1}$. ICOMP(C_1) can be defined similarly.

Alternatively, we may score the complexity of the exact covariance of $\widehat{\boldsymbol{\beta}}$ defined previously (5.19), which is

$$Cov(\widehat{\boldsymbol{\beta}}) = \sigma^2(K + \lambda \mathbf{I})^{-2}.$$

5.7 Numerical Results

5.7.1 Simulated Sinc Function Data

This section demonstrates the model selection results using ICOMP as the criterion to choose the appropriate kernel function and ridge parameter λ. Like many other machine learning researchers, we use the popular *sinc* function function defined by

$$y = sinc(x) = \frac{sin(\pi x)}{\pi x}. \tag{5.47}$$

We generated 121 uniformly spaced observations of $\mathbf{x}$ in the range of [-6, 6] and calculated the corresponding values of $sinc(\mathbf{x})$. To show if the Gaussian RBF kernel is a good candidate to model this nonlinear function, we applied the Gaussian RBF kernel with $a = 1$ to the noise-free *sinc* function. It is shown that good fitting can be achieved when λ is equal to 0.01 or less (goodness-of-fit is less sensitive to λ in this range). Note, it is possible that there exist other combinations of a and λ which can lead to the similar good fitting since this is a nonlinear modeling procedure.

We add i.i.d. normal random noise to the *sinc* function. Two noise levels were tested. $N(0, 0.04^2)$ is corresponding to the noise-signal-ratio of 15% and $N(0, 0.14^2)$ is corresponding to the noise-signal-ratio of 51%. To understand the influence of the training sample size, we uses 121 training observations and 50 training observations respectively. 80 additional noisy observations (different from the training observations) are generated in the same $\mathbf{x}$ range to be used as the testing data.

ICOMP and the leave-one-out cross validation (LOOCV) are compared in selecting the optimal ridge parameter and kernel function. To select the optimal form of ICOMP for KRR, the simulation is repeated 100 times using the Gaussian RBF kernel function under

the two different noise levels and two different training sample sizes. In each simulation, different random errors are generated for the training and testing data. The ridge parameters λ are selected from

$$[10^{-7}, 10^{-6}, 10^{-5}, \ldots, 0.1, 0.2, 0.3, \ldots, 0.9, 1].$$

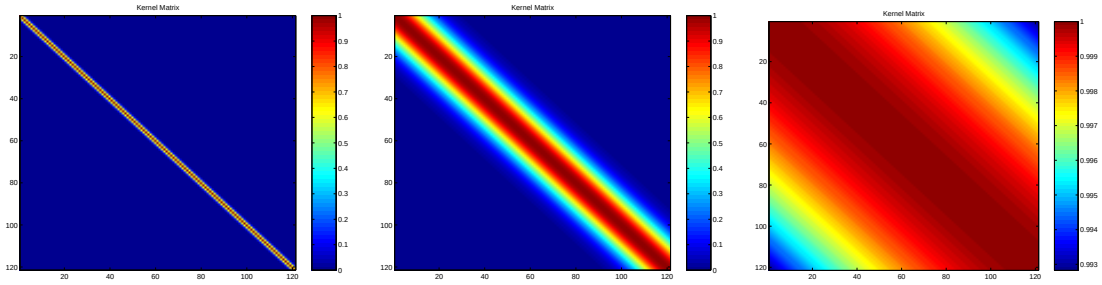
One of the critical issues is choosing the range of the candidate scale parameters a. The Gaussian RBF kernel is a function of the squared distance ($\|\mathbf{x} - \mathbf{z}\|^{\mathbf{2}}$) of two observations in the input space. In other words, the Gaussian RBF kernel is a similarity measurement. For the Gaussian RBF kernel, $0 \leq k(\mathbf{x}, \mathbf{z}) \leq 1$ with $k(\mathbf{x}, \mathbf{z}) = 1$ for two identical observations.

We now consider the following two extreme cases:

- When a is very small, $k(\mathbf{x_i}, \mathbf{x_j}) \longrightarrow 0$. The Gaussian RBF kernel tends to be an identity matrix, which indicates that all the observations are different to each other. KRR tries to fit each observation perfectly and tends to overfit the data under this circumstance (Figure 5.1(a));

- When a is very big, $k(\mathbf{x_i}, \mathbf{x_j}) \longrightarrow 1$. The kernel matrix tends to be a matrix with all 1s, which indicates that all the observations are the same. Since there is no variation information provided, KRR tends to underfit the data in this case (Figure 5.1(c)).

Sorting the training observations $\mathbf{x}$ in ascending order, it is easier to demonstrate these two kinds of high-dimensional kernel matrices with square images in which the color code of a patch is corresponding to the element value in the matrix. The optimal kernel function should be like Figure 5.1 (b) whose adjacent observations are similar and far way observations are less relevant. To select the range of a, we plot the logirthm of the condition number of the kernel matrix versus the different values of a (Figure 5.2). It can be observed that there is a steep drop in condition number when a is smaller than 0.3. The condition numbers of $\mathbf{K}$ are 1×10^{18}, 1.76×10^{8} and 69 for $a = 0.3$, $a = 0.2$ and $a = 0.1$ respectively. The significant decreasing of the condition number is due to the limitation of the computer precision. It is more appropriate to let $a \geq 0.3$. For the sinc function, the scale parameters



(a) (b) (c)

Figure 5.1: Image Demonstration of different Gaussian RBF kerenel matrices. (a) Overfitting ($a = 0.1$). (b) Good Generalization ($a = 1$). (c) Underfitting ($a = 100$.)

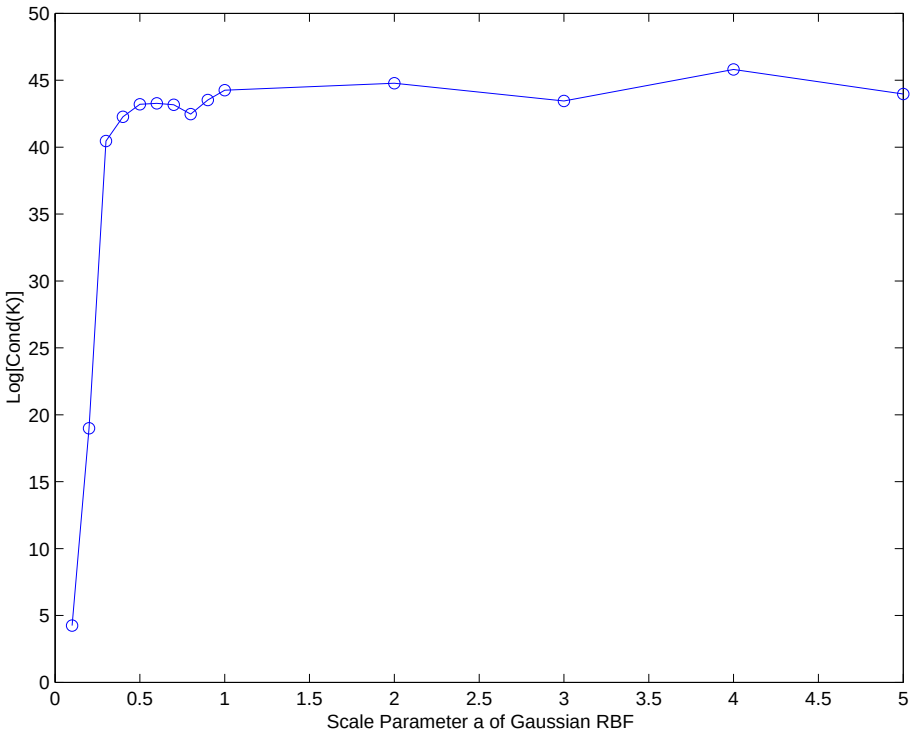


Figure 5.2: Sinc Data: Log of the Condition Number of Kernel Matrix

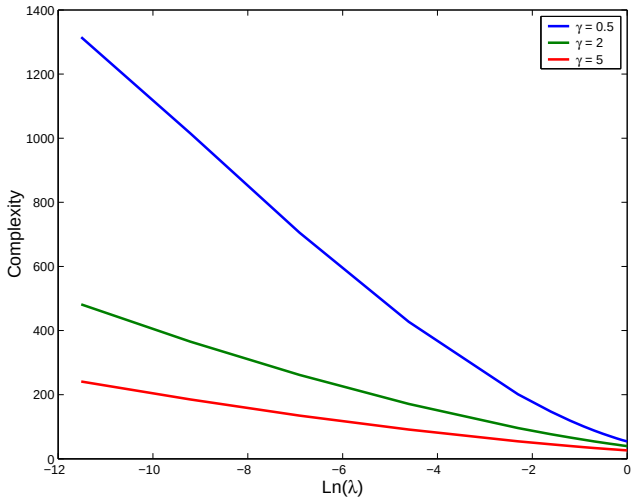


Figure 5.3: Complexity versus Ridge Parameter and Scale Parameter

a are selected from the range of [0.3, 5]. Beyond this range, the Gaussian RBF kernel will either seriously underfit or overfit the data.

We first use the sinc function to demonstrate the trend of ICOMP, lack-of-fit and the complexity with the changing of ridge parameters (0.00001 to 1) and the scale parameters ([0.5, 2, 5]) of the Gaussian RBF kernel. The chosen three scale parameters lead to overfitting, appropriate fitting and underfitting respectively. We see that the complexity of the model decreases as the ridge parameter increases and the complexity of the model decreases as the scale parameter increases, which tends to underfit (Figure 5.3) the data. The lack-of-fit changes adversely (Figure 5.4). ICOMP finds the balance point (Figure 5.5) and the resulting model has good generalization as well as good fit.

Four forms of the complexity measure are compared. They are:

a. ICOMP1 – Score the exact form of $Cov(\widehat{\boldsymbol{\beta}})$ using $C_1(\cdot)$;

b. ICOMP2 – Score the exact form of $Cov(\widehat{\boldsymbol{\beta}})$ using $C_{1F}(\cdot)$;

c. ICOMP3 – Score $C_1(\mathcal{F}^{-1})$, which applies the asymptotic covariance of all the estimated parameters;

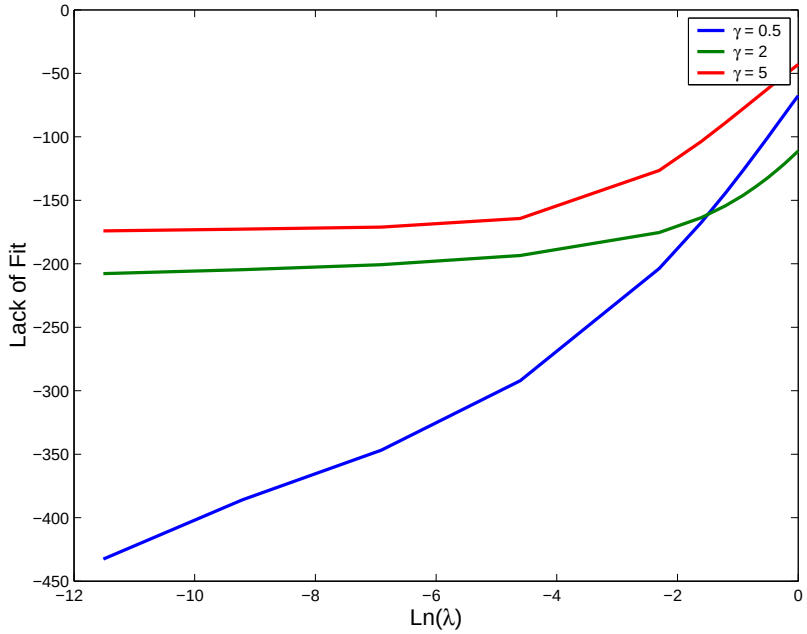


Figure 5.4: Lack-of-fit versus Ridge Parameter and Scale Parameter

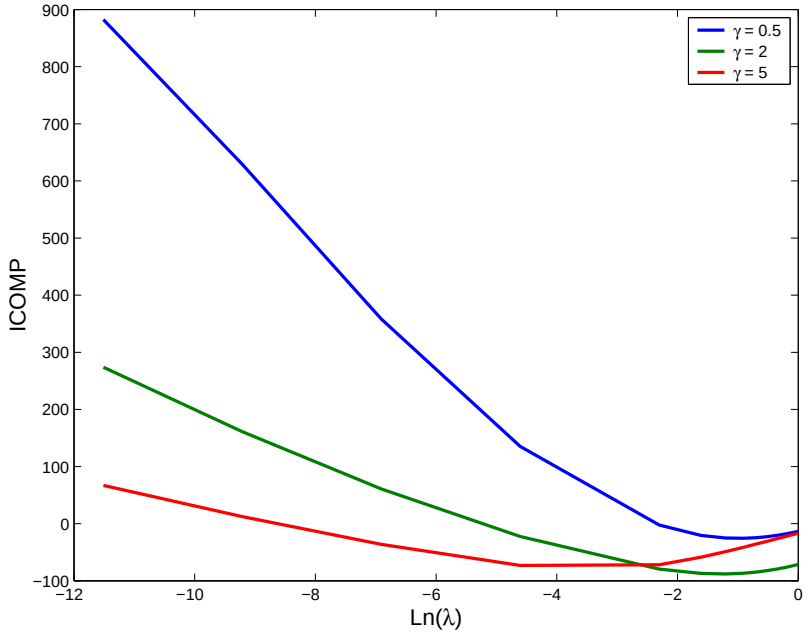


Figure 5.5: ICOMP versus Ridge Parameter and Scale Parameter

d. ICOMP4 – Score $C_{1F}(\mathcal{F}^{-1})$.

The simulation results are summarized in Table 5.1 to Table 5.4 with different noise (N) and signal (S) ratios and sample sizes. A good model selection criterion should find the models with low average and low standard deviation of MSE for the testing data out of 100 simulations. It is obvious that ICOMP1 outperforms the other forms in all noise levels and sample sizes. And, the performance of ICOMP1 is similar to LOOCV. If we use MSE of the training data as the estimate of the real noise variance σ^2, both ICOMP1 and LOOCV have the similar performance whose estimates are close to the real number of 0.0016. It is also observed (Figure 5.6) that the models selected by ICOMP1 are more consistent compared with those chosen by LOOCV, which measures the generalization error only. In the following analysis, we use ICOMP1 to choose the optimal kernel functions and their parameters.

We will apply ICOMP1 to different kernel functions to demonstrate its ability to choose kernel functions. We do the comparison simply from the statistical point of view, an application may require a specific kernel function based on the inside knowledge of the area.

In a single simulation, the predictive performance of the different kernel functions are compared and summarized in Table 5.5 and Table 5.6. The optimal parameter(s) of a kernel function is/are the minimizer of ICOMP1. By comparing ICOMP, the first order B-spline is the best kernel and the Gaussian RBF kernel is the second best one when the noise level is 15%. Both models have high predictive ability in terms of MSE_{test}. However, B-spline is much more computationally intensive than the Gaussian RBF kernel. Therefore, we will use the Gaussian RBF Kernel which is almost as good as the 1^{st} order B-spline. At the 51% noise level, the B-spline with zero order outperforms the other kernels in terms of the smallest ICOMP1. However, due to its computational intensity, Sigmoid and Gaussian RBF kernels are both good second choices.

We further test the combination of LOOCV with ICOMP. That is, we use LOOCV to estimate σ^2 for ICOMP. We use the Wilcoxon Sign-Rank test to compare the matched pairs

Table 5.1: Prediction Performance of Five Criteria ($N/S = 15\%$, 100 observations)

Criterion	Ave. MSEtrain	Ave. MSEtest	Std. MSEtrain	Std. MSEtest
ICOMP1	0.001357	0.001824	0.0001944	0.0003091
ICOMP2	0.000828	0.002297	0.0002583	0.0004528
ICOMP3	0.002110	0.002537	0.0047652	0.0052156
ICOMP4	0.001211	0.001965	0.0002481	0.0003900
LOOCV	0.001357	0.001826	0.0002072	0.0003059

Table 5.2: Prediction Performance of Five Criteria ($N/S = 15\%$ 50 observations)

Criterion	Ave. MSEtrain	Ave. MSEtest	Std. MSEtrain	Std. MSEtest
ICOMP1	0.00105	0.0023	0.00024	0.00046
ICOMP2	0.00070	0.0046	0.00028	0.00512
ICOMP3	0.06000	0.0538	0.01307	0.01137
ICOMP4	0.00072	0.0097	0.00028	0.05649
LOOCV	0.00101	0.0023	0.00029	0.00064

Table 5.3: Prediction Performance of Five Criteria ($N/S = 51\%$, 100 observations)

Criterion	Ave. MSEtrain	Ave. MSEtest	Std. MSEtrain	Std. MSEtest
ICOMP1	0.0172	0.0222	0.0025	0.0037
ICOMP2	0.0130	0.0256	0.0023	0.0045
ICOMP3	0.0678	0.0711	0.0071	0.0087
ICOMP4	0.0177	0.0250	0.0084	0.0089
LOOCV	0.0169	0.0220	0.0025	0.0038

Table 5.4: Prediction Performance of Five Criteria ($N/S = 51\%$, 50 observations)

Criterion	Ave. MSEtrain	Ave. MSEtest	Std. MSEtrain	Std. MSEtest
ICOMP1	0.0146	0.0258	0.0038	0.0050
ICOMP2	0.0086	0.0565	0.0035	0.0638
ICOMP3	0.0777	0.0765	0.0112	0.0087
ICOMP4	0.0099	0.0507	0.0063	0.0395
LOOCV	0.0133	0.0264	0.0036	0.0062

Table 5.5: Comparing Kernel Functions ($N/S = 15\%$, 100 observations)

Kernel	λ	ICOMP	MSEtrain	MSEtest
Polynomial: $(< \mathbf{x}, \mathbf{z} > +1)$	1	56.45	0.07766	0.0789
Gaussian RBF: $\exp\left[-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\times 0.7^2}\right]$	0.1	-246.89	0.00168	0.00201
Cauchy: $\frac{1}{1+\frac{\|\mathbf{x}^{0.6}-\mathbf{z}^{0.6}\|^2}{0.6}}$	0.01	-240.59	0.00154	0.00215
Sigmoid: $tanh(< \mathbf{x}, \mathbf{z} > +1)$	0.01	-224.39	0.00346	0.00443
Fourier:	1	667.62	0.00037	0.4823
Spline:	0.01	-185.92	0.00159	0.00206
B-spline: $B_{2\times 1+1}(\mathbf{x}-\mathbf{z})$	0.5	-249.73	0.00161	0.00202

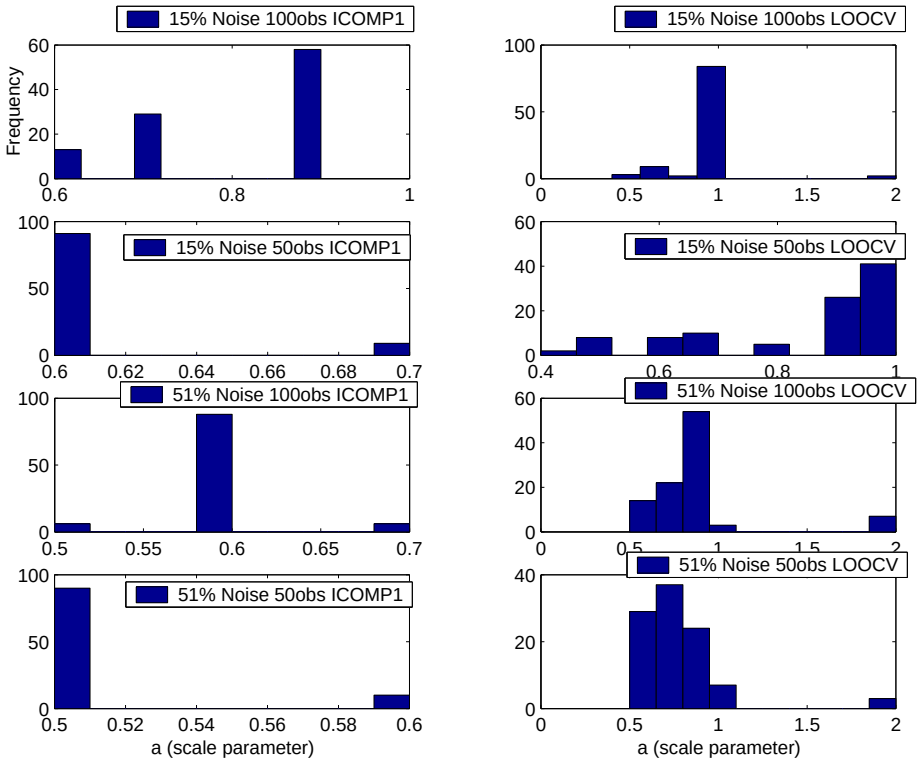


Figure 5.6: Simulated Sinc Data: Consistency of the Selected Models

Table 5.6: Comparing Kernel Functions ($N/S = 51\%$, 100 observations)

Kernel	λ	ICOMP	MSEtrain	MSEtest
Polynomial: $(< \mathbf{x}, \mathbf{z} > +1)$	1	80.72	0.0949	0.09087
Gaussian RBF: $\exp\left[-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\times 0.6^2}\right]$	0.4	-66.72	0.0135	0.0196
Cauchy: $\frac{1}{1+\frac{\|\mathbf{x}^2-\mathbf{z}^2\|^2}{2}}$	0.6	-66.33	0.0120	0.0226
Sigmoid: $tanh(0.8 < \mathbf{x}, \mathbf{z} > +1)$	0.2	-80.23	0.0178	0.0236
Fourier:	1	484.94	0.0001	0.4160
Spline:	0.01	-3.61	0.0113	0.0206
B-spline: $B_{2\times 0+1}(\mathbf{x}-\mathbf{z})$	0.6	-74.14	0.0108	0.0210

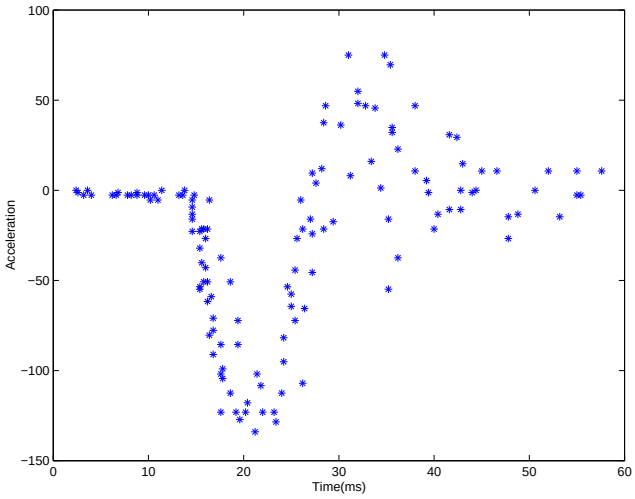


Figure 5.7: Motorcycle Benchmark Data

of the testing data MSE. The results of 100 simulation indicates that this combination is significantly better than using LOOCV alone at the significance level of 0.02.

5.7.2 Motorcycle Benchmark Data

We now apply the model selection criterion to the motorcycle benchmark data (Schmidt et al., 1981). This simple data set consists of a sequence of acceleration reading taken through time (in microseconds) for an experiment to determine the efficacy of crash helmets (Figure 5.7). Since there is only one independent variable in this data, it is obvious that the collinearity problem does not exist in the linear regression case. However, the kernel ridge regression builds high-dimensional variables in the feature space. The collinearity problem exists in the feature space, where the ridge parameter plays the regularization role. Therefore, both the kernel function and the ridge parameter control the overfitting of the fitted model. However, the impact of the ridge parameter is influenced by the parameter of the kernel function as well. For examples, for the Gaussian RBF kernel, when the the scale parameter a is relative small, the smaller ridge parameter has the effect of generating more local ridgeness (Figure 5.8). When the scale parameter is relative big, which tends to make

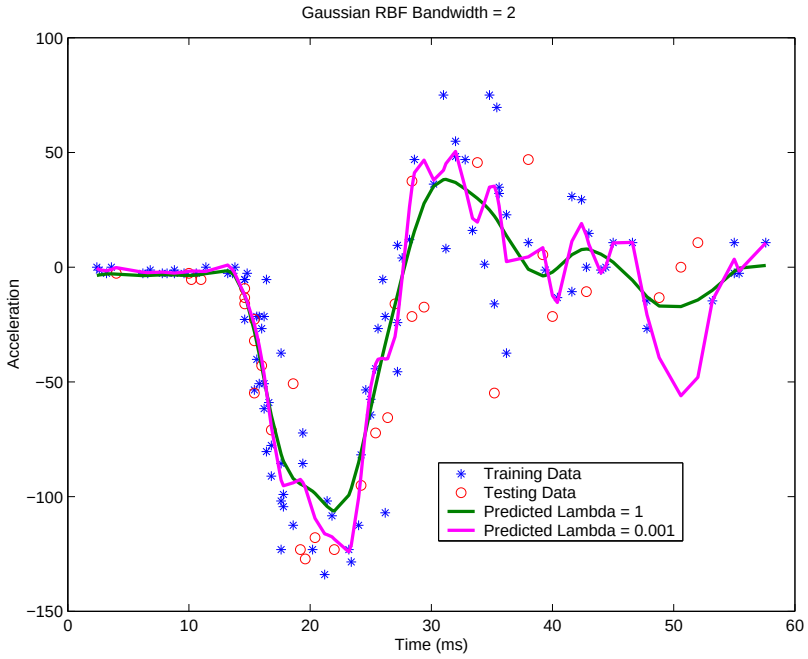


Figure 5.8: Motorcycle Benchmark Data: Impact of Ridge Parameter Bandwidth = 2

the model smoother (or simpler), smaller ridge parameters still generate more ridgeness. But the ridgeness tends to be smoother (Figure 5.9).

According the current literature (Cawley, 2002), we first use the Gaussian RBF kernel. Actually, the lower order polynomial kernel can not map this strong nonlinearity and the higher order polynomial kernel's collinearity problem can not be solved given $\lambda \leq 1$. For this data set, we applied another widely used generalization error measurement, GCV (Wahba et al., 1979), to find the optimal λ. ICOMP1 is used to compare different kernel functions given the λ values selected by GCV. It is also shown that one may use ICOMP1 as the criterion to search the optimal λ and the scale parameter of the Gaussian RBF kernel simultaneously.

The experiments are set up as follows. One hundred observations are randomly chosen to be the training data and the rest 33 observations are serving as the testing data. Different combinations of ridge parameters and the scale parameters are tested. Since the ridge parameter in a kernel ridge regression can not be zero, we choose a small value $\lambda = 10^{-5}$ as

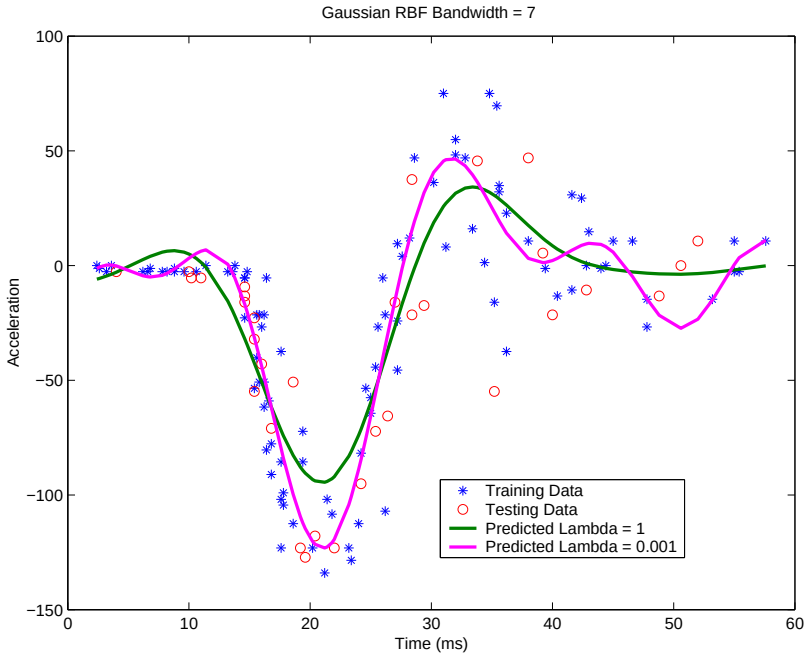


Figure 5.9: Motorcycle Benchmark Data: Impact of Ridge Parameter Bandwidth = 7

the start point. Twenty-six logarithm spaced λ values in the range of $[10^{-5}, 1]$ are selected. The chosen scale parameters are from 0.001 to 20.

We first use ICOMP1 as the criterion to choose the best combination of λ and scale parameter a simultaneously. The optimal model is chosen as the minimizer of ICOMP1 where $a = 7$ and $\lambda = 0.0398$ (Figure 5.10). Alternatively, we may use the GCV method to choose the optimal ridge parameter given the kernel function. Then, we use ICOMP as the model selection criterion to find the optimal kernel parameter (Figure 5.11). The selected optimal model has a different combination of a and λ. However, when comparing this model with the previous model in the sample plot, one can not tell the difference by eyeballing. We also compare this model with the optimal model found by the cross-validation method, which minimizes the mean squared error of the testing data (Figure 5.12). It is shown that two optimal models are similar though the cross-validation model uses a different pair of the kernel parameter and the ridge parameter, $\lambda = 0.6310$ and $a = 5$.

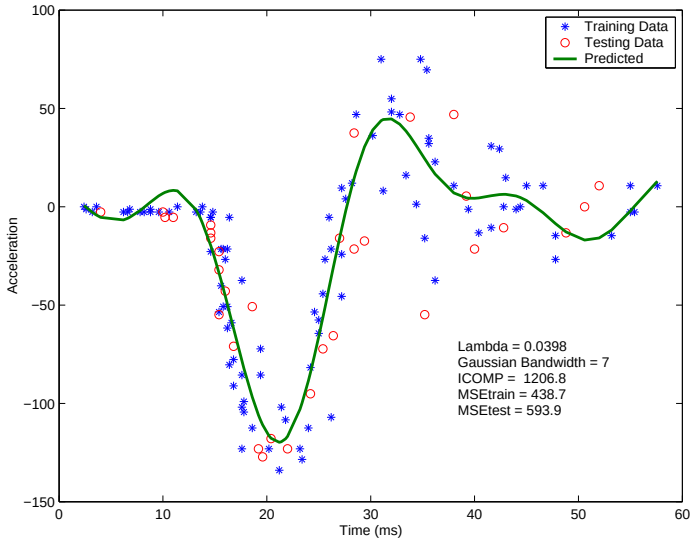


Figure 5.10: Motorcycle Data: Best Model Chosen by ICOMP

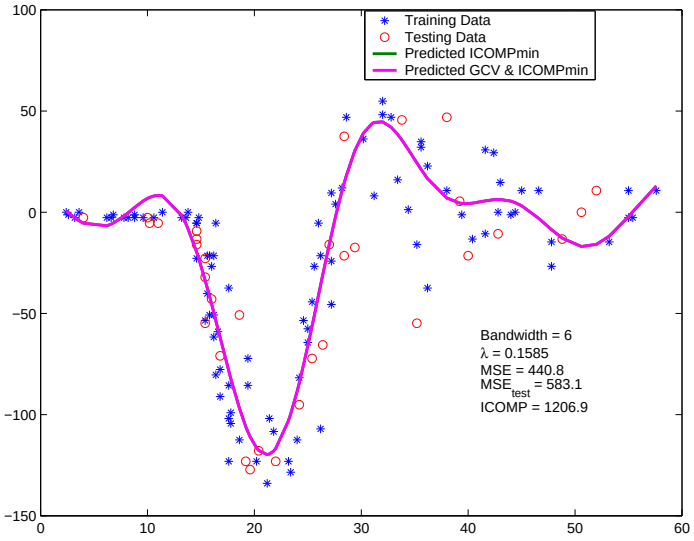


Figure 5.11: Motorcycle Data: Using GCV to choose lambda

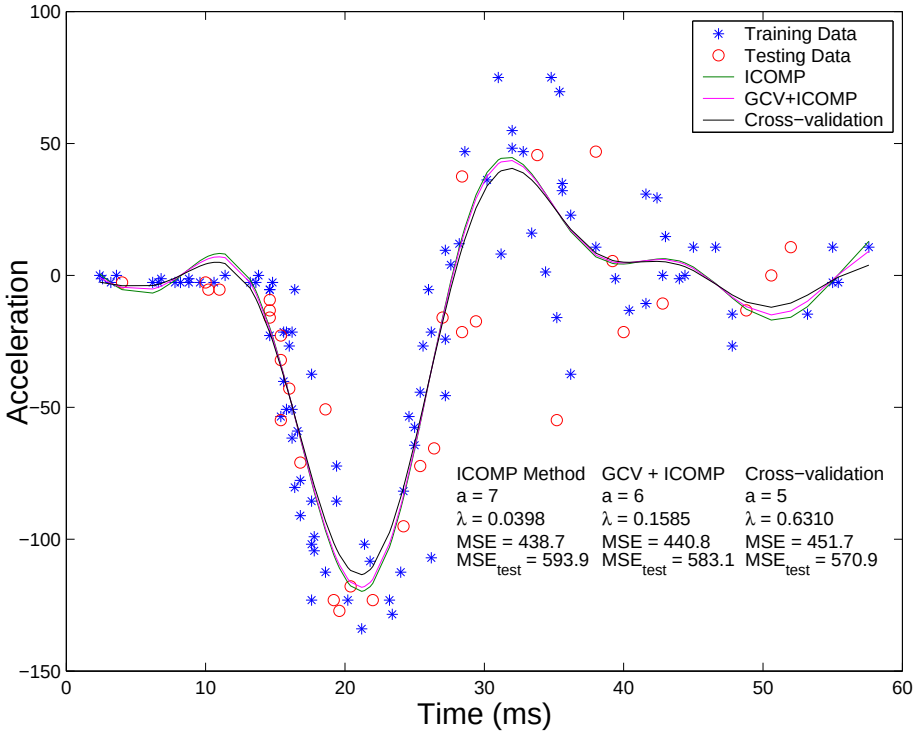


Figure 5.12: Motorcycle Data: Comparing Three Methods (Single Run)

To compare the ICOMP method with the cross-validation method thoroughly, we split the 133 observations into three groups, 100 training observations, 20 validation observations (for cross-validation) and 13 test observations (for evaluating the fitted models). One hundred runs are conducted. The observations are randomly assigned to the three groups for each run. Cross-validation, ICOMP and GCV-ICOMP hybrid are the three model selection criteria being compared.

In an experiment of 100 runs, it is indicated from Table 5.7 that ICOMP and GCV-ICOMP hybrid outperformed cross-validation in terms of average MSE of the test data. All three methods have similar standard deviations of MSE for the test data.

Based on this finding, we use all the 133 observations as the training data to build a KRR model and give the interval estimates as well as point estimates. Using ICOMP, the selected optimal scale parameter is 7 and the ridge parameter is 0.0398. One difficulty

Table 5.7: Motorcycle Data: Comparing Model Selection Methods (100 runs)

Method	Ave. MSE	Ave. MSE_{val}	Ave. MSE_{test}
1. ICOMP	457.1	543.1	573.7
2. GCV & ICOMP	458.9	542.5	573.1
3. CV	466.5	556.3	580.9
	Std. MSE	Std. MSE_{val}	Std. MSE_{test}
1. ICOMP	42.0	216.6	236.8
2. GCV & ICOMP	41.8	215.4	236.4
3. CV	42.7	212.6	237.0

about this Motor cycle data is the heteroscedasticity. One may observe this fact from the residual plot (Figure 5.13). If one would like to give the interval estimates of the acceleration assuming constant variance, it would overestimate or underestimate the uncertainty of some predicted response values (Figure 5.14).

It is obvious that the residual is a function of Time. We therefore define a diagonal weight matrix $\mathbf{W}$ such that its i^{th} diagonal element w_i is the moving average of the squared residuals of the regular kernel ridge regression (Silverman, 1985) defined as

$$w_i = \frac{c_i - d_i + 1}{\sum_{j=d_i}^{c_i} e_j^2} \tag{5.48}$$

where

$$d_i = \max(1, i-k) \qquad \text{and} \qquad c_i = \min(n, i+k)$$

k is a constant which defines the span of the moving average. k varies for different data sets. $k = 5$ is used for the motorcycle data. It is observed from our numerical results, that the ridge parameter used for the regular KRR is not applicable to WKRR. Assuming $a = 7$ of the Gaussian RBF kernel is still appropriate, the optimal ridge parameter chosen by ICOMP is $\lambda = 0.00012$ (Figure 5.15). It can be observed that the non-constant variance problem has been significantly improved (Figure 5.16).

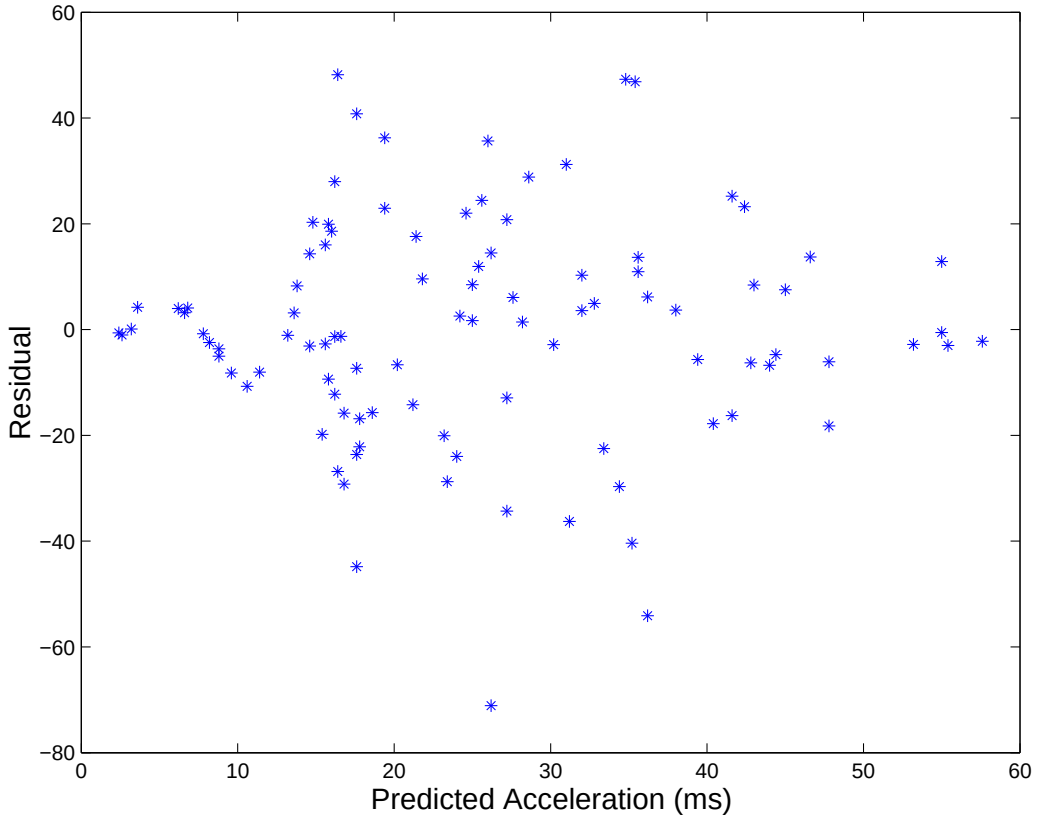


Figure 5.13: Motorcycle Data: Residual Plot of Kernel Ridge Regression

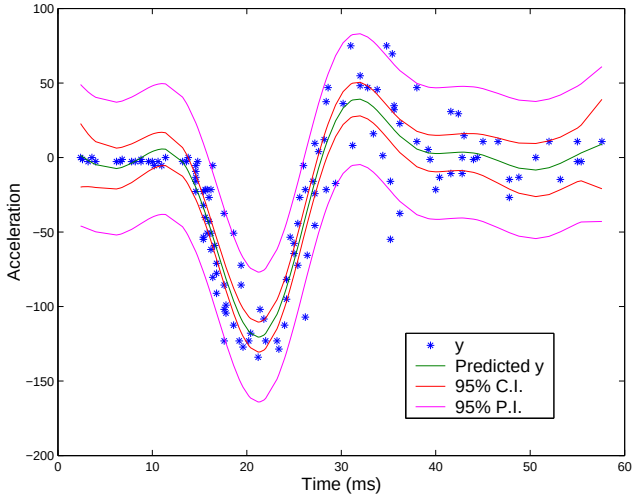


Figure 5.14: Motorcycle Data: Interval Estimates of Kernel Ridge Regression ($\lambda = 0.0398$, a = 7)

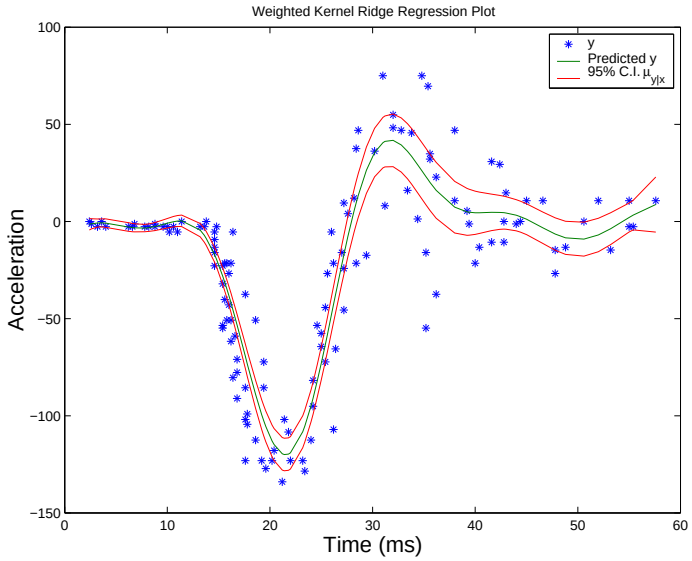


Figure 5.15: Motorcycle Data: Weighted Kernel Ridge Regression ($\lambda = 0.00012$, a = 7)

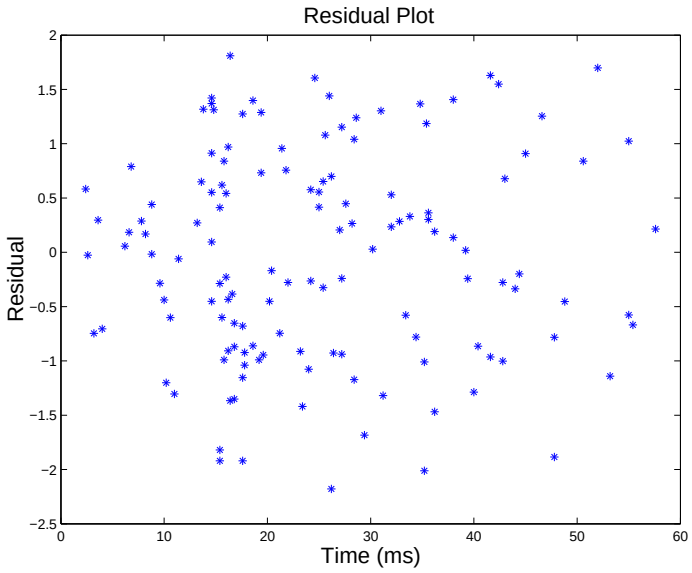


Figure 5.16: Motorcycle Data: Residual Plot of WKRR ($\lambda = 0.00012$, a = 7)

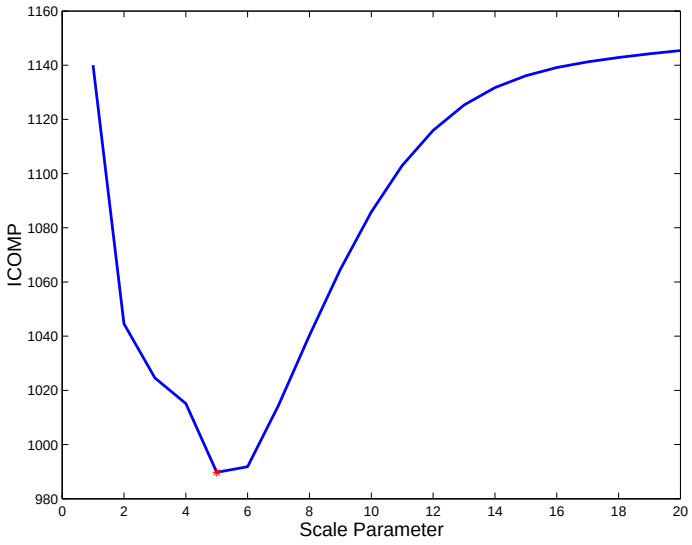


Figure 5.17: Friedman's Data: Choosing the Scale Parameter

5.7.3 Friedman's Data

We now apply ICOMP to the multiple regression applications and search for the optimal subset variables as well as the kernel function and the ridge parameter. The Friedman's data (Appendix A2) is demonstrated in this experiment. We utilize the widely used Gaussian RBF kernel. First, all the 10 regressors are included (saturated model). ICOMP is utilized to choose the optimal scale parameter of the Gaussian RBF kernel function and the ridge parameter (Figure 5.17). Then, we compare all the $2^{10} - 1 = 1023$ subset models (including at least one regressor) in terms of the ICOMP score given the selected optimal kernel. The optimal subset model is the minimizer of ICOMP. We hoped the selected model excludes $x_6 - x_{10}$.

We first use KRR to the saturated model. In 100 simulations, the average testing data MSE is 3.78 with the standard deviation of 0.39. This is comparable with the results using LOOCV (The average testing MSE is 3.56 and the standard deviation is 0.36.). In a single simulation, the optimal scale parameter (γ) chosen by ICOMP is 5 and the corresponding ridge parameter is 0.0001. We use these chosen parameters to all the subsets.

Table 5.8: GA Parameters for Friedman Data

n_{gen}	50
n_{pop}	50
P_{cross}	0.7
$P_{mutation}$	0.01
Elitism	YES
Type of Crossover	Uniform

The saturated model gives a testing error of 3.45. Its ICOMP value is 989.7. After performing the all-possible-subset-selection (APSS), it is found out that $\{x_1, x_2, x_3, x_4, x_5\}$ is the best subset, as expected, with the testing error of 2.47. Its ICOMP value is 935.7. We perform APSS to 100 simulated Friedman data sets, the same best model is chosen every time.

We can also use GA to find the optimal subset which is especially useful when the number of variables are large. The parameters of the GA procedure are configured as follows in Table 5.8. We repeat the GA procedure 100 times. A random initial population is chosen for each run. The result shows that all 100 runs found the best subset, which contains only the first 5 variables.

Chapter 6

Kernel Partial Least Squares Regression

6.1 Introduction

The Partial Least Squares (PLS) algorithm deals with the multicollinearity problem when multiple correlated variables are involved in the regression analysis. It sequentially extracts orthogonal components whose variances are in a descending order. The resulting model is regularized by retaining a small amount of important components. The PLS algorithm was first introduced by the Swedish statistician Herman Wold (1966) in the field of econometrics. The PLS algorithm became popular first in the field of chemometrics. The pioneers include Kowalski (1982), S. Wold and Martens (1983). PLS now is a widely applied techniques in many areas including psychology, economics, chemical engineering, pharmaceutical science, machine learning and image processing.

PLS is more famous for its regression application - PLS regression (PLSR). PLSR is similar to principal component regression (PCR). However, in stead of extracting the components from the covariance matrix $\mathbf{X^T X}$ of the independent variables as PCR does, PLSR extracts components from the covariance matrix between the independent variables and the dependent variables $\mathbf{X^T Y}$. In chemometrics, the univariate PLSR is generally referred as

PLS1 algorithm while the multivariate PLSR is called PLS2 algorithm (Gil and Romera, 1998). The interested readers may find the detailed and easy-to-understand description of the PLS algorithm given by Manne (1987), Geladi and Kowalski (1986b; 1986a). This chapter briefly addresses the standard algorithm of univariate PLSR (UPLSR) and multivariate PLSR (MPLSR) and their kernel extensions. Then, an information theoretic measure approach will be used to choose the number of retained components, kernel function and the optimal subset of the independent variables. Numerical results of the simulation data and benchmark data sets will be presented.

6.2 Linear Partial Least Squares Regression

The standard algorithm for computing partial least squares regression components is nonlinear iterative partial least squares (NIPALS) (Wold, 1966). Wold et al. (1984) utilized the NIPLS algorithm to PLSR. There also exist several variants (de Jong, 1993; Helland, 1988; Martens and Naes, 1989; Manne, 1987) of the PLSR algorithm in which certain vectors are normalized or not normalized. This chapter first introduces the most popular and efficient NIPALS PLSR then a variant that is convenient for the kernel extension.

6.2.1 Wold's NIPLS PLSR

Let an $n \times p$ matrix $\mathbf{X} \in \mathbf{R}^p$ represent the sample data with p independent variables and n observations. Let an $n \times q$ vector $\mathbf{Y}$ represent the corresponding dependent variables. We assume $\mathbf{X}$ and $\mathbf{Y}$ are both mean-centered such that the first extracted component reflects the covariance information instead of the mean information. The NIPALS algorithm sequentially extracts orthogonal additive components from $\mathbf{X}$ such that

$$\mathbf{X} = \sum_{\mathbf{i=1}}^{\mathbf{r}} \mathbf{t_i p_i^T} + \mathbf{E}, \tag{6.1}$$

where the unit vector $\mathbf{p_i}$, called input loading vector, indicates the direction of the i^{th} component, the latent variable (also called score variable) $\mathbf{t_i}$ is the projection of $\mathbf{X}$ on the

direction of the ith components and $\mathbf{E}$ is the residual matrix of $\mathbf{X}$, which contains the noise information. And, r is the number of the retained components. The dependent variable $\mathbf{Y}$ can be decomposed to the same number of additive components:

$$\mathbf{y} = \sum_{\mathbf{i=1}}^{\mathbf{r}} \hat{\mathbf{Y}}_\mathbf{i} + \mathbf{F} \tag{6.2}$$

where $\hat{\mathbf{y}}_\mathbf{i}$ is estimated by $\mathbf{t_i}$ and $\mathbf{F}$ is the residual matrix of $\mathbf{Y}$. Both (6.1) and (6.2) can be expressed in matrix forms:

$$\mathbf{X} = \mathbf{TP^T} + \mathbf{E} \qquad \mathbf{Y} = \hat{\mathbf{Y}} + \mathbf{F}, \tag{6.3}$$

where the columns of $\mathbf{T}$ and $\mathbf{P}$ are latent variables and loading vectors respectively.

To achieve the good model generalization, researchers usually retain the first r components, where r is decided by a model selection criterion. The detailed PLSR procedure using the NIPALS algorithm can be described as follows.

1. Begin to extract the ith components. Randomly initialize $\mathbf{u_1}$. For instance, $\mathbf{u_1}$ can be the first response of $\mathbf{Y}$.

2. Calculate the normalized covariance vector $\mathbf{w_i}$ between $\mathbf{X}$ and $\mathbf{u}$:

$$\mathbf{w_i} = \mathbf{X^T u_i} \qquad \mathbf{w_i} = \frac{\mathbf{w_i}}{\|\mathbf{w_i}\|}$$

3. Calculate the latent variable: $\mathbf{t_i} = \mathbf{X w_i}$

4. $\mathbf{q_i} = \mathbf{Y^T t}/\mathbf{t^T t} \qquad \mathbf{q_i} = \mathbf{q_i}/\|\mathbf{q_i}\|$

5. Update $\mathbf{u_i}$: $\mathbf{u_i} = \mathbf{Y q_i}$

6. Check convergence for MPLSR: Repeat step 2 to 5 until the updated $\mathbf{t_i}$ and the preceding $\mathbf{t_i}$ is within a certain error. If UPLS is conducted, initialize $\mathbf{u_i}$ using $\mathbf{y}$ and omit step 4 to 5.

7. Find the loading vector: $\mathbf{p_i} = \mathbf{X^T t_i} / \mathbf{t_i^T t_i}$

8. Normalize the loading vector: $\mathbf{p_i} = \mathbf{p_i} / \|\mathbf{p_i}\|$

9. In the linear regression case, regress $\mathbf{u_i}$ on $\mathbf{t_i}$:

   $\hat{\mathbf{u}}_\mathbf{i} = \mathbf{t_i} \hat{\mathbf{b}}_\mathbf{i}$ where $\hat{\mathbf{b}}_\mathbf{i} = (\mathbf{t_i^T t_i})^{-1} \mathbf{t_i^T u_i}$

10. Deflate $\mathbf{X}$: $\mathbf{X} = \mathbf{X} - \mathbf{t_i p_i^T}$

11. Deflate $\mathbf{Y}$: $\mathbf{Y} = \mathbf{Y} - \hat{\mathbf{u}}_\mathbf{i} \mathbf{q_i} = \mathbf{Y} - \mathbf{t_i} \hat{\mathbf{b}}_\mathbf{i} \mathbf{q_i}$, where $\mathbf{q_i} = 1$ for UPLSR.

12. i = i + 1. Repeat step 1 to step 12 until all the pre-determined number of components are extracted.

To predict the response given the new observations $\mathbf{X_{new}}$(centered or normalized), the above input loading vectors $\mathbf{p_i}$, the normalized covariance vectors $\mathbf{w_i}$ and the regression coefficients $\hat{\beta}_\mathrm{i}$ are used as the model parameters. The detailed procedure is shown below.

1. i = 1. Initialize the predicted response:

$$\hat{\mathbf{y}}_\mathbf{new} = 0$$

2. Calculate the i^{th} score variable of the new observations:

$$\mathbf{t_{new(i)}} = \mathbf{X_{new} w_i}$$

3. Update the predicted response:

$$\hat{\mathbf{y}}_\mathbf{new} = \hat{\mathbf{y}}_\mathbf{new} + \mathbf{t_{new(i)}} \hat{\beta}_\mathrm{i}$$

4. Deflate $\mathbf{X_{new}}$:

$$\mathbf{X_{new}} = \mathbf{X_{new}} - \mathbf{t_{new(i)} p_i}$$

5. i = i + 1. Repeat Step 2 - 4 until all k components are used.

6.2.2 Lew's PLS Edition

Lewi's edition of NIPALS (Lewi, 1995) normalizes the score variable $\mathbf{t_i}$ instead of the covariance vector (called weight in some literature) $\mathbf{w_i}$, making it easy to be extended to the kernel space. Lewi's NIPALS algorithm is as follows.

1. Start from i = 1.
2. $\mathbf{w_i} = \mathbf{X^T y}$
3. $\mathbf{t_i} = \mathbf{X w_i} / \|\mathbf{X w_i}\|$
4. $\mathbf{p_i} = \mathbf{X^T t_i}$ since $\|\mathbf{t_i}\| = 1$
5. $\hat{\beta}_\mathrm{i} = \mathbf{t_i^T y}$
6. $\mathbf{X} = \mathbf{X} - \mathbf{t_i p_i^T} = \mathbf{X} - \mathbf{t_i t_i^T X} = (\mathbf{I} - \mathbf{t_i t_i^T})\mathbf{X}$
7. $\mathbf{y} = \mathbf{y} - \mathbf{t_i}\hat{\beta}_\mathrm{i} = \mathbf{y} - \mathbf{t_i t_i^T y} = (\mathbf{I} - \mathbf{t_i t_i^T})\mathbf{y}$
8. i = i + 1, Repeat Step 2 - 7.

To estimate the response given the new observations $\mathbf{X_{new}}$, one may use the following steps.

1. Start from i = 1. $\mathbf{\hat{y}_{new}} = 0$
2. $\mathbf{t_{new(i)}} = \mathbf{X_{new} w_i} / \|\mathbf{X_{new} w_i}\|$
3. $\mathbf{\hat{y}_{new}} = \mathbf{\hat{y}_{new}} + \mathbf{t_{new(i)}}\hat{\beta}_\mathrm{i}$
4. $\mathbf{X_{new}} = \mathbf{X_{new}} - \mathbf{t_{new(i)} p_i^T} = \mathbf{X_{new}} - \mathbf{t_{new(i)} t_i^T X}$
5. Repeat Step 2 - 4 until all k components are used.

6.3 Univariate Kernel Partial Least Squares Regression

To capture the nonlinear relationship between $\mathbf{X}$ and $\mathbf{y}$, one can replace the linear regression procedure between $\mathbf{t_i}$ and $\mathbf{y_i}$ by a nonlinear modeling technique. Wold (1989) proposed a

quadratic PLS modeling method to extend the PLS method to the data with relative simpler nonlinearity. To capture more complicate nonlinear relationships, Qin and McAvoy (1992) proposed a neural network PLS method, in which a series (one for each component) of single-input-single-output(SISO) feedforward neural networks are employed to model the nonlinear pattern. However, designing the neural network structure is a complicate procedure and the neural network training could be time consuming. Rosipal (2001) proposed the kernel PLS (KPLS) regression, in which the linear PLS regression is conducted between the dependent variable $\mathbf{y}$ and $\Phi(\mathbf{X})$, the nonlinear mapping of the independent variables $\mathbf{X}$ on the feature space. Lewi (1995) proposed a modified NIPALS algorithm using the crossproduct matrix $\mathbf{XX^T}$. KPLS is based on this crossproduct version of PLS since the explicit form of $\Phi(\mathbf{X})$ is unknown but $\Phi(\mathbf{X})\Phi(\mathbf{X})^T = \mathbf{K}$ can be obtained. The detailed procedure of Lewi's modified NIPALS is shown below (Note: $\mathbf{y_i}$ must be saved for the prediction purpose).

1. Start from i = 1. $\mathbf{y_i} = \mathbf{y}$.
2. $\mathbf{w_i} = \mathbf{XX^T y_i}$
3. $\mathbf{t_i} = \mathbf{w_i}/\|\mathbf{w_i}\|$
4. $\mathbf{p_i} = \mathbf{X^T t_i}$
5. $\hat{\beta}_\mathrm{i} = \mathbf{t_i^T y_i}$
6. Deflate $\mathbf{XX^T}$

$$\begin{aligned}
\mathbf{XX^T} &= (\mathbf{X} - \mathbf{t_i p_i^T})(\mathbf{X} - \mathbf{t_i p_i^T})^\mathbf{T} \\
&= (\mathbf{X} - \mathbf{t_i t_i^T X})(\mathbf{X} - \mathbf{t_i t_i^T X})^\mathbf{T} \\
&= (\mathbf{I} - \mathbf{t_i t_i^T})\mathbf{XX^T}(\mathbf{I} - \mathbf{t_i t_i^T}) \\
&= \mathbf{XX^T} - \mathbf{XX^T t_i t_i^T} - \mathbf{t_i t_i^T XX^T} + \mathbf{t_i t_i^T XX^T t_i t_i^T}
\end{aligned}$$

7. $\mathbf{y_{i+1}} = \mathbf{y_i} - \mathbf{t_i}\hat{\beta}_\mathrm{i} = \mathbf{y_i} - \mathbf{t_i t_i^T y_i} = (\mathbf{I} - \mathbf{t_i t_i^T})\mathbf{y_i}$

8. i = i + 1, Repeat Step 2 - 7 until k components are extracted.

The following steps are for predicting the response given the new observations $\mathbf{X_{new}}$.

1. Start from i = 1. $\mathbf{\hat{y}_{new}} = \mathbf{0}$

2. $\mathbf{w_{new(i)}} = \mathbf{X_{new}X^T y_i}$

3. $\mathbf{t_{new(i)}} = \mathbf{w_{new(i)}}/\|\mathbf{w_{new(i)}}\|$

4. $\mathbf{\hat{y}_{new}} = \mathbf{\hat{y}_{new}} + \mathbf{t_{new(i)}}\hat{\beta}_{\mathrm{i}}$

5. Deflate $\mathbf{X_{new}X^T}$

$$\begin{aligned}
\mathbf{X_{new}X^T} &= (\mathbf{X_{new}} - \mathbf{t_{new(i)}p_i^T})(\mathbf{X} - \mathbf{t_{new(i)}p_i^T})^\mathbf{T} \\
&= (\mathbf{X_{new}} - \mathbf{t_{new(i)}t_i^T X})(\mathbf{X} - \mathbf{t_{new(i)}t_i^T X})^\mathbf{T} \\
&= \mathbf{X_{new}X^T} - \mathbf{X_{new}X^T t_i t_{new(i)}^T} - \mathbf{t_{new(i)}t_i^T XX^T} \\
&\quad + \mathbf{t_{new(i)}t_i^T XX^T t_i t_{new(i)}^T}
\end{aligned}$$

To extend the above results to the kernel space, simply replace $\mathbf{X}$ with $\Phi(\mathbf{X})$ and apply the kernel trick as shown below.

1. Start from i = 1, $\mathbf{y_i} = \mathbf{y}$

2. $\mathbf{w_i} = \mathbf{\Phi(X)\Phi(X)^T y_i} = \mathbf{K y_i}$.

3. $\mathbf{t_i} = \mathbf{w_i}/\|\mathbf{w_i}\|$

4. $\mathbf{p_i} = \mathbf{\Phi(X)^T t_i}$

5. $\hat{\beta}_{\mathrm{i}} = \mathbf{t_i^T y_i}$

6. Deflate $\mathbf{\Phi(X)\Phi(X)^T}$

$$
\begin{aligned}
\mathbf{\Phi(X)\Phi(X)^T} &= \left[\mathbf{\Phi(X)} - \mathbf{t_i t_i^T \Phi(X)}\right]\left[\mathbf{\Phi(X)} - \mathbf{t_i t_i^T \Phi(X)}\right]^{\mathbf{T}} \\
&= (\mathbf{I} - \mathbf{t_i t_i^T})\mathbf{\Phi(X)\Phi(X)^T}(\mathbf{I} - \mathbf{t_i t_i^T}) \\
\Rightarrow \mathbf{K} &= (\mathbf{I} - \mathbf{t_i t_i^T})\mathbf{K}(\mathbf{I} - \mathbf{t_i t_i^T})
\end{aligned}
$$

7. $\mathbf{y_{i+1}} = \mathbf{y_i} - \mathbf{t_i t_i^T y_i} = (\mathbf{I} - \mathbf{t_i t_i^T})\mathbf{y_i}$

8. i = i + 1, Repeat Step 2 - 7 until k components are extracted.

where $\mathbf{K} = \mathbf{\Phi(X)\Phi(X)^T}$ is the kernel matrix defined in Chapter 2.

To estimate the response given the new observations $\mathbf{X_{new}}$ one may use the following steps.

1. Start from i = 1.

   $\mathbf{\hat{y}_{new}} = \mathbf{0}$

2. $\mathbf{w_{new(i)}} = \mathbf{\Phi(X)_{new}\Phi(X)^T y_i} = \mathbf{K_{new} y_i}$

3. $\mathbf{t_{new(i)}} = \mathbf{w_{new(i)}}/\|\mathbf{w_{new(i)}}\|$

4. $\mathbf{\hat{y}_{new}} = \mathbf{\hat{y}_{new}} + \mathbf{t_{new(i)}}\hat{\beta}_{\mathrm{i}}$

5. Deflate $\mathbf{\Phi(X)_{new}\Phi(X)^T}$.

$$
\begin{aligned}
& \mathbf{\Phi(X)_{new}\Phi(X)^T} = \mathbf{K_{new}} \\
= \;& \left[\mathbf{\Phi(X)_{new}} - \mathbf{t_{new(i)} p_i^T}\right]\left[\mathbf{\Phi(X)} - \mathbf{t_{new(i)} p_i^T}\right]^{\mathbf{T}} \\
= \;& \left[\mathbf{\Phi(X)_{new}} - \mathbf{t_{new(i)} t_i^T \Phi(X)}\right]\left[\mathbf{\Phi(X)} - \mathbf{t_{new(i)} t_i^T \Phi(X)}\right]^{\mathbf{T}} \\
= \;& \mathbf{\Phi(X)_{new}\Phi(X)^T} - \mathbf{\Phi(X)_{new}\Phi(X)^T t_i t_{new(i)}^T} \\
& - \mathbf{t_{new(i)} t_i^T \Phi(X)\Phi(X)^T} + \mathbf{t_{new(i)} t_i^T \Phi(X)\Phi(X)^T t_i t_{new(i)}^T} \\
= \;& \mathbf{K_{new}} - \mathbf{K_{new} t_i t_{new(i)}^T} - \mathbf{t_{new(i)} t_i^T K} + \mathbf{t_{new(i)} t_i^T K t_i t_{new(i)}^T} \\
= \;& (\mathbf{K_{new}} - \mathbf{t_{new(i)} t_i^T K})(\mathbf{I} - \mathbf{t_i t_{new(i)}^T})
\end{aligned}
$$

6. Repeat Step 2 - 5 until all k components are used.

So far, we assume the KPLS regression is based on the original un-centered $\mathbf{\Phi}(\mathbf{X})$. Centering of $\mathbf{\Phi}(\mathbf{X})$ is not directly available since the explicit form of $\mathbf{\Phi}(\mathbf{X})$ is unknown. However, the modeling procedure requires the kernel matrix $\mathbf{K}$ or/and $\mathbf{K_{new}}$ only. As discussed in Chapter 2, one may use $\mathbf{K_c}$ (Equation 2.17) and $\mathbf{K_{new(c)}}$(Equation 2.18) to replace $\mathbf{K}$ and $\mathbf{K_{new}}$ respectively if the mean-centered $\mathbf{\Phi}(\mathbf{X})$ is used.

The PLS regression model can be expressed in a matrix form (Manne, 1987). Similarly, the KPLS regression can be expressed in a matrix form (Rosipal and Trejo, 2001) as follows. For a multivariate regression model in the feature space defined as:

$$\mathbf{Y} = \mathbf{\Phi}(\mathbf{X})\mathbf{B} + \mathbf{E}, \tag{6.4}$$

where $\mathbf{E}$ is the multivariate i.i.d. random noise. It's KPLS estimator of the regression coefficients $\mathbf{B}$ is given by

$$\hat{\mathbf{B}} = \mathbf{\Phi}(\mathbf{X})^{\mathbf{T}}\mathbf{U}(\mathbf{T}^{\mathbf{T}}\mathbf{K}\mathbf{U})^{-\mathbf{1}}\mathbf{T}^{\mathbf{T}}\mathbf{Y}. \tag{6.5}$$

To make prediction on the training data, there is

$$\widehat{\mathbf{Y}} = \mathbf{\Phi}(\mathbf{X})\widehat{\mathbf{B}} = \mathbf{K}\mathbf{U}(\mathbf{T}^{\mathbf{T}}\mathbf{K}\mathbf{U})^{-\mathbf{1}}\mathbf{T}^{\mathbf{T}}\mathbf{Y} = \mathbf{T}\mathbf{T}^{\mathbf{T}}\mathbf{Y}, \tag{6.6}$$

where $\mathbf{T} = \mathbf{K}\mathbf{U}(\mathbf{T}^{\mathbf{T}}\mathbf{K}\mathbf{U})^{-\mathbf{1}}\mathbf{T}$ (de Jong, 1993). It can be shown (Höskuldsson, 1988) that $\mathbf{t_i'Y_i}$ is the same as $\mathbf{t_i'Y}$. Therefore, the last equality can be written as:

$$\widehat{\mathbf{Y}} = \mathbf{T}\mathbf{T}^{\mathbf{T}}\mathbf{Y_i} = \mathbf{T}\widehat{\boldsymbol{\beta}}, \tag{6.7}$$

where $\mathbf{Y_i}$ stands for the deflated $\mathbf{Y}$ after extracting the first $i-1$ LVs and $\widehat{\boldsymbol{\beta}}$ contains the coefficients of regression $\mathbf{Y_i}$ on $\mathbf{t_i}$, the project on the i^{th} LV. This is the underlying regression

part of KPLS. It will be used to define the covariance matrix of the estimated parameters in the next section.

The predicted new observations is defined by

$$\widehat{\mathbf{Y}}_{\mathbf{new}} = \mathbf{\Phi}(\mathbf{X}_{\mathbf{new}})\widehat{\mathbf{B}} = \mathbf{K}_{\mathbf{new}}\mathbf{U}(\mathbf{T}^{\mathbf{T}}\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^{\mathbf{T}}\mathbf{Y} = \mathbf{T}_{\mathbf{new}}\widehat{\boldsymbol{\beta}}, \tag{6.8}$$

where $\mathbf{K}_{\mathbf{new}} = \mathbf{\Phi}(\mathbf{X}_{\mathbf{new}})\mathbf{\Phi}(\mathbf{X})^{\mathbf{T}}$. We have assumed that the data in the feature space is centered. Centering $\mathbf{\Phi}(\mathbf{X})$ and $\mathbf{\Phi}(\mathbf{X}_{\mathbf{new}})$ in the feature space is conducted through the kernel function using (2.17) and (2.18).

6.4 ICOMP for KPLSR

In KPLSR, the good generalization properties can be achieved by the appropriate selection of

1. the form of kernel function and the corresponding parameters,
2. the number of retained latent vectors,
3. and the subset of the independent variables.

These model selection problems are still open to researchers. Currently the computationally intensive cross-validation is the widely used method for comparing the kernel functions (Rosipal and Trejo, 2001) and selecting the subset variables (Han et al., 2006; Mehdi and Kyani, 2007). In this chapter, we use the information complexity (ICOMP) measure technique to choose kernel functions and subset models.

We consider the univariate KPLSR model where the estimate of the response is given by:

$$\widehat{\mathbf{y}} = \mathbf{T}\mathbf{T}^{\mathbf{T}}\mathbf{y} = \mathbf{T}\hat{\mathbf{b}}. \tag{6.9}$$

Assuming the random noise follows an i.i.d. normal distribution $N(0, \sigma^2)$, the general ICOMP form for UKPLS can be defined by

$$
\begin{aligned}
ICOMP(Cov(\hat{\mathbf{b}},\hat{\sigma}^2))_{UKPLS} &= -2\log L(\hat{\mathbf{b}},\hat{\sigma}^2)) + 2C(Cov(\hat{\mathbf{b}},\hat{\sigma}^2)) \\
&= n\log(2\pi) + n\log(\hat{\sigma}^2) + n + 2C(Cov(\hat{\mathbf{b}},\hat{\sigma}^2)), \quad (6.10)
\end{aligned}
$$

where $C(\cdot)$ can be either $C_1(\cdot)$ defined in (3.9) or the quadratic measure $C_{1F}(\cdot)$ defined in (3.32). Assuming, $\hat{\mathbf{b}}$ is uncorrelated with $\hat{\sigma}$, the covariance matrix of the estimated parameters is defined as:

$$
Cov(\hat{\mathbf{b}},\hat{\sigma}^2) = \begin{bmatrix} Cov(\hat{\mathbf{b}}) & \mathbf{0}_{k\times 1} \\ \mathbf{0}_{1\times k} & \frac{2\hat{\sigma}^4}{n} \end{bmatrix} \quad (6.11)
$$

where k is the number of retained components and $Cov(\hat{\mathbf{b}})$ is a diagonal matrix whose i^{th} diagonal element is the variance of the deflated $\mathbf{y}$ (or the residuals) when the first i latent vectors have been extracted.

The PEU version of ICOMP for UKPLSR can be defined as

$$
\begin{aligned}
ICOMPPEU_{UKPLS} &= -2\log L(\hat{\mathbf{b}},\hat{\sigma}^2) + k + 2C(Cov(\hat{b},\hat{\sigma}^2)) \quad (6.12) \\
&= n\log(2\pi) + n\log(\hat{\sigma}^2) + n + k + 2C(Cov(\hat{\mathbf{b}},\hat{\sigma}^2)) \quad (6.13)
\end{aligned}
$$

where k is added as the extra penalty to the model complexity. Inspired by SBC (Schwarz, 1978), the constant "2" of the complexity term can be replaced by $\log(n)$, which leads to the modified ICOMPPEU:

$$
\begin{aligned}
ICOMPPEU^{*}_{UKPLS} &= n\log(2\pi) + n\log(\hat{\sigma}^2) + n \quad (6.14) \\
&\quad + k + \log(n)C(Cov(\hat{\mathbf{b}},\hat{\sigma}^2)) \quad (6.15)
\end{aligned}
$$

For the fixed kernel function and parameters, when the number of LVs increases, LOF decreases and the model complexity increases. The penalty terms of ICOMP measure the complexity increasing.

We further propose a regularization method to give a more reasonable estimate of σ^2. In the linear regression model, we regularize the covariance matrix $\mathbf{X'X}$. In the ridge regression, we resolve the singularity by adding the bias, that is, $\mathbf{X'X} + \lambda\mathbf{I}$. In KRR, the kernel matrix $\mathbf{K}$ is playing the role of the covariance matrix. The training data response is predicted as:

$$\hat{\mathbf{y}} = \mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y} \tag{6.16}$$

Inspired by this idea, we want to perform the regularization to KPLSR in the same manner. However, we could not find such symmetrical covariance in the KPLSR model since the components are sequentially extracted. By comparing (6.6) with (6.16), we found $\mathbf{M} = \mathbf{U}(\mathbf{T'KU})^{-1}\mathbf{T'}$ is the counter part that plays role of $(\mathbf{K} + \lambda\mathbf{I})^{-1}$ though the former one is not symmetrical. Therefore, we decided to regularize $\mathbf{M}$ for KPLSR. As discussed in Chapter 5, there are many different methods to find the appropriate ridge parameter λ. However, there is no method that always outperforms the other. This research utilizes the maximum likelihood/empirical Bayes (MLE/EB) covariance matrix estimator (Bozdogan, 2007):

$$\lambda = \frac{k-1}{n \cdot trace(\widehat{\mathbf{\Sigma}})}, \tag{6.17}$$

where k is the number of estimated parameters and n is the number of training observations. In KPLSR, the above estimator is modified to

$$\lambda = \frac{n-1}{n \cdot trace(\widehat{\boldsymbol{M}})}, \tag{6.18}$$

where n is the number of observations, that is, the dimension of $\mathbf{M}$. When the scale parameter is small, the magnitude of $Trace(\mathbf{M})$ is low, which leads to big regularization. When the scale parameter is high, the magnitude of $Trace(\mathbf{M})$ is high and the effect of the

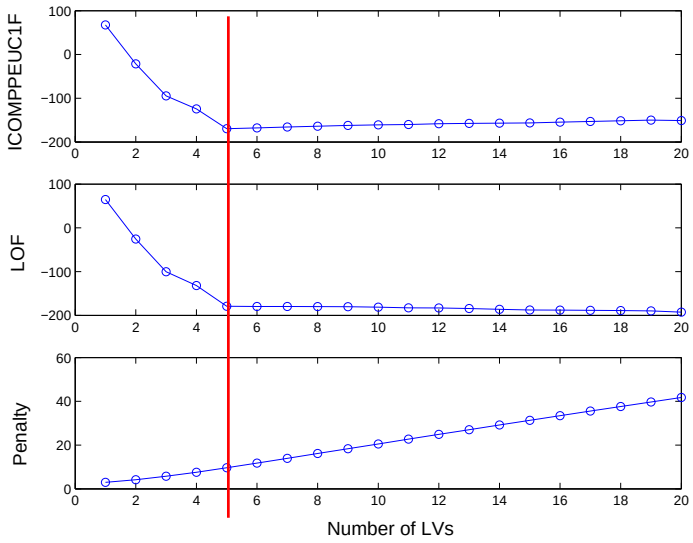


Figure 6.1: Selecting Number of LVs Using ICOMPPEUC1F

regularization is negligible. The regularized matrix $\mathbf{M}$ is given by

$$\mathbf{M_r} = \mathbf{M} + \lambda \mathbf{I}, \tag{6.19}$$

where λ is given by (6.18).

6.5 Numerical Results

6.5.1 Sinc Function

We first demonstrate the results using the popular sinc function data (Appendix A1), which contains one independent variable and one response.

First, the scale parameter of the Gaussian RBF kernel is fixed and different criteria are used to select the number of LVs. We use $\gamma = 4$ in the light of some preliminary trials. Using the PEU form of ICOMPC1F, it can be observed that LOF keeps decreasing as the number of LVs increases while the penalty $(2C_{1F} + k)$ increases as the number of LVs increases (Figure 6.1). The balanced choice is using 5 LVs, which minimizes ICOMP.

Table 6.1: KPLS: Selecting the Number of LVs for KPLSR

# LVs	MSEtest	HO	AIC	SBC	ICOMPC1F	ICOMPPEUC1F
1	0.11026	0.11058	73.4	78.6	65.6	67.6
2	0.04633	0.04582	-13.1	-5.3	-24.5	-21.5
3	0.02265	0.02331	-84.0	-73.6	-98.5	-94.5
4	0.01736	0.01838	-112.5	-99.5	-129.3	-124.3
5	0.01082	0.01035	-156.5	-140.9	-175.7	-169.7
6	0.01084	0.01036	-154.3	-136.1	-174.8	-167.8
7	0.01077	0.01024	-152.1	-131.2	-173.8	-165.8
8	0.01092	0.01029	-149.9	-126.5	-172.9	-163.9
9	0.01102	0.01041	-147.9	-121.8	-172.0	-162.0
10	0.01110	0.01058	-146.5	-117.9	-171.8	-160.8
11	0.01122	0.01082	-145.6	-114.3	-172.0	-160.0
12	0.01127	0.01086	-143.6	-109.8	-171.1	-158.1
13	0.01148	0.01102	-142.8	-106.3	-171.4	-157.4
14	0.01163	0.01111	-142.3	-103.2	-172.0	-157.0
15	0.01172	0.01128	-141.5	-99.8	-172.3	-156.3
16	0.01176	0.01131	-139.7	-95.4	-171.6	-154.6
17	0.01177	0.01135	-138.0	-91.1	-171.0	-153.0
18	0.01180	0.01144	-136.3	-86.8	-170.5	-151.5
19	0.01182	0.01150	-134.7	-82.6	-170.0	-150.0
20	0.01190	0.01178	-135.5	-80.8	-171.9	-150.9

We compared five criteria including hold-out validation, AIC, SBC, ICOMPC1F and ICOMPPEUC1F in this example (Table 6.1). The criteria based on information measure agree that using 5 LVs is the optimal choice. The hold-out validation prefers 7 LVs. However, the testing data MSEs are similar.

To test the robustness of the model selection criteria, we repeat the simulation 100 times. Four different criteria are compared in terms of the average and the variation of the testing data MSE. Since 1000 hold-out observations almost represent the whole population, it is not surprised that the hold-out cross-validation is doing the best overall (Figure 6.2). However, without using the additional data, ICOMPPEUC1F, AIC and SBC are all giving similar predictive performance (Table 6.2) while ICOMPPEUC1F is slightly better compared with AIC and SBC. This is a very exciting result.

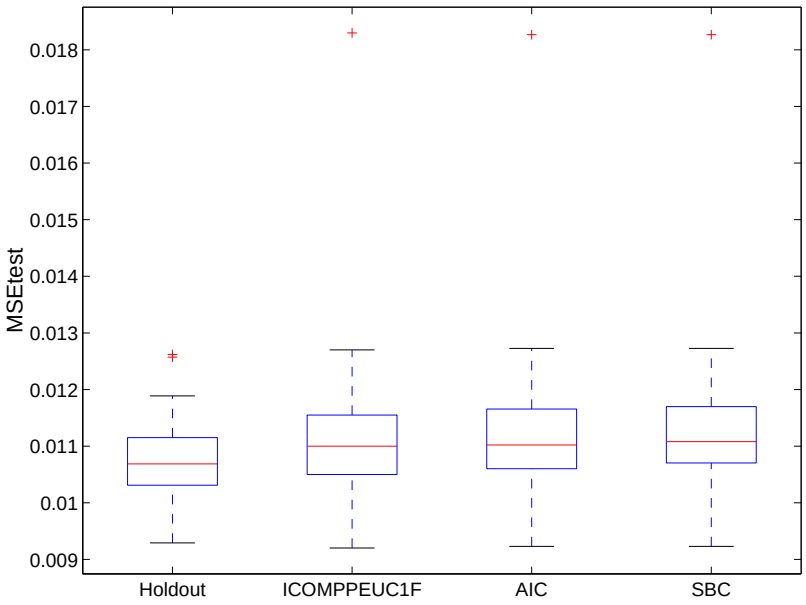


Figure 6.2: Sinc: Comparing Model Selection Criteria in 100 simulations

Table 6.2: Sinc: Comparing Model Selection Criteria

Criterion	Ave. MSEtest	Std. MSEtest	# LVs
ICOMPC1F	0.0115	0.00092	10.5
ICOMPPEUC1F	0.0111	0.00074	5.7
Hold-out	0.0107	0.00064	5.7
AIC	0.0112	0.00069	6.3
SBC	0.0112	0.00073	5.0

Table 6.3: Friedman: Comparing Model Selection Criteria

Criterion	Ave. MSEtest	Std. MSEtest	Ave. #LVs
ICOMPPEUC1F	3.96	0.37	22
Hold-out	3.58	0.29	17
AIC	4.30	0.43	40
SBC	4.29	0.43	39

6.5.2 Friedman's Data

We now apply ICOMP to the multiple regression applications and search for the optimal subset variables as well as the kernel function and the number of RVs. The Friedman's data (Appendix A2) is used for the demonstration in this experiment. We utilize the widely used Gaussian RBF kernel. First, all the 10 regressors are included (saturated model). ICOMP is utilized to choose the optimal scale parameter of the Gaussian RBF kernel function and the number of LVs retained. Then, we compare all the $2^{10} - 1 = 1023$ subset models (including at least one regressor) in terms of the ICOMP score given the selected optimal kernel. The optimal subset model is the minimizer of ICOMP. We hoped the selected model excludes $x_6 - x_{10}$.

We compare different criteria to choose the scale parameter including ICOMPC1F, ICOMPPEUC1F, Hold-out method, AIC and SBC. It is concluded (Table 6.3) that AIC and SBC tends to under-penalize the number of LVs. The results of ICOMPPEUC1F are closest to the results of the hold-out method. Combining the results of the sinc data and the Friedman's data, we decide to use ICOMPPEUC1F as the criterion to choose the optimal subset model.

We now perform the subset selection given the selected Gaussian RBF kernel with the scale parmaeter $\gamma = 5$. ICOMPC1FPEU is used as the model selection criterion. There are only $2^{10} - 1 = 1023$ subsets for this data. One may conduct the all-possible-subset-selection (APSS). Our APSS results show that the best model is $\{x_1, x_2, x_3, x_4, x_5\}$ which is the expected true model. The testing data MSE of this model is 1.626. It is a tremendous improvement compared with 3.96 of the saturated model. If the number of variables are

Table 6.4: GA Parameters for Friedman Data

n_{gen}	50
n_{pop}	50
P_{cross}	0.7
$P_{mutation}$	0.01
Elitism	YES
Type of crossover	Uniform

Table 6.5: Summary of GA for Friedman Data KPLS

Generation	Best Subset	ICOMP	# LVs	MSEtest
1-4	1 2 3 4 5 7 8	829.08	23	2.239
5-8	1 2 3 4 5 8	802.34	21	1.991
9-41	1 2 3 4 5 7	796.54	21	1.871
42-100	1 2 3 4 5	780.41	19	1.626

large, it is more efficient to use GA to find the optimal model. We use Friedman's data to demonstrate the GA procedure though it is not really necessary here.

The parameter of GA are configured as shown in Table 6.4. In a single run of the above GA procedure, the best subset of the initial population is $\{x_1, x_2, x_3, x_4, x_5, x_7, x_8\}$ (Table 6.5). The true subset is found in the 42^{th} generation.

To confirm the robustness of the GA procedure. We run the same GA 100 times with random initial population for each run. It is concluded that the true model is selected 99 times. The missing one selected a "good" model $\{x_1, x_2, x_3, x_4, x_5, x_7\}$, which is also acceptable. One may argue that using popsize = 50 and generation = 50 are not efficient for 10 variables. Our experiments show that using the appropriate population size and number of generations are required for this 1023-subset case to find the best model. However, when the number of subsets is huge, for instance, in millions, it is not necessary to further increase the generations and population size for searching the optimal model, which means that using GA is efficient.

Chapter 7

Kernel PCA/PCR

7.1 Introduction

Principal component analysis (PCA) is a widely used statistical data preprocessing technique for dimensionality reduction and denoising. PCA is also called the (discrete) Karhunen-Love transform (KLT), named after Kari Karhunen and Michel Love, or the Hotelling (1933) transform, in honor of Harold Hotelling. It has been applied in various areas including but not limited to applications in image processing, agriculture, biology, chemistry, climatology, demography, genetics, psychology and industrial process control. The origin of PCA is difficult to trace. Interested reader may find a brief historical review of PCA in Jolliffe's book (2002).

The geometric idea of principal component analysis (PCA) is to project a data set with a large number of interrelated variables to a set of orthogonal axes such that the projected variables, called principal components (PCs) are uncorrelated. The variances of the resulting PCs are in descending order such that the first principal component (PC) has the highest variance among all the PCs.

Given a p-variable data matrix $\mathbf{X} \in \mathbf{R}^p$, the first principal component is found such that the projection of $\mathbf{X}$ in its direction has the maximum variance. The next principal component must be on a direction that is orthogonal to the first PC. Among these possible

orthogonal directions, the second PC is found such that the projection of $\mathbf{X}$ in its direction has the maximum variance. Based on the same logic, the k^{th} PC must be orthogonal to all the previous PCs and in a direction that the variance of the projected $\mathbf{X}$ is maximized. The maximum number of PCs can be found is equal to the rank of $\mathbf{X}$. Dimensionality reduction and signal denoising can be achieved by retaining only a subset of all PCs, which is believed to be crucial to represent the most information needed for a statistical model. The traditional criteria are majorally focused on the variances of PCs. PCs with relative small variances are assumed to contain nuisance information and need to be dropped. However, in some circumstances, the highest-variance PC is not equivalent as the most useful PC. Jolliffe (2002) gives a typical illustration of such situation in the application of discriminant analysis. The orthogonal property of PCs can be used in regression models to treat collinearity problems when the nuisance PCs are truncated.

7.2 Methodology of Linear PCA

Let an $n \times p$ matrix $\mathbf{X} \in \mathbf{R}^p$ stands for a p-variate data set with n observations. Generally, $\mathbf{X}$ needs to be centered such that each variable (column) has a mean of zero. Otherwise, the first PC represents the average of each variable instead of the the variability information. We assume $\mathbf{X}$ is zero-centered in the following part of this chapter.

The principal component scores, the projections of $\mathbf{X}$ on the direction of the PCs, can be found by multiplying the centered data matrix $\mathbf{X}$ by a $q \times q$ orthonormal matrix $\mathbf{A}$, where q is the rank of $\mathbf{X}$. It can be expressed in a matrix form as shown below:

$$\mathbf{Z} = \mathbf{XA}, \tag{7.1}$$

where the i^{th} column of $\mathbf{Z}$ is the projection of $\mathbf{X}$ on the i^{th} PC, $\mathbf{A}$ is the orthonormal matrix whose i^{th} columns is the i^{th} normalized eigenvector of $\mathbf{X}^T\mathbf{X}/(n-1)$, the sample covariance matrix.

7.3 Kernel Principal Component Analysis

In kernel PCA, the original p-variable data $\mathbf{X} \in \mathbb{R}^p$ will be nonlinearly transformed to $\Phi(\mathbf{X})$ in a feature space $\mathcal{F}$. This is to be done by the kernel trick. Then, the regular linear PCA is conducted on the transformed data $\Phi(\mathbf{X})$ in the feature space. The kernel principal components are to be found by

$$\mathbf{Z} = \Phi(\mathbf{X})\mathbf{A}, \tag{7.2}$$

where the $n \times p$ matrix $\mathbf{Z}$ contains kernel PCs in columns, the i^{th} column of $\mathbf{A}$ is the normalized eigenvector corresponding to the i^{th} largest eigenvalues of $\mathbf{\Sigma}$, the covariance matrix of $\Phi(\mathbf{X})$. When $\mathbf{\Sigma}$ is unknown, it can be estimated by the sample covariance matrix defined as:

$$\mathbf{S} = \frac{1}{n-1}\Phi(\mathbf{X})^T\Phi(\mathbf{X}). \tag{7.3}$$

However, in the kernel methods, the dimension and the explicit expression of $\Phi(\mathbf{X})$ are unknown. Fortunately, KPCA can be performed through the kernel matrix $\mathbf{K} = \Phi(\mathbf{X})\Phi(\mathbf{X})^T$. Let $\mathbf{v}_i$ denote the i^{th} normalized eigenvector of $\mathbf{S}$ defined in Equation 7.3 and λ_i denote the corresponding i^{th} largest eigenvalue. There is

$$\mathbf{S}\mathbf{v}_i = \lambda_i\mathbf{v}_i,$$

i.e.,

$$\frac{1}{n-1}\Phi(\mathbf{X})^T\Phi(\mathbf{X})\mathbf{v}_i = \lambda_i\mathbf{v}_i. \tag{7.4}$$

Multiplying on the left of both sides by $\Phi(\mathbf{X})$, we have:

$$\begin{aligned} \frac{1}{n-1}\Phi(\mathbf{X})\Phi(\mathbf{X})^T\Phi(\mathbf{X})\mathbf{v}_i &= \lambda_i\Phi(\mathbf{X})\mathbf{v}_i \\ \frac{1}{n-1}\mathbf{K}\Phi(\mathbf{X})\mathbf{v}_i &= \lambda_i\Phi(\mathbf{X})\mathbf{v}_i. \end{aligned}$$

Let $\Phi(\mathbf{X})\mathbf{v}_i = \mathbf{u}_i^*$, we see $\mathbf{u}_i^*$ is simply the eigenvector of $\mathbf{K}/(n-1)$ corresponding to the eigenvalue λ_i. The normalized eigenvector $\mathbf{u}_i$ can be calculated by:

$$\mathbf{u}_i = \frac{\mathbf{u}_i^*}{\|\mathbf{u}_i^*\|} = \frac{\mathbf{u}_i^*}{\sqrt{\mathbf{u}_i^{*T}\mathbf{u}_i^*}} = \frac{\mathbf{u}_i^*}{\sqrt{\mathbf{v}_i^T\Phi(\mathbf{X})^T\Phi(\mathbf{X})\mathbf{v}_i}}$$

Multiplying on the left of both sides of Equation 7.4 by $\mathbf{v}_i^T$, there is:

$$\frac{1}{n-1}\mathbf{v}_i^T\Phi(\mathbf{X})^T\Phi(\mathbf{X})\mathbf{v}_i = \mathbf{v}_i^T\lambda_i\mathbf{v}_i = \lambda_i \tag{7.5}$$

Therefore, there is

$$\mathbf{u}_i = \frac{\mathbf{u}_i^*}{\sqrt{(n-1)\lambda_i}}. \tag{7.6}$$

Let $\mathbf{U}$ denote the normalized eigenvector matrix whose i^{th} column is $\mathbf{u}_i$, $\mathbf{U}^*$ denote the eigenvector matrix whose i^{th} column is $\mathbf{u}_i^*$ and $\mathbf{\Lambda}$ denote the diagonal matrix whose i^{th} diagonal element is $(n-1)\lambda_i$, the i^{th} largest eigenvalue of $\mathbf{K}$, it can be written that

$$\mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}} = \mathbf{U}^* = \Phi(\mathbf{X})\mathbf{A}, \tag{7.7}$$

where $\mathbf{A}$ was defined in Equation 7.2 and the i^{th} column of $\mathbf{A}$ is $\mathbf{v}_i$. It can be written that

$$\mathbf{Z} = \Phi(\mathbf{X})\mathbf{A} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}. \tag{7.8}$$

The variance of each kernel PC is equal to the corresponding eigenvalue of $\mathbf{K}/(n-1)$ or $1/(n-1)$ eigenvalue of $\mathbf{K}$. Using the fact that the positive eigenvalues of $\Phi(\mathbf{X})^T\Phi(\mathbf{X})$ and $\Phi(\mathbf{X})\Phi(\mathbf{X})^T$ are the same and the corresponding eigenvector matrices are both $\mathbf{A}$, there is

$$\Phi(\mathbf{X})^T\Phi(\mathbf{X})\mathbf{A} = \mathbf{A}\mathbf{\Lambda}.$$

Move $\mathbf{\Lambda}$ to the other side, there is

$$\mathbf{A} = \Phi(\mathbf{X})^T\Phi(\mathbf{X})\mathbf{A}\mathbf{\Lambda}^{-1} = \Phi(\mathbf{X})^T\mathbf{U}\Lambda^{-\frac{1}{2}}.$$

Using the above result, kernel PCs can be found by

$$\mathbf{Z} = \Phi(\mathbf{X})\mathbf{A} = \Phi(\mathbf{X})\Phi(\mathbf{X})^T\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{K}\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}. \tag{7.9}$$

This alternative expression using kernel matrix is very useful for finding the projection of the testing data on the axes of PCs. Given the test data or new data $\mathbf{X}_{\text{new}}$ with m observations, the projection of m points on the axes of kernel PCs, defined by the training data $\mathbf{X}$, can be found using

$$\mathbf{Z}_{\text{new}} = \Phi(\mathbf{X}_{\text{new}})\mathbf{A} = \Phi(\mathbf{X}_{\text{new}})\Phi(\mathbf{X})^T\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}} = \mathbf{K}_{\text{new}}\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}, \tag{7.10}$$

where

$$\mathbf{K}_{\text{new}} = \Phi(\mathbf{X}_{\text{new}})\Phi(\mathbf{X})^T. \tag{7.11}$$

7.4 Reconstruction of Kernel PCA

7.4.1 Reconstruction of Linear PCA

In linear PCA, the original data matrix is decomposed as:

$$\mathbf{X} = \sum_{j=1}^{p} \mathbf{z_j}\mathbf{a_j^T}, \tag{7.12}$$

where $\mathbf{z_j}$ and $\mathbf{a_j}$ are the j^{th} PC score (the projection of $\mathbf{X}$ on the j^{th} principal component) and eigenvector respectively. In (7.12), $\mathbf{X}$ is fully reconstructed by using all the PCs. $\mathbf{X}$ is generally reconstructed using a subsets of principal components for dimension reduction, de-noising or missing value imputation. The retained principle components are assumed to keep the important signal information such that the reconstructed $\mathbf{X}$ has little difference from the real $\mathbf{X}$. We assume the difference is due to the random error, which is contained in the ignored PCs. Since the variances of the principal components are in a descending order, the subset selection problem is simplified to a problem of selecting number of PCs.

That is, we assume the statistical model defined by

$$\mathbf{X} = \mathbf{X_r} + \text{Random Error} = \sum_{j=1}^{k} \mathbf{z_j a_j^T} + \sum_{j=k+1}^{p} \mathbf{z_j a_j^T}, \tag{7.13}$$

where k is the number of the retained PCs. The methods of choosing k will be introduced in the next section.

7.4.2 Difficulties in Reconstructing Kernel PCA

In KPCA, the reconstructed $\Phi(\mathbf{X})$ does not have the explicit form as one can see from

$$\begin{aligned} \Phi(\mathbf{X})_r &= \mathbf{Z_k A_k^T} \\ &= \mathbf{K U_k \Lambda_k}^{-\frac{1}{2}} \left[\Phi(\mathbf{X})^T \mathbf{U_k \Lambda_k}^{-\frac{1}{2}}\right]^T \\ &= \mathbf{K U_k \Lambda_k}^{-1} \mathbf{U_k^T} \Phi(\mathbf{X}), \end{aligned} \tag{7.14}$$

where the subscript k stands for retaining the first k PCs. Actually, the reconstruction of $\Phi(\mathbf{X})$ in the feature space is not a researcher's direct interest since $\mathbf{X}$ in the original space is the data that the researcher tries to reconstruct. Such reconstructed $\mathbf{X}$ is generally called pre-image in the image processing applications.

Let $\Phi(\mathbf{x})_r$ stand for the reconstruction of the observation $\Phi(\mathbf{x})$ in the feature space, Mika (1998) claimed that its corresponding reconstructed observation $\mathbf{x_r}$ in the original space does not always exist and need not be unique if it exists. If $\mathbf{x_r}$ does not exist, Mika proposed a nonlinear optimization approximation defined by

$$\widehat{\mathbf{x}}_\mathbf{r} = \text{argmin.} \left\{ \|\Phi(\mathbf{x_r}) - \Phi(\mathbf{x})_r\|^2 \right\} \tag{7.15}$$

$$= \text{argmin.} \left\{ \|\Phi(\mathbf{x_r})\|^2 - 2[\Phi(\mathbf{x})_r \Phi(\mathbf{x_r})^T] + \|\Phi(\mathbf{x})_r\|^2 \right\} \tag{7.16}$$

$$= \text{argmin.} \left\{ k(\mathbf{x_r}, \mathbf{x_r}) - 2\mathbf{z_k \Lambda_k}^{-\frac{1}{2}} \mathbf{U_k^T} k(\mathbf{X}, \mathbf{x_r}) \right\} + \|\Phi(\mathbf{x})_r\|^2, \tag{7.17}$$

where $\mathbf{z_k}$ stands for the projection of an observation in the feature space on the first k PCs and $\|\Phi(\mathbf{x})_r\|^2$ is independent of $\mathbf{x_r}$. According to (7.17), $\widehat{\mathbf{x}}_\mathbf{r}$ has an explicit form. For the Gaussian kernel of the form $k(x,z) = \exp(\|\mathbf{x}-\mathbf{z}\|^2/a)$, this optimization problem can be solved by a fixed-point iteration method. However, as mentioned by Mika (1998), this nonlinear optimization solution suffers from the local minimum problem and could be very time consuming.

7.4.3 Estimating Pre-Image Non-Iteratively

Kwok and Tsang (2003) proposed a non-iterative method to approximate $\mathbf{x_r}$ in the original space based on the distance constraints in the feature space. We will utilize this method to save the computational time. Williams (2002) claimed, for many kernel functions, there is a simple linear relationship between the Euclidean distance of two observations $\mathbf{x_i}$ and $\mathbf{x_j}$ and the Euclidean distance of their mappings, $\Phi(\mathbf{x_i})$ and $\Phi(\mathbf{x_j})$ in the feature space. Kwok and Tsang's method is based on this idea. In the feature space, the squared Euclidean distance between the reconstructed observation $\Phi(\mathbf{x})_r$ and the mapping of a training observation $\Phi(\mathbf{x_i})$ can be calculated as

$$d^2[\Phi(\mathbf{x})_r, \Phi(\mathbf{x_i})]_f = \|\Phi(\mathbf{x})_r\|^2 + \|\Phi(\mathbf{x_i})\|^2 - 2\Phi(\mathbf{x})_r\Phi(\mathbf{x_i})^T, \tag{7.18}$$

where

$$\begin{aligned}
\|\Phi(\mathbf{x})_r\|^2 &= \Phi(\mathbf{x})_r\Phi(\mathbf{x})_r^T \\
&= \left[\mathbf{z_k}\boldsymbol{\Lambda_k}^{-\frac{1}{2}}\mathbf{U_k^T}\Phi(\mathbf{X})\right]\left[\mathbf{z_k}\boldsymbol{\Lambda_k}^{-\frac{1}{2}}\mathbf{U_k^T}\Phi(\mathbf{X})\right]^T \\
&= \mathbf{z_k}\boldsymbol{\Lambda_k}^{-\frac{1}{2}}\mathbf{U_k^T}\mathbf{K}\mathbf{U_k}\boldsymbol{\Lambda_k}^{-\frac{1}{2}}\mathbf{z_k^T} \qquad (7.19) \\
\|\Phi(\mathbf{x_i})\|^2 &= \Phi(\mathbf{x_i})\Phi(\mathbf{x_i})^T \\
&= k(\mathbf{x_i}, \mathbf{x_i}) \qquad (7.20) \\
2\Phi(\mathbf{x})_r\Phi(\mathbf{x_i})^T &= 2\mathbf{z_k}\boldsymbol{\Lambda_k}^{-\frac{1}{2}}\mathbf{U_k^T}\Phi(\mathbf{X})\Phi(\mathbf{x_i})^T \\
&= 2\mathbf{z_k}\boldsymbol{\Lambda_k}^{-\frac{1}{2}}\mathbf{U_k^T}k(\mathbf{X}, \mathbf{x_i}).
\end{aligned}$$

We can see for some kernel functions, the corresponding squared Euclidean distance $d^2(\mathbf{x_r}, \mathbf{x_i})$ in the original data space can be calculated using $d^2[\Phi(\mathbf{x})_r, \Phi(\mathbf{x_i})]_f$. Considering a isotropic kernel

$$k(\mathbf{x_r}, \mathbf{x_i}) = f(\|\mathbf{x_r} - \mathbf{x_i}\|^2), \tag{7.21}$$

in which the kernel function is a function of the squared Euclidean distance $\|\mathbf{x_r} - \mathbf{x_i}\|^2$ There is

$$\begin{aligned} d^2[\Phi(\mathbf{x})_r, \Phi(\mathbf{x_i})]_f &= \|\Phi(\mathbf{x})_r\|^2 + \|\Phi(\mathbf{x_i})\|^2 - 2\Phi(\mathbf{x})_r\Phi(\mathbf{x_i})^T \\ &= \|\Phi(\mathbf{x})_r\|^2 + k(\mathbf{x_i}, \mathbf{x_i}) - 2f(\|\mathbf{x_r} - \mathbf{x_i}\|^2) \\ &= \|\Phi(\mathbf{x})_r\|^2 + k(\mathbf{x_i}, \mathbf{x_i}) - 2f(d^2(\mathbf{x_r}, \mathbf{x_i})). \end{aligned} \tag{7.22}$$

Therefore,

$$f(d^2(\mathbf{x_r}, \mathbf{x_i})) = \frac{1}{2}\{\|\Phi(\mathbf{x})_r\|^2 + k(\mathbf{x_i}, \mathbf{x_i}) - d^2[\Phi(\mathbf{x})_r, \Phi(\mathbf{x_i})]_f\}. \tag{7.23}$$

For the Gaussian kernel with the form

$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp\left[-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{2a^2}\right] = \exp\left[-\frac{d^2(\mathbf{x_i}, \mathbf{x_j})}{2a^2}\right],$$

the squared Euclidean distance in the original data space is given by

$$d^2(\mathbf{x_r}, \mathbf{x_i}) = -2a^2 \log\left\{\frac{1}{2}\{\|\Phi(\mathbf{x})_r\|^2 + k(\mathbf{x_i}, \mathbf{x_i}) - d^2[\Phi(\mathbf{x})_r, \Phi(\mathbf{x_i})]_f\}\right\}. \tag{7.24}$$

Similarly, for the exponential radial basis function (ERBF) kernel defined as

$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp\left[-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|}{2a^2}\right] = \exp\left[-\frac{\sqrt{d^2(\mathbf{x_i}, \mathbf{x_j})}}{2a^2}\right],$$

the squared Euclidean distance between $\mathbf{x_r}$ and $\mathbf{x_i}$ is given by

$$d^2(\mathbf{x_r}, \mathbf{x_i}) = \left\{-2a^2 \log\left\{\frac{1}{2}\{\|\Phi(\mathbf{x})_r\|^2 + k(\mathbf{x_i}, \mathbf{x_i}) - d^2[\Phi(\mathbf{x})_r, \Phi(\mathbf{x_i})]_f\}\right\}\right\}^2. \tag{7.25}$$

Consider the odd order polynomial kernel with the form

$$k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i}\mathbf{x_j^T} + a)^b,$$

where b stands for the odd positive integra, the squared distance between two points in the original space can be expressed by kernel functions. The derivation is given by

$$\begin{aligned} d^2(\mathbf{x_r}, \mathbf{x_i}) = \|\mathbf{x_r} - \mathbf{x_i}\|^2 &= \|\mathbf{x_r}\|^2 + \|\mathbf{x_i}\|^2 - 2\mathbf{x_r}\mathbf{x_i^T} \\ &= \mathbf{x_r}\mathbf{x_r^T} + \mathbf{x_i}\mathbf{x_i^T} - 2\mathbf{x_r}\mathbf{x_i^T} \\ &= (\mathbf{x_r}\mathbf{x_r^T} + a) + (\mathbf{x_i}\mathbf{x_i^T} + a) - 2(\mathbf{x_r}\mathbf{x_i^T} + a) \\ &= [k(\mathbf{x_r}, \mathbf{x_r})]^{\frac{1}{b}} + [k(\mathbf{x_i}, \mathbf{x_i})]^{\frac{1}{b}} - 2[k(\mathbf{x_r}, \mathbf{x_i})]^{\frac{1}{b}} \\ &= [\Phi(\mathbf{x})_r \Phi(\mathbf{x})_r^T]^{\frac{1}{b}} + [k(\mathbf{x_i}, \mathbf{x_i})]^{\frac{1}{b}} - 2[\Phi(\mathbf{x})_r \Phi(\mathbf{x_i})^T]^{\frac{1}{b}}. \quad (7.26) \end{aligned}$$

For the sigmoid kernel defined as

$$k(\mathbf{x_i}, \mathbf{x_j}) = \tanh(a \cdot \mathbf{x_i}\mathbf{x_j^T} + b),$$

the inner dot product can be expressed as

$$\mathbf{x_i}\mathbf{x_j^T} = \frac{\tanh^{-1}(k(\mathbf{x_i}, \mathbf{x_j})) - b}{a}.$$

The squared Euclidean distance between $\mathbf{x_r}$ and $\mathbf{x_i}$ is can be calculated as:

$$
\begin{aligned}
d^2(\mathbf{x_r}, \mathbf{x_i}) = \|\mathbf{x_r} - \mathbf{x_i}\|^2 &= \|\mathbf{x_r}\|^2 + \|\mathbf{x_i}\|^2 - 2\mathbf{x_r}\mathbf{x_i^T} \\
&= \mathbf{x_r}\mathbf{x_r^T} + \mathbf{x_i}\mathbf{x_i^T} - 2\mathbf{x_r}\mathbf{x_i^T} \\
&= \frac{\tanh^{-1}(k(\mathbf{x_r}, \mathbf{x_r})) - b}{a} + \frac{\tanh^{-1}(k(\mathbf{x_i}, \mathbf{x_i})) - b}{a} \\
&\quad -2\frac{\tanh^{-1}(k(\mathbf{x_r}, \mathbf{x_i})) - b}{a} \\
&= \frac{\tanh^{-1}(\Phi(\mathbf{x})_r \Phi(\mathbf{x})_r^T) - b}{a} + \frac{\tanh^{-1}(k(\mathbf{x_i}, \mathbf{x_i})) - b}{a} \\
&\quad -2\frac{\tanh^{-1}(\Phi(\mathbf{x})_r \Phi(\mathbf{x_i})^T) - b}{a}. \qquad (7.27)
\end{aligned}
$$

Now, we know the distance between the reconstructed observation $\mathbf{x_r}$ and each training observation and the distance between $\Phi(\mathbf{x_r})$ and each feature mapping of the training observations. The distances with the neighbors are the most important in determining the location of $\mathbf{x_r}$ (Kwok and Tsang, 2003). For the Gaussian kernel function, the contribution of $\mathbf{x_i}$ drops exponentially with the increasing distance from $\mathbf{x_r}$. Follows (Kwok and Tsang, 2003), we choose the m nearest neighbors of $\Phi(\mathbf{x_r})$ in the feature space and identify the corresponding neighbors of $\mathbf{x_r}$ in the original data space. As claimed by Kwok, one can use the m nearest neighbors of $\mathbf{x_r}$ directly. Let an $m \times p$ matrix $\mathbf{X_{nb}}$ stand for the neighborhood data whose each row stands for a specific p-dimensional neighbor of $\mathbf{x_r}$. As discussed above, we are able to obtain the distances between $\mathbf{x_r}$ and its neighbors. Let an $m \times 1$ vector $\mathbf{d^2}$ stand for the m squared distances from $\mathbf{x_r}$, our goal is to find the coordinates of $\mathbf{x_r}$ to preserve the distances in $\mathbf{d^2}$ as much as possible. Follows (Gower, 1968), knowing the coordinates of orthogonal axes of the neighbor points and the distances between an unknown point and these neighbor points, the coordinates of the unknown point can be estimated using a least-squares method. Already knowing the coordinates $\mathbf{X_{nb}}$, its corresponding coordinates in an orthogonal space can be obtained by performing singular value decomposition (SVD). $\mathbf{X_{nb}}$ is first centered by its mean $\bar{\mathbf{x}}_{\mathbf{nb}}$. SVD of the centered coordinates $\mathbf{X_{nb(c)}^T}$ with rank of q is defined as

$$\mathbf{X}^{\mathbf{T}}_{\mathbf{nb(c)}} = \mathbf{USV}^{\mathbf{T}} = \mathbf{UZ}, \tag{7.28}$$

where $\mathbf{U}$ is a $p \times q$ matrix whose columns are orthonormal eigenvectors and the $q \times m$ matrix $\mathbf{Z}$ is the projection of the m points (observations) on the q principal components. Let the row vector $\mathbf{z_i}$ stand for the projection of the i^{th} point. Since $\mathbf{X}^{\mathbf{T}}_{\mathbf{nb(c)}}$ is centered by its mean vector, $\|\mathbf{z_i}\|^2$ is the squared Euclidean distance from $\bar{\mathbf{x}}_{\mathbf{nb}}$. Let an $m \times 1$ vector $\mathbf{d_0^2}$ stand for the m squared distances between m neighbor points and their center $\bar{\mathbf{x}}_{\mathbf{nb}}$. The projection of $\mathbf{x_r}$ on the q principal components can be estimated by

$$\hat{\mathbf{z}}_{\mathbf{r}} = -\frac{1}{2}(\mathbf{ZZ}^{\mathbf{T}})^{-\mathbf{1}}\mathbf{Z}(\mathbf{d^2} - \mathbf{d_0^2}) = -\frac{1}{2}\mathbf{\Lambda}^{-\mathbf{1}}\mathbf{V}^{\mathbf{T}}(\mathbf{d^2} - \mathbf{d_0^2}). \tag{7.29}$$

The corresponding coordinates in the original space can be obtained by

$$\hat{\mathbf{x}}_{\mathbf{r}} = (\mathbf{U}\hat{\mathbf{z}}_{\mathbf{r}})^{\mathbf{T}} + \bar{\mathbf{x}}_{\mathbf{nb}}. \tag{7.30}$$

The only parameter that a research has to configure is the number of neighbors. Following Kwok's work, 10 is the default number of neighbors.

7.5 Kernel Principal Components Regression

Principal components regression (PCR) can be used to deal with collinearity problems in regression using orthogonal PCs as the regressors. The PCs with very small variances will be drop to prevent from overfitting (Jolliffe, 2002). Similar as the standard multiple regression model, the univariate PCR can be expressed as:

$$\mathbf{y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{7.31}$$

where the $n \times 1$ vector $\mathbf{y}$ represents dependent variable with n observations, the $n \times k$ matrix $\mathbf{Z}$ represents the retained k PC scores, $\boldsymbol{\beta}$ is a column vector of k regression coefficients and

$\boldsymbol{\epsilon}$ is the independent and identically distributed (i.i.d.) random error with mean of zero and constant variance σ^2. The estimated regression coefficients are

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{y}. \tag{7.32}$$

Using the complexity measure defined in (3.32) for orthogonal components, the ICOMP for the univariate PCR can be defined as

$$\text{ICOMP}(\text{C}_{1\text{F}}) = n\log(2\pi) + n\log(\hat{\sigma}^2) + n + 2\text{C}_{1\text{F}}(\widehat{\text{Cov}}(\widehat{\mathbf{PC}})), \tag{7.33}$$

where

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{Z}\hat{\boldsymbol{\beta}})^{\mathbf{T}}(\mathbf{y} - \mathbf{Z}\hat{\boldsymbol{\beta}})}{n}$$

is the estimated variance of the random error and

$$\widehat{\text{Cov}}(\widehat{\mathbf{PC}}) = \frac{\boldsymbol{\Lambda}_{\boldsymbol{k}}}{n-1}$$

for kernel PCA, in which $\boldsymbol{\Lambda}_{\boldsymbol{k}}$ is the diagonal matrix whose k diagonal elements are the first k eigenvalues of the kernel matrix.

Using the model selection criterion ICOMP, one can choose the number of principal components, the kernel functions and/or the subset variables of $\mathbf{X}$ which contribute to the response prediction the most.

7.6 Model Selection Using ICOMP

7.6.1 Choosing Number of Retained Kernel PCs

A decision must be made on how many principal components should be retained in order to find a balance point between the good generalization and the low reconstruction error. Some non-statistical guidelines include:

1. Retain components to account for a specified high percentage of the total variance, say 90%.

2. Retain the components whose eigenvalues are greater than the average of all the eigenvalues.

3. Observe the scree graph (Cattell, 1966), the plot of eigenvalues versus the indices of the eigenvalues and look for the natural cut-off point.

4. Use the Kaiser's rule (Kaiser, 1960). That is, for the PCA based on the correlation matrix, ignore the PCs whose variances are smaller than 1. However, it can be argued that this method retains too few variables.

5. Cross-validation is currently the most widely used method to decide the number of retained PCs. However, the computational intensity is the major drawback.

The above techniques for choosing the optimal number of PCs are either subjective or lack of statistical foundation or computationally expensive. There exists a statistical method that decides the number of PCs known as Bartlett's test. It is to test if the last eigenvalues are all equal versus at least two of the last eigenvalues are different. The test statistic of this hypothesis test approximately follows a χ^2 distribution. In practice, when the variables are fairly highly correlated, Bartlett's test will often indicate more than enough number of components.

In this chapter, we use an information measure approach to decide the number of PCs retained. We assume the random error of the general multivariate PCA model defined in (7.13) follows a multivariate normal distribution $N(\mathbf{0}, \mathbf{\Sigma})$. Using the complexity $C_{1F}(\cdot)$ for the orthogonal components defined in (3.32), the ICOMP form for PCA or Kernel PCA can be defined as:

$$\begin{aligned}
\text{ICOMP}(\widehat{\text{Cov}}(\mathbf{PC}))_{\text{Multivar.}} &= \text{Lack of Fit} + 2C_{1F}(\widehat{\text{Cov}}(\mathbf{PC})) \\
&= np\log(2\pi) + n\log|\hat{\mathbf{\Sigma}}| + np + 2C_{1F}(\widehat{\text{Cov}}(\mathbf{PC})) \quad (7.34) \\
&= np\log(2\pi) + n\log|\hat{\mathbf{\Sigma}}| + np + 2\frac{1}{4\overline{\lambda}_a^2}\sum_{i=1}^{s}(\lambda_i - \overline{\lambda}_a)^2 \quad (7.35)
\end{aligned}$$

where

p represents the number of the variables of $\mathbf{X}$,

$\widehat{\mathbf{\Sigma}}$ is a $p \times p$ matrix, which is the likelihood estimate of the covariance of the random errors given by

$$\widehat{\mathbf{\Sigma}} = \frac{(\mathbf{X} - \mathbf{X_r})^{\mathbf{T}}(\mathbf{X} - \mathbf{X_r})}{n} \quad (7.36)$$

$\widehat{\text{Cov}}(\mathbf{PC})$ is a diagonal matrix , since PCs are orthogonal, whose diagonal elements are the variances of the retained PCs.

The information measure approach finds the balance point between the small reconstruction error and the complexity of the PCs retained. As one can observe from the above equation, the complexity increases as the spread of the variances of the PCs increases. $C_{1F}(\cdot)$ is a monotonously increasing function of the number of PCs. The optimal number of PCs is the minimizer of ICOMP. Furthermore, we can using ICOMP to choose the optimal kernel function and its appropriate parameter without using validation data. The results of the numerical experiments will be summarized in the next section.

7.7 Numerical Examples

7.7.1 Simulated Toy Example

We first use the two-dimensional three-cluster toy example illustrated by Schölkopf (1998). The centers of the three clusters are $(-0.5, -0.2)$, $(0, 0.6)$ and $(0.5, 0)$ respectively. The

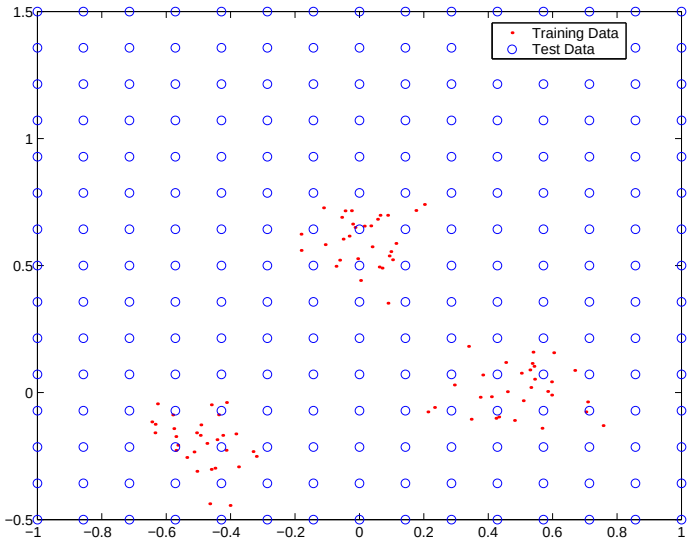


Figure 7.1: Scatter Plot of the 2D 3-Cluster Toy Example

Gaussian noise of $N(0, 0.1^2)$ is added to generate 30 training points around each cluster center. 225 testing points are spread evenly in the span of the train points (Figure 7.1).

The first advantage of using kernel PCA is that it allows more components to be extracted to ensure both small reconstruction error and high generalization. If we use linear PCA, only two nonzero principal components can be extracted and they explained about 64% and 36% of the total variance respectively. Keeping both components will fail to do de-noising or reduce collinearity and retaining the first PC will lose 36% of the total variance. The second advantage is that kernel PCA describes the nonlinear pattern of the data better than linear PCA. As one can observe from (Figure 7.2), linear PCA can not express the pattern of 3 clusters. By using the Gaussian RBF kernel with $a^2 = 0.05$ suggested by Schölkopf (1998), we see that the first two PCs separate three clusters and the rest PCs further separate the points inside the clusters (Figure 7.3).

We can use the ICOMP form for KPCA defined by (7.35) to choose the optimal number of PCs to retain given a specific kernel function. Using 25 neighbors to estimate the pre-image (reconstructed observation), the optimal number of PCs retained is 8 according to the minimized ICOMP. The first 8 PCs explained 92.8% of the total variance and the rest

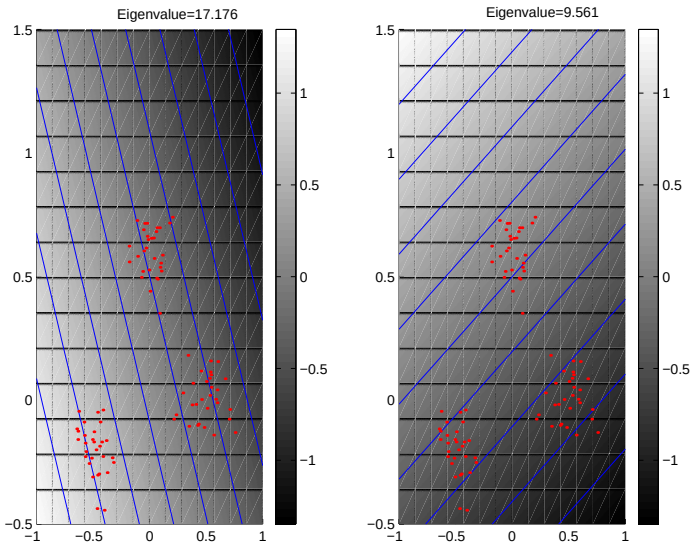


Figure 7.2: Toy Example: PC Plots Using Linear Kernel

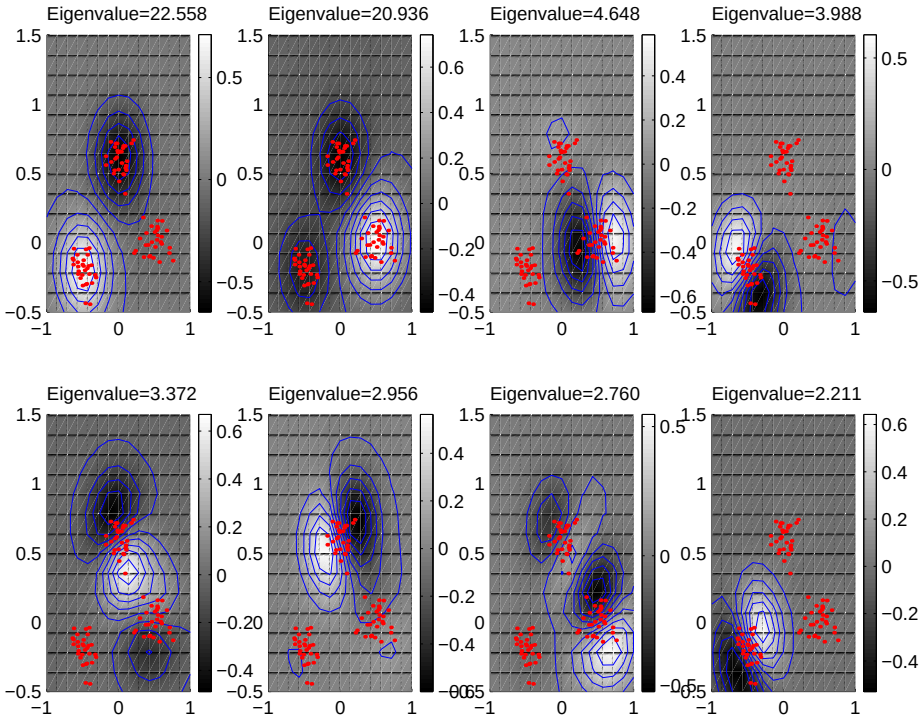


Figure 7.3: Toy Example: PC Plots Using Gaussian RBF Kernel($a^2 = 0.05$)

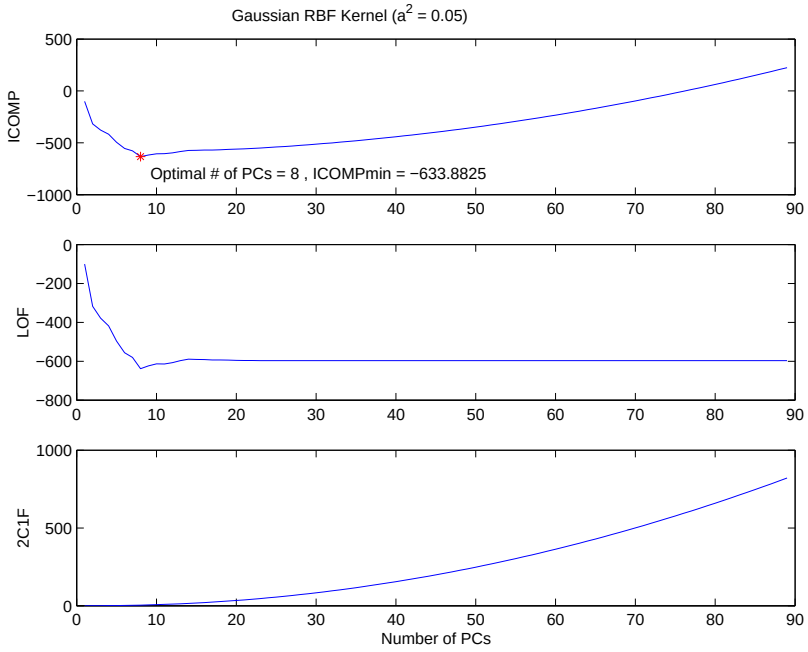


Figure 7.4: Toy Example: ICOMP vs. Number of PCs. Gaussian RBF Kernel($a^2 = 0.05$)

nuisance PCs can be ignored without losing the important information (Figure 7.4). The Mean Squared Error (MSE) of the reconstruction is 1.6995×10^{-3} for the x coordinates and 1.6893×10^{-3} for the y coordinates (Figure 7.5). Adding more PCs does not yield significantly smaller MSE for either x or y. Furthermore, we can use ICOMP as a model selection criterion to choose the appropriate parameters of the kernel function as well as the number of optimal PCs given the kernel function. We first compare different parameters of the Gaussian RBF kernel function. Given the difference scale parameters between 0.01 and 1, the optimal parameter for Gaussian RBF kernel is 0.06 where the number of retained PCs is 8 (Figure 7.6). We can even compare different kernels functions using ICOMP. In Table 7.1, we compared the polynomial kernels with the optimal Gaussian RBF kernel, the optimal exponential RBF kernel and the optimal Sigmoid kernel. The comparison result shows the Gaussian RBF kernel with $a^2 = 0.06$ is the most appropriate kernel function for the given toy example.

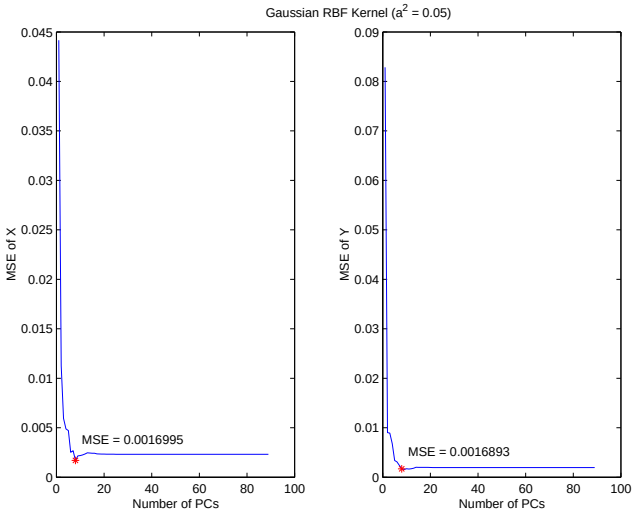


Figure 7.5: Toy Example: MSE vs. Number of PCs. Gaussian RBF Kernel($a^2 = 0.05$)

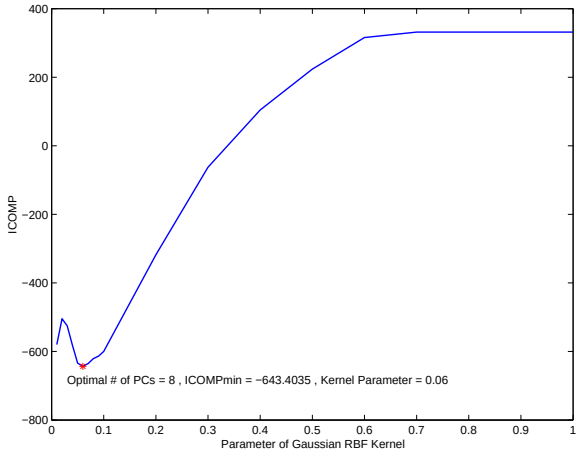


Figure 7.6: Toy Example: Parameters of Gaussian RBF Kernel vs. ICOMP

Table 7.1: Toy Example: Comparing Kernel Functions

Kernel Function	# PCs	ICOMP	VAR
Polynomial Kernel $(< x, z > +1)^3$	4	-418.3247	98.6%
Polynomial Kernel $(< x, z > +1)^5$	5	-476.0168	98.4%
Polynomial Kernel $(< x, z > +1)^7$	5	-412.0198	97.1%
Polynomial Kernel $(< x, z > +1)^9$	6	-340.6103	97.7%
Gaussian Kernel $\exp(-\frac{\|x-z\|^2}{2\times 0.06})$	8	$\mathbf{-643.4035}$	94.5%
Exponential RBF Kernel $\exp(-\frac{\|x-z\|}{2\times 0.03})$	58	-495.3857	88.9%
Sigmoid Kernel $\tanh(0.04 < x, z > +2)$	2	−479.5337	99.9%

7.7.2 Corn Data

This data set, originally taken at Cargill, consists of 80 samples of corn measured on 3 different near-infra-red (NIR) spectrometers. The wavelength range is 1100-2498nm at 2nm intervals, which makes the number of variables be 700 (channels). These 700 variables (regressors) are highly correlated since the first PC represents the 99.5% of the total variance. The moisture, oil, protein and starch values represent four response variables. This is a typical data set for the kernel method. It is more computational efficient to work with the 80×80 kernel matrix in the kernel PCR than work with the 700×700 covariance matrix in the linear PCR. Following Rosipal's (2001) works, instead of modeling the real responses mentioned above, four simulated responses are used since using the nonlinear PCR does not yield significantly better results than the linear PCR does. The generated responses are

$$y_{1i} = \exp\left(\frac{\mathbf{x_i x_i^T}}{2c_1}\right) \tag{7.37}$$

$$y_{2i} = \exp\left(\frac{\mathbf{x_i A^{-1} x_i^T}}{2c_2}\right) \tag{7.38}$$

$$y_{3i} = \left(\frac{\mathbf{x_i x_i^T}}{c_1}\right)^3 y_{1i} \tag{7.39}$$

$$y_{4i} = 0.3y_{1i} + 0.25y_{2i} - 0.7y_{3i}, \tag{7.40}$$

where

$$c_1 = \sum_{i=1}^{80} \mathbf{x_i x_i^T}$$

$$c_2 = \sum_{i=1}^{80} \mathbf{x_i A^{-1} x_i^T},$$

and where $\mathbf{A}$ is a symmetric matrix with off-diagonal elements set to 0.8 and diagonal elements set to 1.0. We use the first 60 observations as the training data and the remaining 20 observations as the testing data. Independent Gaussian noise is added to each one of the four simulated responses. The noise level in this experiment is defined as $\sigma_{\text{noise}}/\sigma_y$.

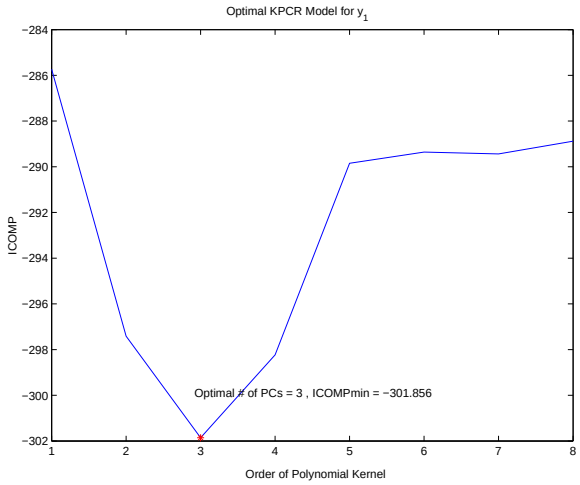


Figure 7.7: Corn Example: Order of Polynomial Kernel vs. ICOMP

In stead of using the leave-one-out cross validation applied by Rosipal (2001), we use ICOMP as the model selection criterion to choose the optimal number of principal components retained, which is more computationally efficient.

Compared with $\mathbf{y_3}$ and $\mathbf{y_4}$, $\mathbf{y_1}$ and $\mathbf{y_2}$ have relative weaker nonlinear relationships with the regressors, the principal component scores. To model $\mathbf{y_1}$, for instance, different orders of polynomial kernel functions are compared. The optimal polynomial kernel function for modeling $\mathbf{y_1}$ is the 3^{rd} order polynomial kernel (Figure 7.7) and the number of PC retained is 3 (Figure 7.8).

We compare the optimal polynomial kernel with the selected Gaussian RBF kernel, exponential RBF kernel and sigmoid kernel in this research (Table 7.2). The interested readers may use ICOMP to evaluate any other kernel functions if needed. The selected model is the minimizer of ICOMP. We use R^2 and MSE of the test data to validate the models selected by ICOMP. It is shown that the models selected by ICOMP also have the smallest testing error or their testing errors are close to the smallest ones.

Since the 700 variables (channels) in this data set is highly correlated, an intuitive thinking is that not all the 700 variables are needed for the modeling. Again, ICOMP can be used as the criterion to chose the best subset of the 700 variables with good prediction

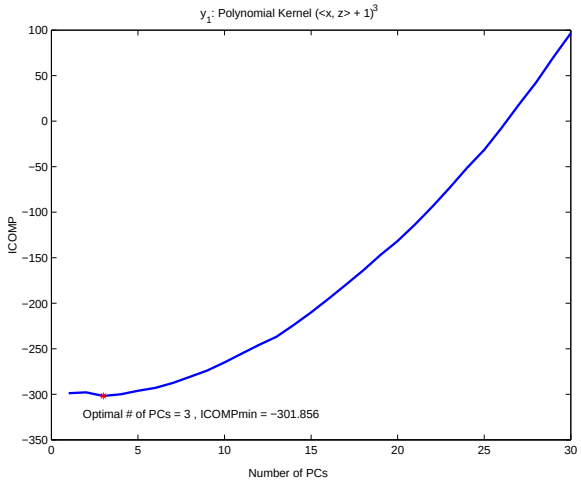


Figure 7.8: Corn Example: Number of PCs vs. ICOMP (Polynomial kernel order 3)

Table 7.2: Corn Data: Univariate KPCR

Kernel Function	# PCs	ICOMP	R^2_{test}	MSE_{test}
y_1				
Polynomial Kernel $(< x, z > +1)^3$	3	$\mathbf{-301.8560}$	0.97	3.6×10^{-4}
Gaussian Kernel $\exp(-\frac{\|x-z\|^2}{2\times 4})$	4	-299.1085	0.96	1.1×10^{-3}
Exponential RBF Kernel $\exp(-\frac{\|x-z\|}{2\times 30})$	3	-301.3357	0.84	4.5×10^{-3}
Sigmoid Kernel $\tanh(0.04 < x, z > +1)$	1	-80.3423	0.45	1.5×10^{-2}
y_2				
Polynomial Kernel $(< x, z > +1)^3$	3	-301.8490	0.96	3.9×10^{-4}
Gaussian Kernel $\exp(-\frac{\|x-z\|^2}{2\times 3})$	3	$\mathbf{-316.8399}$	0.95	5.5×10^{-4}
Exponential RBF Kernel $\exp(-\frac{\|x-z\|}{2\times 10})$	12	-303.4989	0.92	7.7×10^{-4}
Sigmoid Kernel $\tanh(0.04 < x, z > +1)$	2	-106.1552	0.62	3.8×10^{-3}
y_3				
Polynomial Kernel $(< x, z > +1)^7$	1	-36.3522	0.98	7.4×10^{-2}
Gaussian Kernel $\exp(-\frac{\|x-z\|^2}{2\times 2})$	3	$\mathbf{-38.6037}$	0.91	0.30
Exponential RBF Kernel $\exp(-\frac{\|x-z\|}{2\times 10})$	8	-38.2194	0.77	0.78
Sigmoid Kernel $\tanh(0.0001 < x, z > +1)$	1	41.5229	0.77	0.79
y_4				
Polynomial Kernel $(< x, z > +1)^8$	1	$\mathbf{-66.8598}$	0.98	0.02
Gaussian Kernel $\exp(-\frac{\|x-z\|^2}{2\times 10})$	2	-56.2145	0.95	0.07
Exponential RBF Kernel $\exp(-\frac{\|x-z\|}{2\times 2})$	8	-65.5052	0.75	0.34
Sigmoid Kernel $\tanh(0.0001 < x, z > +1)$	1	0.4988	0.77	0.32

ability and small model complexity. However, there are $2^{700} - 1$ possible subset models. All possible subsets selection is not a reasonable choice. We use GA to search the optimal subset models. The configuration of GA options are given as follows (Table 7.3). Given the chosen optimal kernel functions when all the 700 variables are used, the optimal subset variables chosen by GA using ICOMP as the fitness function is summarized in Table 7.4.

7.7.3 KPCR: Sinc Function

Again, we apply the popular sinc function to test our model selection criterion for KPCR models. In this numerical example, we demonstrate how ICOMP chooses the optimal kernel function and the corresponding parameters.

We start with the widely used Gaussian RBF kernel. Given a fixed scale parameter $\gamma = 3$, as the number of PCs retained increases, the lack-of-fit decreases while the complexity of the model increases (Table 7.5). ICOMP finds the balance point where 6 PCs are retained (Figure 7.9). This is also the optimal choice where the testing data MSE is around the minimum. Using the 1000 holdout observations, 9 PCs are retained. Its resulting testing data MSE is a little higher.

ICOMP is also capable of choosing the optimal scale parameter. However, the range of the scale parameters must be selected carefully. It is observed that when the scale parameter of the Gaussian RBF is too small (leading to serious overfitting), the kernel matrix becomes inaccurate. It is because the small elements of the kernel matrix are treated as zero due to the limited precision that a computer may achieve. When selecting the range of the scale

Table 7.3: GA Options for KPCR

Number of Generations	30
Population Size	30
Type of Crossover	Uniform
Probability of Crossover	0.7
Probability of Mutation	0.01
Keep the Best Model from the Previous Generation	Yes

Table 7.4: Corn Data: Searching Subset Variables using GA

Kernel Function	# PCs	ICOMP	R^2_{test}	MSE_{test}
y1				
Polynomial Kernel $(< x, z > +1)^3$	1	-300.7565	0.95	0.00137
Variable ID: 597, 601, 603, 629, 648				
y2				
Gaussian Kernel $\exp(-\frac{\|x-z\|^2}{2\times 3})$	4	-299.7381	0.94	6.2×10^{-4}
Variable ID: 647, 649, 651, 654, 656, 658, 659, 661, 662, 663, 664, 665, 668, 669, 670, 671, 672, 673, 674, 675, 677, 679, 680, 682, 683, 688, 689, 692, 694, 696, 697, 699				
y3				
Gaussian Kernel $\exp(-\frac{\|x-z\|^2}{2\times 2})$	5	-29.6022	0.93	0.22
Variable ID: 647, 648, 655, 656, 659, 660, 661, 664, 665, 668, 674, 675, 678, 679, 680, 684, 685, 690,				
y4				
Polynomial Kernel $(< x, z > +1)^8$	4	-67.8688	0.94	0.082
Variable ID: 607, 656				

Table 7.5: Selecting the Number of PCs using ICOMP

γ	ICOMPC1F	LOF	Complexity	MSEho	MSEtest
1	80.0	79.0	1.0	0.13413	0.13501
2	19.3	17.3	2.0	0.07229	0.07233
3	21.4	18.3	3.1	0.07229	0.07233
4	-80.0	-84.2	4.2	0.03023	0.02803
5	-77.8	-83.2	5.5	0.03025	0.02810
6	-156.5	-163.4	6.9	0.01113	0.01046
7	-153.9	-162.5	8.6	0.01116	0.01050
8	-151.7	-162.2	10.5	0.01094	0.01046
9	-148.9	-161.5	12.6	0.01102	0.01051
10	-145.9	-160.9	15.0	0.01105	0.01058
11	-142.9	-160.4	17.5	0.01106	0.01071
12	-143.6	-163.9	20.3	0.01124	0.01109

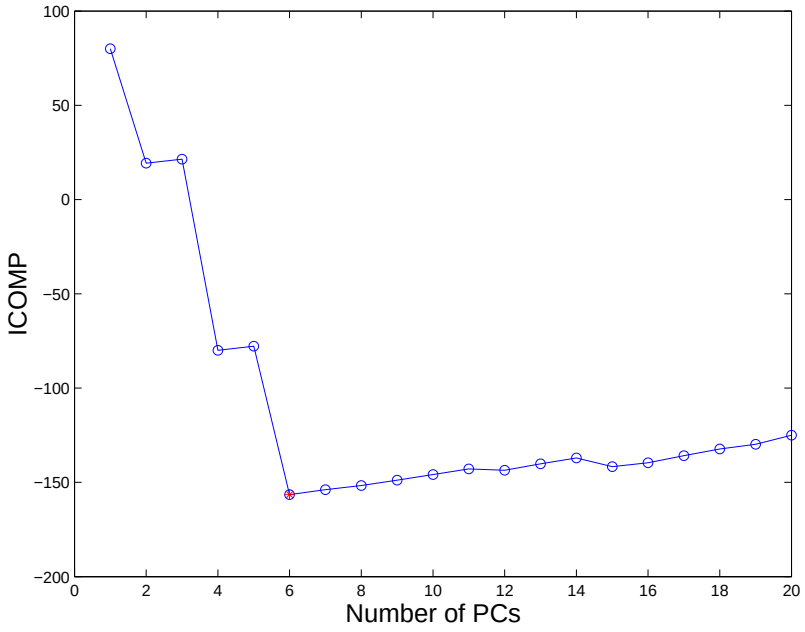


Figure 7.9: Sinc Function: Selecting Number of PCs using ICOMP

parameter candidates, one should avoid this area. Otherwise, the covariance matrix for ICOMP may be biased. For the sinc function, one may plot the nature log of the condition number of the kernel matrix against the scale parameter (Figure 7.10). It can be shown that the kernel matrix becomes inaccurate when the scale parameter γ is smaller than 1. Therefore, we decide to choose the optimal scale parameter from the range of [2, 10].

In an experiment of 100 simulations, the most chosen scale parameters are 3 or 4, where the number of retained PCs is most likely around 6. Using ICOMP to choose the kernel parameter and the number of PCs, the predictive performance is similar to that where the 1000 holdout observations are used (Table 7.6).

We now compare three popular kernels, including the polynomial kernel, Gaussian kernel and cubic kernel, using ICOMP as the model selection criterion. The simulation are repeated 100 times. In each simulation, different random errors are generated. It is concluded that the Gaussian kernel is chosen 75 times, the cubic kernel is chosen 25 times while polynomial kernel is not good enough to build a model that generalizes well (Table 7.7).

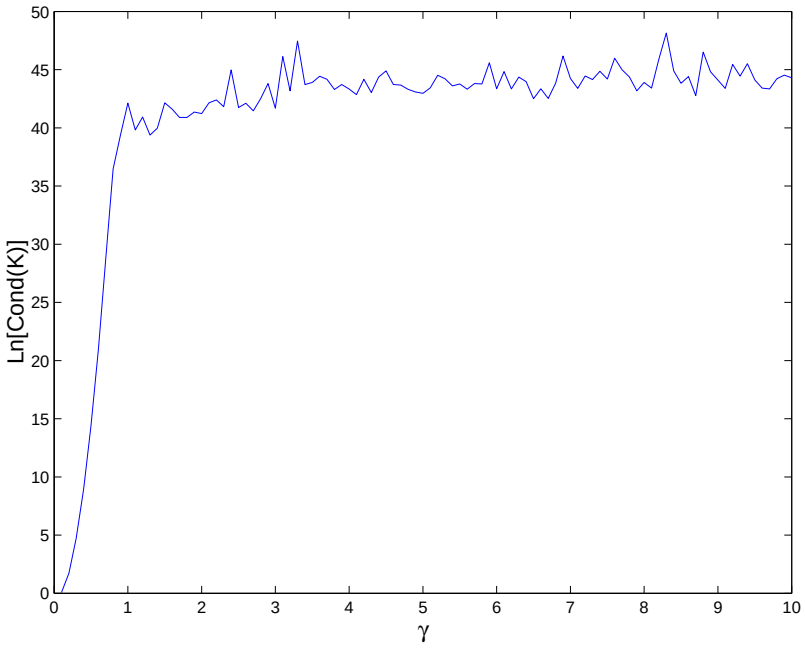


Figure 7.10: Finding the Range of the Scale Parameters

Table 7.6: Sinc Function: Selecting Scale Parameters (100 runs)

Method	Ave. MSEtest	Std. MSEtest	Ave. # PCs
ICOMPC1F	0.01095	0.00069	6.3
AIC	0.01093	0.00070	6.3
BIC	0.01086	0.00069	6.0
Holdout	0.01081	0.00066	7.2

Table 7.7: Sinc Function: Comparing Kernels (100 runs)

Kernel	Ave. MSEtest	Std. MSEtest	Ave. # PCs	Chosen Freq.
Polynomial	0.07674	0.00169	3	0
Gaussian RBF	0.01095	0.00069	6.3	75
Cubic	0.01085	0.00065	5.2	25

Chapter 8

Relevance Vector Machine

8.1 Introduction

The support vector machines (SVMs), also known as the kernel-based methods (Shawe-Taylor and Cristianini, 2004), are a set of nonlinear statistical learning techniques that have drawn much attention since mid-90s. SVM was developed at AT&T Bell Laboratories by Vapnik and his co-workers (Vapnik, 1995). First appears as a nonlinear binary classifier, SVM has been widely used in different nonlinear modeling areas including regression, principal component analysis, canonical correlation analysis, discriminant analysis, clustering and classification. The relevance vector Machine (RVM), first proposed by Tipping (2001), is an improvement of SVM from the Baysian learning perspective. It offers a number of advantages over the traditional SVM. High sparsity of the kernel matrix is one of the major highlights of RVM.

An open question left for RVM is the model selection method. The model selection issues of RVM include choosing the form of the kernel function, the parameters of the kernel function and subset regressors. Currently in the literature, the widely utilized model selection method for RVM is still the cross-validation (Tipping, 2001). However, the increased computational intensity is the major drawback. In this chapter, we use an information complexity (ICOMP) (Bozdogan, 1988, 2004a) measure of the RVM model as a novel model

selection criterion for RVM. ICOMP provides a simple real valued index for each model, which measures both the lack-of-fit (LOF) and the complexity. The complexity of the model is controlled to prevent from the overfitting. The optimal RVM model is chosen as the minimizer of ICOMP without using the validation data. We also apply Genetic Algorithm (GA) to be efficient when searching the optimal subset variables

In the next section, we briefly introduce SVM. The procedure of RVM for the regression and the logistic regression will be detailed in the sections followed. We then derive the form of ICOMP for RVM. Some benchmark examples are used for the demonstration including sinc function, Friedman's data, Boston's housing data and Ripley's data.

8.2 Support Vector Machine

In SVMs, the original p-dimensional training data (in the input space) $\mathbf{X} \subseteq \mathbf{R}^p$ is nonlinearly transformed to a high-dimensional feature space through Φ. The traditional linear statistical learning techniques then are performed on $\Phi(\mathbf{X})$, the nonlinear projection of $\mathbf{X}$ in the feature space. The mapping from $\mathbf{X}$ to $\Phi(\mathbf{X})$ is conducted through the kernel function efficiently. The kernel function $k(\mathbf{x_i}, \mathbf{x_j})$ is the inner product of two observations in the feature space:

$$k(\mathbf{x_i}, \mathbf{x_j}) =< \mathbf{\Phi}(\mathbf{x_i}), \mathbf{\Phi}(\mathbf{x_j}) > . \tag{8.1}$$

Using kernel functions, the explicit form of $\Phi(\mathbf{X})$ is not needed. The kernel matrix $\mathbf{K}$, sometimes called the gram matrix, is a finitely positive semi-definite symmetrical matrix with elements $\mathbf{K}_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$. The kernel matrix is similar as the covariance matrix in the original data space, which contains the information for the learning. If the dimension of the input space is higher than the number of training observations, applying the kernel matrix is more efficient in computation.

Consider the SVM model for the multiple regression in the feature space defined by

$$y = w_0 + \Phi(\mathbf{x})\mathbf{w}. \tag{8.2}$$

The idea of SVM is to find the coefficients $\mathbf{w}$ that minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{8.3}$$

subject to

$$\begin{cases} y_i - \Phi(\mathbf{x_i})\mathbf{w} - w_0 \leq \epsilon + \xi_i \\ \Phi(\mathbf{x_i})\mathbf{w} + w_0 - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{8.4}$$

The collinearity problem is controlled by trying to minimize the norm of the coefficients, $\|\mathbf{w}\|$. The smaller $\|\mathbf{w}\|$ will lead to a simpler model, thus better generalization. Minimizing the second term of (8.3) controls the modeling error, however, will require more complicate model, thus be less general. The constant $C > 0$ controls the trade off between the model generalization and the goodness-of-fit. ϵ is an insensitive parameter which defines a margin such that the prediction error will be penalized only when it is beyond ϵ. This is corresponding to an "ϵ-insensitive loss function" defined by

$$|\xi|_\epsilon := \begin{cases} 0 & \text{if } |\xi| \leq \epsilon; \\ |\xi| - \epsilon, & \text{otherwise.} \end{cases} \tag{8.5}$$

Since the explicit form of $\Phi(\mathbf{X})$ is unknown, SVM is conducted using its dual form through the kernel function. That is

$$y(\mathbf{x}) = w_0 + \sum_{i=1}^{n} K(\mathbf{x}, \mathbf{x_i})w_i. \tag{8.6}$$

Tipping (2001) pointed out several disadvantages of SVM despite its big success in the machine learning:

1. Cross-validation is needed to estimate the insensitive parameter ϵ and the trade off parameter C. This procedure will be either time consuming or wasting of data.

2. Predictions are not probabilistic. Therefore, the uncertain measurement of the prediction is not possible.

3. The number of coefficients grows steeply with the size of the training observations.

4. The kernel function must satisfy Mercer's condition.

RVM is a Bayesian treatment of SVM that does not suffer from any of the above limitations. We describe the details of RVM for regression and logistic regression in the next section.

8.3 Methodology of RVMs

8.3.1 Relevance Vector Regression

Given the observation x_i and the response y_i, the relevance vector regression (RVR) model is defined by

$$y_i = f(\mathbf{x_i}) + \varepsilon_i \qquad i = 1, 2, 3, ..., n, \tag{8.7}$$

where $\varepsilon_i \sim N(0, \sigma^2)$ are i.i.d. random errors. Using the Bayesian inference notes, there is $p(y_i|\mathbf{x_i}) = N(y_i|f(\mathbf{x_i}), \sigma^2)$, where $f(\mathbf{x_i})$ is defined by

$$f(\mathbf{x_i}) = [\ 1 \quad \mathbf{K}(\mathbf{x_i}, \mathbf{X})\ ]\mathbf{w} = \boldsymbol{\psi}(\mathbf{x_i})\mathbf{w}, \tag{8.8}$$

where $\mathbf{K}(\mathbf{x_i}, \mathbf{X}) = [\mathbf{k}(\mathbf{x_i}, \mathbf{x_1}) \quad \mathbf{k}(\mathbf{x_i}, \mathbf{x_2}) \quad \dots \quad \mathbf{k}(\mathbf{x_i}, \mathbf{x_n})]$ is the $1 \times n$ kernel function row vector and $\mathbf{w} = [w_0, w_1, \dots, w_n]^T$ contains the regression coefficients. Assume the response y_i are independent to each other, the likelihood function of the n observations can be defined by

$$p(\mathbf{y}|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left[\frac{\|\mathbf{y} - \boldsymbol{\Psi}\mathbf{w}\|^2}{2\sigma^2}\right], \tag{8.9}$$

where $\boldsymbol{\Psi} = [\ \boldsymbol{\psi}(\mathbf{x_1}) \quad \boldsymbol{\psi}(\mathbf{x_2}) \quad \dots \quad \boldsymbol{\psi}(\mathbf{x_n})\ ]^T$ is an $n \times (n+1)$ matrix.

To avoid the overfitting when estimating the parameters $\mathbf{w}$ and σ^2, one may impose the constraint to the model. SVM applies the soft margin (Vapnik, 1995) and KRR (Shawe-Taylor and Cristianini, 2004) includes the ridge parameter as the added bias. From the Bayesian perspective, the constraint can be imposed using a prior probability distribution. Tipping (2001) proposed a Gaussian prior over $\mathbf{w}$ with zero-mean:

$$p(\mathbf{w}|\mathbf{q}) = \prod_{i=0}^{n} N(w_i|0, \frac{1}{q_i}), \tag{8.10}$$

where each q_i is a hyperparameter associated with w_i. These hyperparameters have the key contribution to the sparsity of RVM. The Gamma priors (Berger, 1985) are used over the parameters $\mathbf{q}$ and σ^2 since they can be treated as the scale parameters:

$$p(\mathbf{q}) = \prod_{i=0}^{n} Gamma(q_i|a, b), \tag{8.11}$$

$$p(\beta) = Gamma(\beta|c, d), \tag{8.12}$$

where $\beta = \sigma^{-2}$ and

$$Gamma(q_i|a, b) = \Gamma(a)^{-1} b^a q_i^{a-1} e^{-bq_i}. \tag{8.13}$$

In the following analysis, we let $a = b = c = d = 0$ such that the above scale priors will be uniform, thus non-informative. The detailed discussion of the more general priors is covered in (Tipping, 2001).

Given the priors defined above, the posterior over the unknown parameters is given by

$$p(\mathbf{w}, \mathbf{q}, \sigma^2|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w}, \mathbf{q}, \sigma^2) p(\mathbf{w}, \mathbf{q}, \sigma^2)}{p(\mathbf{y})}. \tag{8.14}$$

Given the observed $\mathbf{y}$, the unknown new observation $\tilde{y}$ can be estimated from the same process. The distribution of $\tilde{y}$, called the posterior predictive distribution, has the form of (Gelman et al., 2003):

$$p(\tilde{y}|\mathbf{y}) = \int p(\tilde{y}|\mathbf{w}, \mathbf{q}, \sigma^2) p(\mathbf{w}, \mathbf{q}, \sigma^2|\mathbf{y}) d\mathbf{w} d\mathbf{q} d\sigma^2. \tag{8.15}$$

The posterior $p(\mathbf{w}, \mathbf{q}, \sigma^2|\mathbf{y})$ can not be computed analytically. However, it can be decomposed into

$$p(\mathbf{w}, \mathbf{q}, \sigma^2|\mathbf{y}) = p(\mathbf{w}|\mathbf{t}, \mathbf{q}, \sigma^2) p(\mathbf{q}, \sigma^2|\mathbf{y}), \tag{8.16}$$

in which the posterior of the weights has the form defined as

$$\begin{aligned} p(\mathbf{w}|\mathbf{y}, \mathbf{q}, \sigma^2) &= \frac{p(\mathbf{y}|\mathbf{w}, \sigma^2) p(\mathbf{w}|\mathbf{q})}{p(\mathbf{y}|\mathbf{q}, \sigma^2)} & (8.17) \\ &= (2\pi)^{-} \frac{(n+1)}{2} |\mathbf{\Sigma}|^{-0.5} \exp\left\{ -\frac{1}{2} (\mathbf{w} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{w} - \boldsymbol{\mu}) \right\}, & (8.18) \end{aligned}$$

whose mean and covariance are given by

$$\begin{aligned} \mathbf{\Sigma_W} &= (\sigma^{-2} \mathbf{\Psi^T \Psi} + \mathbf{Q})^{-1} & (8.19) \\ \boldsymbol{\mu_W} &= \sigma^{-2} \mathbf{\Sigma \Psi^T} \boldsymbol{y} & (8.20) \end{aligned}$$

respectively, where $\mathbf{Q} = diag(\mathbf{q})$. The marginal likelihood, also known as the "evidence for the hyperparameters" (MacKay, 1992a), has the form of

$$\begin{aligned} p(\mathbf{y}|\mathbf{q}, \sigma^2) &= \int p(\mathbf{y}|\mathbf{w}, \sigma^2) p(\mathbf{w}|\mathbf{q}) d\mathbf{w} & (8.21) \\ &= (2\pi)^{-\frac{n}{2}} |\sigma^2 \mathbf{I} + \mathbf{\Psi Q}^{-1} \mathbf{\Psi^T}|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \mathbf{y}^T (\sigma^2 \mathbf{I} + \mathbf{\Psi Q}^{-1} \mathbf{\Psi^T})^{-1} \mathbf{y} \right\} & (8.22) \end{aligned}$$

The values of the parameters $\mathbf{q}$ and σ^2 that maximize the above marginal likelihood can not be found analytically. The expectation-maximization (EM) algorithm (Hartley, 1958; Dempster et al., 1977) is applied to estimate the parameters iteratively, treating $\mathbf{w}$ as the hidden variables (or called latent variables). In the expectation step of the j^{th} iteration, the mean and the covariance of the weights $\mathbf{w}$ can be computed (or initialized if $j = 1$) from (8.20) and (8.19) using $\mathbf{Q}$ and σ^2 estimated from the previous iteration. In the maximization

step of the j^{th} iteration, following MacKay's work (MacKay, 1992a), let the first derivative of (8.22) respecting to $\mathbf{q}$ be equal to zero. There is

$$q_i^{(j+1)} = \frac{\gamma_i^{(j)}}{\mu_i^{2(j)}}, \tag{8.23}$$

where μ_i is the i^{th} posterior mean weight given by (8.20) and $\gamma_i^{(j)} \in [0, 1]$ is defined as

$$\gamma_i^{(j)} = 1 - q_i^{(j)} \Sigma_{ii}^{(j)}, \tag{8.24}$$

where $\Sigma_{ii}^{(j)}$ is the i^{th} diagonal element of $\mathbf{\Sigma}^{(j)}$, the covariance of the posterior $\mathbf{w}$. γ_i is interpreted as how well the weight w_i fits the data (MacKay, 1992a). Similarly, let the first derivative with respect to σ^2 to be zero. This leads to

$$(\sigma^2)^{(j+1)} = \frac{\|\mathbf{y} - \mathbf{\Psi}\boldsymbol{\mu}^{(j)}\|^2}{n - \Sigma_{i=0}^{n} \gamma_i^{(j)}}. \tag{8.25}$$

At the convergence of the EM algorithm, the final estimates $\mathbf{q}^*$, $(\sigma^2)^*$ and the corresponding $\mathbf{\Sigma^*}$, $\boldsymbol{\mu^*}$ can be used for the prediction of the future observation $\tilde{y}$ since

$$p(\tilde{y}|\mathbf{y}, \mathbf{q}^*, (\sigma^2)^*) = N(\tilde{y}|\mu_{\tilde{y}}, \sigma_{\tilde{y}}^2), \tag{8.26}$$

where

$$\mu_{\tilde{y}} = (\boldsymbol{\mu}^*)^T \psi(\tilde{\mathbf{x}}) \tag{8.27}$$

$$\sigma_{\tilde{y}}^2 = (\sigma^2)^* + \psi(\tilde{\mathbf{x}})^T \mathbf{\Sigma}^* \psi(\tilde{\mathbf{x}}), \tag{8.28}$$

and where the first term of $\sigma_{\tilde{y}}^2$ is the estimated noise of the data and the second term is the uncertainty due to the estimate of $\mathbf{w}$.

In practice, many hyperparameters q_i tend to be infinity, which leads to zero weights w_i. The resulting model will only need a few "relevance vectors", or training observations. The sparsity of RVM is realized because of this.

8.3.2 RVM for Logistic Regression

The above procedure for RVM can be easily extended to the logistic regression for the two-class classification applications. In the logistic regression, the response y_i, which is a Bernoulli random variable, takes the value of 0 or 1, each of which stands for a group label. The general form of the logistic regression model is

$$y_i = E(y_i) + \varepsilon_i. \tag{8.29}$$

where the expectation $E(y_i) = \pi_i$ is the probability that an observation belongs to Group "1". The purpose of the modeling is to predict the posterior probability of π. The probability of observing a specific observation y_i is

$$f_i(y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}. \tag{8.30}$$

Therefore, the likelihood function is given by

$$P(\mathbf{y}|\mathbf{w}) \quad = \quad \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}. \tag{8.31}$$

The logistic response function is applied to relate π with $\psi(\mathbf{x_i})$. It has the form of

$$E(y_i) = \pi_i = \frac{1}{1+\exp(-\psi(\mathbf{x_i})\mathbf{w})}. \tag{8.32}$$

Let $\eta_i = \psi(\mathbf{x_i})\mathbf{w}$, then

$$\eta_i = \ln \frac{\pi_i}{1-\pi_i} \tag{8.33}$$

is called the logistic link function.

The weights $\mathbf{w}$ can not be integrated out or calculated analytically to obtain the marginal likelihood $p(\mathbf{y}|\mathbf{q})$. Therefore, the Laplace's method shown below is utilized to approximate $\mathbf{w}$ iteratively (MacKay, 1992b).

1. Given the estimated hyperparameters $\mathbf{q}$ from the last iteration, find the 'most probable' weights $\mathbf{w}_{MP}$, which gives the mode of the posterior distribution $p(\mathbf{w}|\mathbf{y}, \mathbf{q})$. This is equivalent to finding the $\mathbf{w}$ that maximize the following penalized logistic model since $p(\mathbf{w}|\mathbf{y}, \mathbf{q}) \propto p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\mathbf{q})$:

$$\log \{p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\mathbf{q})\} = \sum_{i=1}^{n} [y_i \log \pi_i + (1 - y_i) \log(1 - \pi_i)] - \frac{1}{2}\mathbf{w^T Q w} \tag{8.34}$$

   The iteratively reweighted least squares (IRLS) method (Nabney, 1999) is applied for the efficient optimization.

2. Using the Laplace's method as the quadratic approximation to the log-posterior at its mode. The second derivative of (8.34), respecting to $\mathbf{w}$ at its mode, is given by

$$\frac{\partial^2 \log p(\mathbf{w}|\mathbf{y}, \mathbf{q})}{\partial \mathbf{w}^2}|_{\mathbf{w}_{MP}} = -(\mathbf{\Psi}^T \mathbf{B} \mathbf{\Psi} + \mathbf{Q}) \tag{8.35}$$

   where $\mathbf{B}$ is a diagonal matrix with $b_i = \pi_i(1 - \pi_i)$.

3. The posterior $p(\mathbf{w}|\mathbf{y}, \mathbf{q})$ is approximated by $N(\mathbf{w}_{MP}, \mathbf{\Sigma})$, where

$$\mathbf{\Sigma}_{MP} = (\mathbf{\Psi}^T \mathbf{B} \mathbf{\Psi} + \mathbf{Q})^{-1} \tag{8.36}$$

$$\mathbf{w}_{MP} = \mathbf{\Sigma}_{MP} \mathbf{\Psi}^T \mathbf{B} \mathbf{y} \tag{8.37}$$

   These two statistics then are used to update the hyperparameter $\mathbf{q}$ according to (8.23). The iteration is repeated until the convergence criterion is satisfied.

After estimating the weights $\mathbf{w}$, (8.32) is used for the prediction of the response.

8.4 Model Selection Using ICOMP

In the previous discussion, we have assumed a fixed kernel function. When fitting a RVR model to the data, one must choose an appropriate kernel function and its corresponding parameters to provide the good model generalization. Furthermore, selecting the appropriate subset of the independent variables improves both the predictive performance and the generalization. Currently, cross-validation is the widely used model selection criterion. In this chapter, we use ICOMP as an alternative model selection criterion when the computational resources or the number of training observations are limited.

The general form of ICOMP for a univariate model is defined by

$$\text{ICOMP} = -2\text{log}L(\hat{\boldsymbol{\theta}}) + 2\text{C}[\widehat{Cov}(\hat{\boldsymbol{\theta}})], \tag{8.38}$$

where C[·] represents a real-valued complexity of the model and $\widehat{Cov}(\hat{\boldsymbol{\theta}})$ represents the estimated covariance matrix of the parameter estimators of the model. In RVM models, $\widehat{Cov}(\hat{\boldsymbol{\theta}})$ is given by (8.19), the covariances of $\mathbf{w}$. We use the marginal likelihood (8.22) as the measure of the fitting. There are several variants of ICOMP applied in different applications, which use different multipliers to the complexity terms or add extra penalty term (Bozdogan, 2007). In this chapter, we use the posterior expected utility (PEU) version of ICOMP Bozdogan (2006, 2007) defined by

$$ICOMP_{PEU} = -2\log\left[p(\mathbf{y}|\mathbf{q},\boldsymbol{\sigma^2})\right] + k + 2C(\boldsymbol{\Sigma}_{\mathbf{w}}), \tag{8.39}$$

where k is the number of non-zero coefficients w_i. Two forms of complexity measures are compared in this paper. The original complexity measure C_1 (Bozdogan, 1990) is defined by

$$\text{C}_1(\boldsymbol{\Sigma}_{\mathbf{W}}) = \frac{k}{2}\text{log}\left(\frac{\bar{\lambda}_a}{\bar{\lambda}_g}\right), \tag{8.40}$$

where $\bar{\lambda}_a$ and $\bar{\lambda}_g$ are the arithmetic mean and the geometric mean of the eigenvalues of $\boldsymbol{\Sigma}_{\mathbf{W}}$. A quadratic equivalent measure of complexity using the Frobenius norm is given by (Bao

and Bozdogan, 2004)

$$\mathrm{C_{1F}}(\mathbf{\Sigma_W}) = \frac{1}{4\bar{\lambda}_a^2}\sum_{i=1}^{k}(\lambda_i - \bar{\lambda}_a)^2, \tag{8.41}$$

where λ_i stands for the i^{th} eigenvalues of $\mathbf{\Sigma_W}$.

The above ICOMP forms are used as the universal criteria to select the kernel function and the subset model. ICOMP is also used to compute the fitness function in GA when performing the subset selection. The interested readers may find the more detailed information of ICOMP from (Bozdogan, 2004a).

8.5 Numerical Results

In this section, we first use the sinc function to demonstrate the selection of kernel function and parameters using ICOMP. Then, we demonstrate the subset selection using the simulated Friedman's multiple-regressor data. We also apply our subset selection technique to the famous benchmark data - Boston Housing data. We apply ICOMP to the Ripley's binary classification application in our last demonstration.

8.5.1 Relevence Vector Regression: the 'sinc' function

The popular sinc function has been widely used as the simulation data in the nonlinear machine learning area. The sinc function is defined as:

$$y = sinc(x) = \frac{\sin(x)}{x}. \tag{8.42}$$

100 observations were generated within the x range of [-10, 10] as the training data. The Gaussian noise $N(0, 0.1^2)$ has been added to the response. Additional 1000 noisy observations are generated as the testing data. ICOMP is used as the model selection criterion to choose the optimal kernel function and its corresponding parameters. The mean squared error (MSE) of the testing data is calculated to evaluate the predictive ability of the selected RVR model.

Table 8.1: RVR Sinc Function: Gaussian RBF Kernel

γ	CV	AIC	ICOMPC1	ICOMPC1F	MSEtrain	MSEtest	# RVs
0.1	0.02195	-55.9	-86.9	1095.6	0.00000	0.02262	100
0.2	0.02017	-127.6	-127.6	225.2	0.00000	0.02064	99
0.3	0.02053	79.7	420.1	1688.2	0.00000	0.02097	100
0.4	0.01495	-276.3	-289.5	-264.3	0.00509	0.01471	26
0.5	0.01449	-280.1	-294.1	-288.0	0.00661	0.01402	23
0.6	0.01361	-293.4	-307.1	-306.4	0.00730	0.01300	17
0.7	0.01325	-299.8	-309.6	-305.0	0.00778	0.01265	15
0.8	0.01300	-302.1	-306.6	-286.9	0.00793	0.01261	15
0.9	0.01272	-306.9	-310.0	-305.6	0.00824	0.01250	13
1	0.01243	-305.5	-309.2	-295.3	0.00831	0.01230	12
2	0.01121	-319.5	-324.2	-324.2	0.00970	0.01101	6
3	0.01190	-312.9	-316.0	-316.1	0.01069	0.01106	5
4	0.01176	-298.6	-291.8	-294.2	0.01067	0.01103	6
5	0.01167	-274.1	-255.2	-264.2	0.01080	0.01126	7
6	0.01276	-254.7	-223.0	-241.9	0.01173	0.01255	7

We first use the popular Gaussian RBF kernel to illustrate how ICOMP chooses the parameters of a kernel function. According to the definition $k(\mathbf{x}, \mathbf{z}) = \exp(-\gamma^{-2}\|\mathbf{x} - \mathbf{z}\|^2)$, γ is a scale parameter that controls the model generalization. The optimal γ value is chosen from the range of [0.1, 6]. The results of a single simulation have been summarized in Table 8.1. Four parameter selection criteria are compared including cross-validation (1000 validation observations), AIC, ICOMP(C_1) and ICOMP(C_{1F}). In a single simulation, all four criteria chose $\gamma = 2$ as the optimal parameter with only 6 RVs needed (Figure 8.1). In this simulation, $\gamma = 2$ is also corresponding to the minimum testing error of 0.01101. We are not surprised that cross-validation provides the good generalization because 1000 validation observations almost represent the whole population. It is glad to see that ICOMP gave the same result without using any validation data.

Next, we compare different kernel functions assuming we do not know that Gaussian RBF kernel is the appropriate kernel function. Again, ICOMP is applied as the model selection criterion to choose kernel functions as well as the parameters for the given kernel function. The model selection results of a single simulation are summarized in Table 8.2

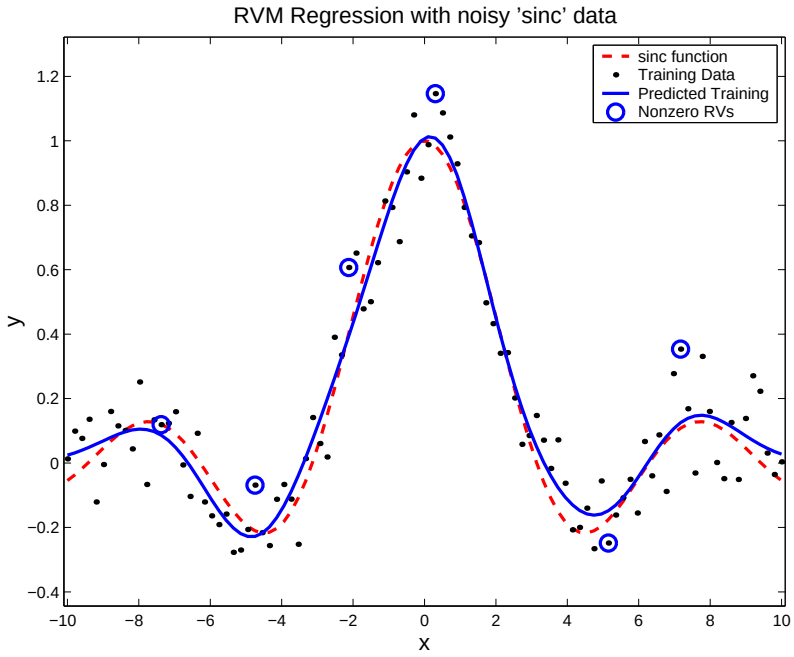


Figure 8.1: Simulated Sinc Data: RVM using Gaussian RBF Kernel

and Table 8.3. Both ICOMP forms chose the Gaussian RBF kernel as the optimal kernel function. The testing data MSE of 0.01101 is low enough since the estimated noise standard deviation is 0.102, which is very close to the true value of 0.01. The normal probability plot also confirms the normality of the estimated noise (Figure 8.2). Now, we repeat the simulation 100 times. At each simulation, ICOMP is used as the model selection criterion to choose the parameters given the kernel function. The testing errors of the different kernel functions are summarized in Table 8.4 and Table 8.5. In addition, we summarized the frequency of being the best kernel (selected using ICOMP) for each kernel out of the 100 simulations. We see the Gaussian RBF kernel has been chosen 70% of the time and is the most preferred kernel by both ICOMP forms. The average testing data MSE of the Gaussian RBF kernel is also among the lowest. The side-by-side boxplots (Figure 8.3) indicate that the Gaussian RBF kernel is consistently better in terms of the testing data MSE. Therefore, we conclude that both ICOMPC$_1$ and ICOMPC$_{1F}$ work well on choosing the appropriate kernel function and parameters for the sinc function data.

Table 8.2: RVR Sinc Function: Comparing Kernel Functions (single run, ICOMP$_{C1}$)

Kernel	Parameter	ICOMP$_{C1}$	MSEtrain	MSEtest	Est. σ	Vectors
Linear Kernel	$< \mathbf{x}, \mathbf{z} >$	-101.4	0.12675	0.13444	0.358	1
Gaussian RBF	$\gamma = 2$	-324.2	0.00970	0.01101	0.102	6
Linear Spline		-230.8	0.01015	0.01027	0.105	8
Exp RBF	a = 0.3	-139.5	2.67e-16	0.01913	0.000	100
Cauchy	a = 0.9	-322.2	0.00846	0.01243	0.096	8
Sigmoid	$a = 0.06$ $b = 1$	-290.6	0.01200	0.01212	0.112	5
TP Spline	a = 0.002	-273.0	0.00929	0.01108	0.100	8
Cubic	a = 0.03	-250.2	0.01041	0.01097	0.105	6
Bubble	a = 0.3	-291.8	0.00283	0.01698	0.063	36
B-Spline	N = 1	-307.8	0.00845	0.01329	0.097	13
ANOVA Spline 1		-230.8	0.01015	0.01073	0.105	8
ANOVA B-Spline	N = 2	-319.1	0.00856	0.01259	0.097	10

Table 8.3: RVR Sinc Function: Comparing Kernel Functions (single run, ICOMP$_{C1F}$)

Kernel	Parameter	ICOMP$_{C1F}$	MSEtrain	MSEtest	Est. σ	Vectors
Linear Kernel	$< \mathbf{x}, \mathbf{z} >$	-101.4	0.12675	0.13444	0.358	1
Gaussian RBF	$\gamma = 2$	-324.2	0.00970	0.01101	0.102	6
Linear Spline		-260.2	0.01015	0.01027	0.105	8
Exp RBF	a = 0.7	-218.2	2.97e-11	0.01819	0.000	100
Cauchy	a = 0.9	-321.4	0.00846	0.01243	0.096	8
Sigmoid	$a = 0.06$ $b = 1$	-294.9	0.01200	0.01212	0.112	5
TP Spline	a = 0.002	-289.7	0.00929	0.01108	0.100	8
Cubic	a = 0.08	-270.3	0.01035	0.01104	0.105	7
Bubble	a = 0.01	-260.9	4.41e-8	0.02163	0.002	99
B-Spline	N = 1	-304.4	0.00845	0.01329	0.097	13
ANOVA Spline 1		-260.2	0.01015	0.01073	0.105	8
ANOVA B-Spline	N = 2	-316.6	0.00856	0.01259	0.097	10

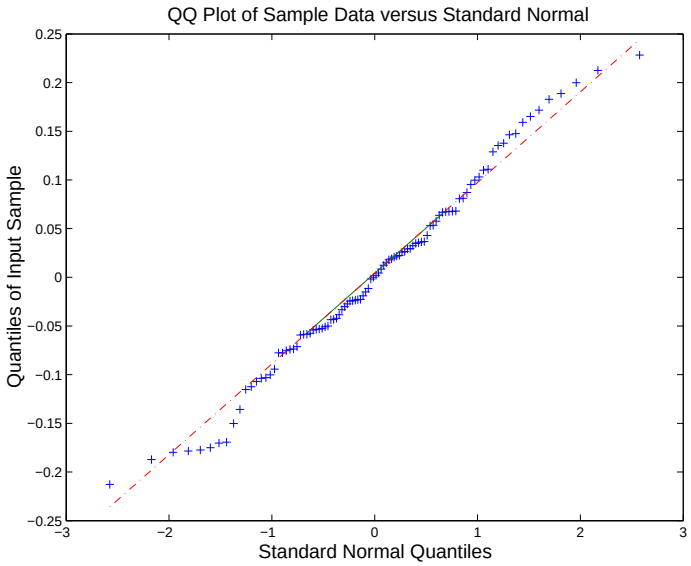


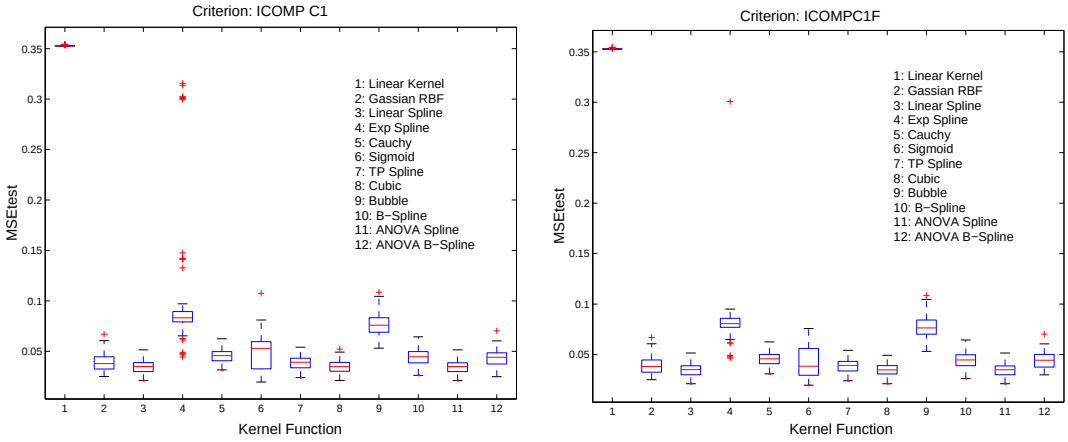
Figure 8.2: Simulated Sinc Data: Normal Probability Plot

Table 8.4: RVR Sinc Function: Comparing Kernel Functions (100 runs, ICOMPC_1)

Kernel	Ave. MSEtest	Std. MSEtest	Ave. # Vectors	Freq.
Linear Kernel	0.13416	0.0023	1	0
Gaussian RBF	0.01160	0.0010	7.4	70
Linear Spine	0.01122	0.0007	7.3	0
Exp RBF	0.01654	0.0015	93.7	0
Cauchy	0.01208	0.0008	10.0	4
Sigmoid	0.01261	0.0019	7.2	0
TP Spline	0.01152	0.0008	8	0
Cubic	0.01125	0.0007	6.4	0
Bubble	0.01596	0.0019	31	3
B-Spline	0.01203	0.0009	10.3	10
ANOVA Spline	0.01122	0.0007	7.3	0
ANOVA B-Spline	0.01195	0.0009	10.0	13

Table 8.5: RVR Sinc Function: Comparing Kernel Functions (100 runs, ICOMPC_{1F})

Kernel	Ave. MSEtest	Std. MSEtest	Ave. # Vectors	Freq.
Linear Kernel	0.13416	0.0023	1	0
Gaussian RBF	0.01160	0.0010	7.4	70
Linear Spine	0.01122	0.0007	7.3	0
Exp RBF	0.01603	0.0009	93.7	0
Cauchy	0.01209	0.0008	10.1	4
Sigmoid	0.01191	0.0013	6.5	0
TP Spline	0.01152	0.0008	8	0
Cubic	0.01123	0.0007	6.8	0
Bubble	0.01607	0.0019	31.9	3
B-Spline	0.01204	0.0009	10.4	10
ANOVA Spline	0.01122	0.0007	7.3	0
ANOVA B-Spline	0.01199	0.0009	10.1	13



(a) (b)

Figure 8.3: Compare Kernel Functions using ICOMP (a) ICOMPC1 (b) ICOMPC1F

Table 8.6: RVR Friedman: Gaussian RBF (100 runs)

ICOMP	Ave. MSEtest	Std. MSEtest	Ave. # Vectors
ICOMP(C_1)	3.06	0.50	44.9
ICOMP(C_{1F})	3.21	0.50	41.1

8.5.2 RVR: Friedman

We now move to the regression problems with multiple regressors. Generally, not all the regressors are independent to each other. Besides choosing the optimal kernel function and parameters, one might be interested in choosing the best subset regressors. The subset regressor selection will help with the model generalization and the practical interpretation.

In this numerical experiment, we applied Friedman's (Friedman, 1991; Tipping, 2001) simulation function. The true function is defined as:

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \sum_{i=6}^{10} 0 \times x_i. \tag{8.43}$$

The 10 independent variables $x_1 - x_{10}$ are randomly generated from the unit hypercube. Variable $x_6 - x_{10}$ have no contribution to the response y. The Gaussian noise $N(0, 1^2)$ is added to the response. We applied the widely used Gaussian RBF kernel to this data. ICOMP is utilized to choose the optimal scale parameter of the kernel function. We first include all the 10 regressors. The average MSE for the test data is around 3 (Table 8.6). Then, we compare all the $2^{10} - 1 = 1023$ subset models (include at least one regressor) in terms of the ICOMP score given the selected optimal kernel. The optimal subset model is the minimizer of ICOMP. We hope the selected model excludes $x_6 - x_{10}$.

The result of the simulation (Table 8.7) indicates the optimal subsets contain $x_1 - x_5$ and the best subset model selected by ICOMP contains $x_1 - x_5$ only. The nuisance variables $x_6 - x_{10}$, as expected, are not included. The testing error of this subset model is 0.38. It is a tremendous improvement compared with 2.86 of the saturated model. Only 30 RVs are needed in stead of 50 for the saturated model, which leads to the better sparsity. This result

Table 8.7: RVR Friedman: Subset Selection (1 run) Gaussian RBF Kernel $\gamma = 2$

Full Model	ICOMPC1	MSEtrain	MSEtest	#Vectors
12345678910	-392.0492	1.01	2.86	55
Best Subsets	ICOMPC1	MSEtrain	MSEtest	#Vectors
12345	-559.3	0.25	0.38	30
1234569	-541.3	0.79	1.93	31
123459	-510.2	0.43	0.85	33
12345689	-503.8	0.91	2.42	34
Full Model	ICOMPC1F	MSEtrain	MSEtest	#Vectors
12345678910	-429.4	1.01	2.86	55
Best Subsets	ICOMPC1F	MSEtrain	MSEtest	# Vectors
12345	-650.5	0.25	0.38	30
1234569	-564.9	0.79	1.93	31
123459	-553.2	0.43	0.85	33
123457	-534.8	0.48	0.85	34
123456	-529.5	0.39	0.82	33
1234510	-527.9	0.42	0.77	37

is also comparable to Tipping's η-RVM (Tipping, 2001). The η-RVM algorithm improves RVM by assigning and estimating additional parameters η_i to each variables respectively. However, as commented by Tipping, estimating η_i is conducted by a gradient-based method. The computational intensity of the estimation is the major disadvantage of η-RVM. Applying the subset selection overcomes this problem. Even if the number of variables is relative big, where the all-possible-subset-selection (APSS) is practically impossible or inefficient, GA can be applied to find the optimal subsets and save the computation time dramatically.

It may not be necessary to apply GA for this 10-variable subset selection problem since there are only 1023 subsets. However, we apply the GA subset selection for the demonstration purpose. The GA parameters configured for this example is summarized in Table 8.8. We force the best subset of the current generation goes to the next generation without crossover or mutation. The best subset of the initial population is $\{x_2, x_4, x_5, x_{10}\}$. GA found the best subset since the 5^{th} generation (Table 8.9). In this experiment, only 500 subsets (not distinct) are evaluated. Practically, one may want to use more generations (for instance 100) to reach the convergence. One may argue that will require more computation

Table 8.8: GA Parameters for Friedman Data

n_{gen}	10
n_{pop}	50
P_{cross}	0.9
$P_{mutation}$	0.01
Elitism	YES
Type of crossover	Uniform

Table 8.9: RVR Friedman Data: GA for Subset Selection

Generation	Best Subset	ICOMPC1
1	2 4 5 10	-415.8
2	2 4 5	-415.9
3	1 2 4 5 6 9	-438.8
4	1 2 4 5 6 9	-438.8
5	1 2 3 4 5	-559.3
6	1 2 3 4 5	-559.3
7	1 2 3 4 5	-559.3
8	1 2 3 4 5	-559.3
9	1 2 3 4 5	-559.3
10	1 2 3 4 5	-559.3

then APSS. It is true for this 10-variable model. However, when the number of variables increases, the total number of subsets increases very fast. When using GA, it is not necessary to further increase the population size or the generation number. From this point of view, GA is efficient for the scenarios with large number of variables. To confirm the robustness of the GA solutions, we repeat the GA procedure 100 times with a random initial population for each run. It is concluded that the true model is selected 56 times. There are 38 times that the selected models include the true model and 2 to 3 nuisance variables. We think this result is acceptable.

Table 8.10: RVR Bonston Housing: Gaussian RBF Kerenel (1 run)

γ	ICOMPC1	LOF	2C1	MSEtrain	MSEvali	MSEtest	#W
0.1	-374.7	-1074.3	298.7	1.57E-16	79.7223916	70.39	401
0.2	171.8	-1236.8	1007.6	1.56E-14	59.8575188	35.63	401
0.3	951.2	-1272.6	1822.8	1.00E-11	44.5588249	19.63	401
0.4	1715.6	-1093.0	2409.7	5.39E-07	37.6876285	23.51	399
0.5	-702.0	-1322.7	439.7	0.93	21.9008714	5.32	181
0.6	-627.5	-1284.6	500.1	1.23	14.8489666	6.31	157
0.7	-648.0	-1302.5	505.5	1.20	11.8526334	6.36	149
0.8	-685.2	-1275.1	456.8	1.36	9.71695417	5.06	133
0.9	-626.8	-1266.0	521.1	1.69	8.97113256	4.54	118
1	-609.7	-1253.6	533.9	1.78	8.44486925	3.96	110
2	-525.1	-1097.3	494.2	2.67	10.5143119	3.22	78
3	4103.5	-919.1	4630.6	2.71	10.4656171	3.82	392
4	-900.9	-1060.6	130.6	7.48	12.6226372	8.19	29
5	-898.7	-1053.9	129.2	7.68	14.1614114	9.17	26
6	1717.1	-1022.3	2565.3	7.38	13.6352305	8.31	174

8.5.3 RVR: Boston Housing Data

In this experiment, we apply RVR to the popular benchmark - Boston Housing. Originally published by Harrison and Rubinfeld (Harrison and Rubinfeld, 1978), the Boston Housing data set includes 13 environmental and social factors that are believed to be relevant to the median value of owner-occupied homes (Appendix A). In stead of transforming the original variables as done by Harrison, a nonlinear modeling technique can be applied on the original variables. Therefore, the Boston Housing data set has been widely used as the benchmark of the kernel-based methods.

The total of 506 observations were randomly split into 406 training observations, 80 validation observations and 25 testing observations. The validation data is needed only when the 2-fold cross-validation is utilized to choose the optimal kernel function. All the 14 variables are linearly scaled to $[-1, 1]$. Following the current literature, the Gaussian RBF kernel is applied. Again, ICOMP is utilized to choose the optimal scale parameter of the saturated model. The average MSE and the standard deviation of MSE of the testing data are evaluated.

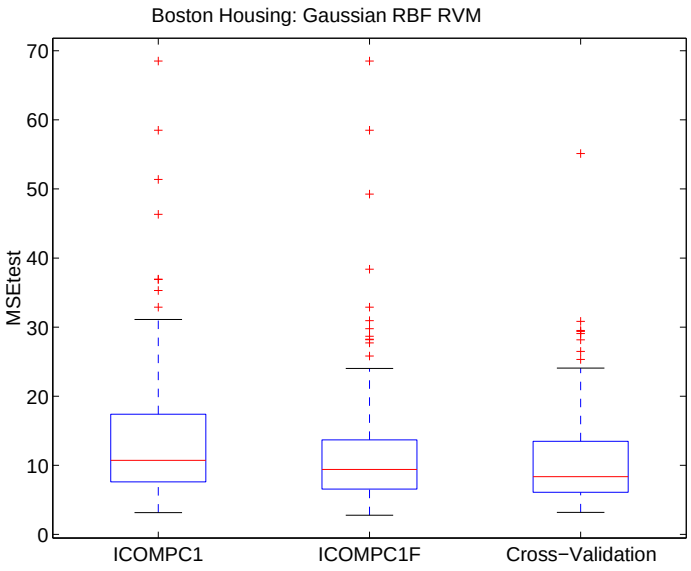


Figure 8.4: Boston Housing: 100 Simulations

In a single simulation (Table 8.10), ICOMPC1 chose $\gamma = 4$ as the optimal scale parameter corresponding to the testing error of 8.19. If the cross-validation method was used, one should have chosen $\gamma = 1$, which is the minimizer of the validation data error. The testing data MSE of this model is 3.96 with 110 RVs. However, one can see this is not likely to be the optimal model because the average number of RVs is 39 and the average testing data error is 7.46 based on the 5-fold cross-validation results (Tipping, 2001). This implies that this model happened to be good for the given testing data and may not be general enough for the other data. ICOMPC1 did not choose $\gamma = 1$, 2 or 3 where the testing errors are tremendously smaller because those models have much higher complexity due to the redundant RVs.

We repeat the simulation 100 times (Figure 8.4). It is shown that ICOMPC1F give the comparable testing testing data MSE.

Besides selecting the right scale parameter of the Gaussian RBF kernel, one may be interested in finding the factors, among the 13 potential factors, that have crucial contribution to the median housing price. We may conduct the all possible subset (8191 subsets)

Table 8.11: RVR Boston Housing: All-Possible-Subset Selection

	Model	ICOMPC1	MSEtrain	MSEtest	# w
Best	1 5 6 7 9 11 12 13	-971.3	9.43	12.55	16
13 vars.	1 2 3 4 5 6 7 8 9 10 11 12 13	-900.9	7.48	8.19	29
12 vars.	1 2 3 4 5 6 7 8 10 11 12 13	-889.5	7.63	8.16	30
11 vars.	1 2 4 5 6 7 8 9 11 12 13	-912.8	8.34	9.52	25
10 vars.	1 2 5 6 7 8 9 11 12 13	-953.7	8.56	10.32	22
9 vars.	1 5 6 7 8 9 11 12 13	-955.2	8.63	10.14	20
8 vars.	1 5 6 7 9 11 12 13	-971.3	9.43	12.55	16
7 vars.	1 5 6 7 9 11 13	-966.3	9.89	13.17	14
6 vars.	5 6 7 9 11 13	-950.7	10.73	13.99	13
5 vars.	5 6 9 11 13	-944.9	11.21	19.10	11
4 vars.	6 11 12 13	-895.6	14.19	23.67	8
3 vars.	5 6 11	-822.0	19.10	47.15	6
2 vars.	6 7	-689.6	27.69	40.33	4
1 vars.	10	-418.3	61.81	80.06	2

selection using ICOMP as the model evaluation criterion. Alternative, one may use the genetic algorithm (GA) to find the optimal (may not be the best) subset more efficiently. We will demonstrate both to the Boston Housing data. GA is extremely useful if the number of variables is relative big.

The results of the all-possible-subset selection using ICOMPC1 as the criterion are summarized in Table 8.11. The best subset model excludes ZN, INDUS, CHAS, DIS and TAX. It offers the better sparsity (16 RVs) but relative higher testing error (12.55) compared with the saturated model. We see that the 12-variable-model or the 13-variable-model is the best in terms of the predictive performance for the 25 testing observations in this experiment. However, there are correlated variables that bring the redundant information. That is the reason that ICOMP did not choose these two complex models. We believe that variable selection will help us understand the contribution of individual variables thus give us better practical interpretation. Therefore, we also present the best subset for each particular size.

In the 12-variable-model, RAD is excluded. RAD is an accessability variable that should have positive impact on the housing values. It is shown that RAD and TAX have very high

positive correlation (r = 0.91). RAD also has relative high positive correlation with other variables including CRIM, INDUS, NOX and DIS. For instance, DIS, also a accessibility variable, can reflect the nearnesss to workplace which RAD can explain. Therefore, RAD is not necessary to be included since its contribution can be provided by some other variables. As we can see, the 12-variable-model actually has slightly smaller testing error. The 11-variable-model further dropped INDUS, the negative environmental influence due to the industry. INDUS is highly correlated (r=0.76) with the accessibility variable DIC and the air pollution variable NOX. Therefore, we are not surprised that INDUS is not in the model. The increasing of the testing error (9.52) is still acceptable. We can compare the variable selection results with some works in the literature. In Fukumizu (2004) and Breiman's (1985) work, RM, LSTAT, PTRATIO and TAX are selected as the variables with the most important contribution. However, the air pollution variable NOX is excluded from their models. Breiman claimed that the convex pattern between NOX (actually NOX^2 in his model) and MV (the median housing value) is difficult for the linear model to pick up. Both Breiman (1985) and Harrison (1978) agreed that NOX has marginal contribution to MV. Our best 5-variable-model includes NOX, RM, RAD, PTRATIO and LSTAT, which agrees well with the previous work. TAX is not included in our model. However, TAX has 0.91 correlation with RAD and 0.66 correlation with NOX. We have reason to believe its contribution has been explained by the other variables in the model.

8.5.4 RVLR: Ripley's Data

The well-known Ripley data set problem consists of two classes where the two-dimensional data for each class have been generated by a mixture of two Gaussian distributions.

In RVLR, we define p as the probability that an observation belongs to Class "1". The classification is done by comparing the probabilities (or the membership). If $p > 0.5$, assign the observation to Class "1", otherwise, to Class "0". 100 training observations are randomly selected from the original Ripley's 250 training data. The testing data includes 1000 observations.

Table 8.12: RVR Ripley's Data: Gaussian RBF Kernel (1 run)

γ	ICOMPC1	ICOMPC1F	Error-train%	Error-test%	#w
0.1	113.7	135.2	6	14	22
0.2	40.1	40.0	9	8.7	7
0.3	22.5	21.6	8	8.1	4
0.4	21.6	21.6	10	8.1	4
0.5	21.4	20.5	11	8.6	4
0.6	22.7	21.5	11	9	4
0.7	26.6	24.6	11	8.8	4
0.8	29.4	26.8	12	9	4
0.9	28.9	26.2	13	9.3	4
1	25.2	23.5	15	9.9	4
2	23.2	18.7	12	10.5	2
3	26.4	20.4	12	10.4	2
4	28.8	21.5	12	10.4	2

We utilize the popular Gaussian RBF kernel. Different scale parameters γ have been compared (Table 8.12). The optimal scale parameter is the minimizer of ICOMP. The optimal scale parameter selected by ICOMP(C_1) is $\gamma = 0.5$, corresponding to the test classification error of 8.6%. Only 4 RVs are needed (Figure 8.5).

We now repeat the simulation 100 times. In each simulation, a random sample of 100 training observations are selected from the original 250 observations. This experiment leads to the average testing data classification error of 9.8% when using ICOMP as the model selection criterion. If we fix the scale parameter to 0.5, as Tipping illustrated in his work Tipping (2001), the average classification error is 9.6%. We would say the performance of both ways are similar.

8.5.5 RVLR: Heart Data

The nuclear magnetic resonance (NMR) imaging has been used to identify fatty tissues in the arteries in order to early detect the cause of heart attack. NMR shows blood flow as well details of the aorta and heart valves. The NMR aorta data applied in this research was collected by Dr. Pearlman (1988) at the Medical School of the University of Virginia.

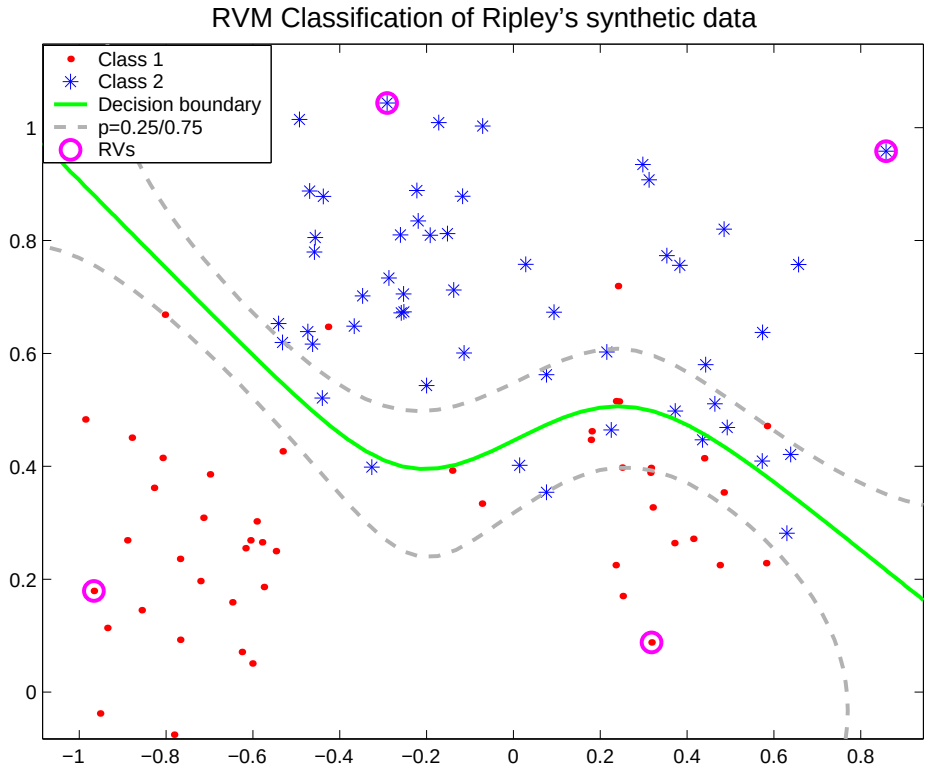


Figure 8.5: Ripley's Data: RVM Classification Using Gaussian RBF Kernel

Data is pooled from 418 patients. There are total of 20 variables including radius, angle, and 16 different image acquisitions variables. The first 194 patients belonged to the class of early atheroma. 20% of the observations in each class are used as the training data. The rest 80% are used as the testing data.

All 20 variables are used first. Different kernel functions, including Linear, Guassian RBF, Cauchy and Cubic, are compared using ICOMPPEUC1F (Table 8.13). It can be concluded that the linear kernel is the optimal with in terms of the ICOMP values. Actually, this data can be easily separated (Figure 8.6). It is not surprised that only two RVs are necessary for the training. A linear classifier is good enough using just Variable 1 and Variable 2. Including all 20 variables may even decrease the classification rate besides increasing the model complexity. We now conduct the subset selection to decide which variables are critical for the classification. The number of subsets is $2^{20} - 1 = 1,048,575$.

Table 8.13: RVRLR: Aorta Data Compare Kernels (Saturated Model)

Kernel	Parameter	ICOMPPEUC1F	# RVs	Training Error	Testing Error
Linear		5.23	2	0	0
Polynomial	Order = 3	7.74	2	0	0
Cubic	a = 0.09	11.81	2	0	0.60
Gaussian	$\gamma = 2$	13.59	3	0	1.50
Cauchy	a = 1.1	14.05	3	0	0.30

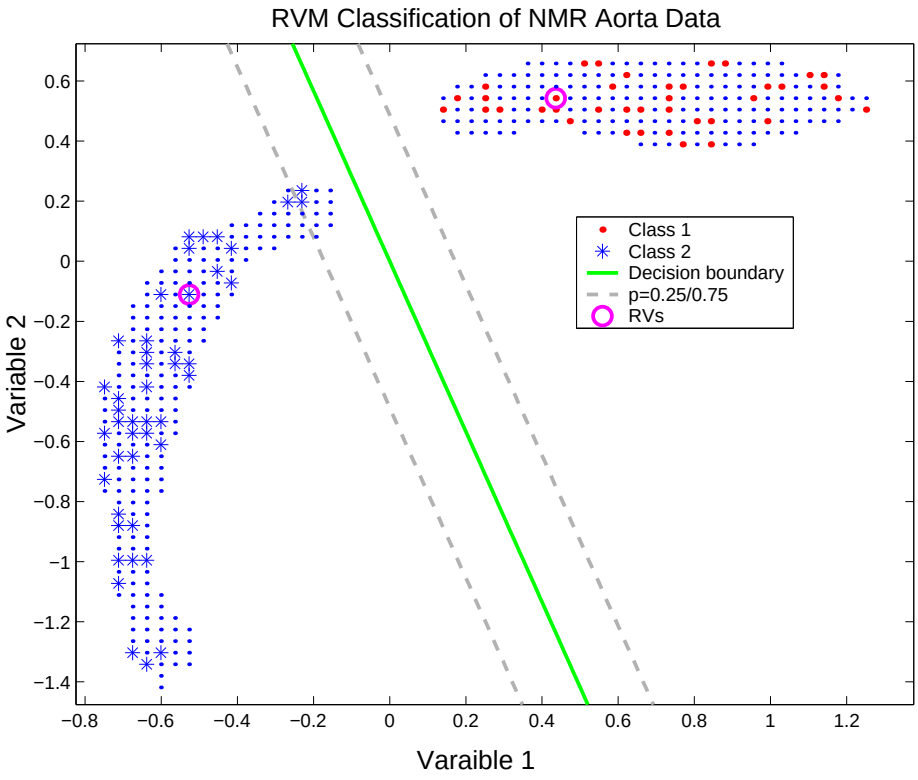


Figure 8.6: Aorta Data: RVM Classification Using Linear Kernel

Table 8.14: RVRLR: Aorta Data Subset Selection

Kernel	Model	ICOMP	# RVs	Train	Test
Linear	1 2 3 5 7 8 10 11 12 13 14 16 17 18 19 20	4.76	2	0	0
Polynomial^3	1 2 4 5 9 13 14 17 18 20	6.54	2	0	0
Cubic	1 2 6 7 10 13 17	7.40	2	0	0
Gaussian	2 3 5 9 10 11 12 13 15 17 18 19	7.93	2	46.43	46.41
Cauchy	1 2 6 7 12 19	9.68	2	0	0

Therefore, we perform a GA procedure to find the optimal subsets efficiently. We use 50 generations for each search with the population size of 50. It can be concluded the best subset still comes from the linear kernel (Table 8.14). The Gaussian RBF kernel is not appropriate for this data whose best subset leads to about 46% classification error. The other kernel functions also have subsets with zero classification error. However, their complexity are higher. The best subsets of the Linear, Polynomial, Cubic and Cauchy kernel all include the first two variables, which can separate two classes easily (Figure 8.6).

Chapter 9

Conclusions and Recommendations of Future Work

9.1 Conclusions

This dissertation extends the using of ICOMP as the model selection criterion to kernel-based regression analysis. This model selection criterion utilizes a simple real-valued information-theoretic measure of complexity to evaluate the goodness-of-fit and the generalization of each candidate model without using the validation data. It is used to choose kernel functions, the parameters of the kernel function, parameters of the regression models, and the best subset of the input variables. Under the circumstance where the number of input variables is large, ICOMP and the Genetic Algorithm are combined to find the optimal subsets efficiently. The regression analysis applications that have been discussed in this dissertation include kernel ridge regression, kernel partial least squares regression, kernel principal component analysis, kernel principal regression, relevance vector machine regression and relevance vector machine logistic regression. Our numerical results indicate that kernel models selected by ICOMP have comparable (sometimes better) predictive performance as those chosen by using cross-validation. However, the decreasing of the computational time is tremendous when ICOMP is used.

For the kernel ridge regression, ICOMP is capable of choosing the ridge parameter, comparing the kernel functions, choosing the kernel parameters given the form of the kernel function, and choosing the optimal subset of independent variables. This dissertation provides the ICOMP forms for KRR using the exact covariance and the asymptotic covariance (using inverse Fisher's information). This dissertation also derives the interval estimates of the kernel ridge regression and the weighted kernel ridge regression. We compared four forms of ICOMP with cross-validation using the simulated sinc function. It is concluded that ICOMPC1, which scores the exact covariances of the estimated regression coefficients, outperforms the others and gives the similar results as LOOCV does. In the repeated simulation experiments, the models chosen by LOOCV is quite different due to the randomly selected validation data for each run. The models selected by ICOMP are more consistent and are generally among a small group of similar candidates. Using ICOMP as the model selection criterion, KRR successfully picked the desired variables of the Friedman's data with 100% correct rate. It is also illustrated using Friedman's data that applying Genetic Algorithm has the similar performance without conducting the all possible subset selection.

Applying the PEU version of ICOMP to KPLS, it is observed that ICOMP successfully chooses the optimal number of latent variables. This dissertation also proposes a regularization method to provide a better estimate of the error variance, which is used in ICOMP to compare different kernel functions. The numerical experiments indicate that ICOMP's performance is very close to that of the cross-validation method. ICOMP also outperforms AIC and SBC in 100 simulations. For the Friedman's data, combining ICOMP and GA, the proposed model selection criterion chooses the desired subset of the independent variables 99% of the time.

Using the multivariate form of ICOMP to KPCA, it is illustrated using the toy example that ICOMP chooses the Gaussian RBF kernel as the optimal kernel function among four candidates with 8 PCs retained which reflects 95% of the total variance. When the Gaussian RBF kernel is utilized, the optimal scale parameter selected by ICOMP is similar as the one

selected by the cross-validation method. This dissertation also demonstrates that ICOMP can be used to choose the kernel function and the number of PCs for KPCR.

This dissertation uses ICOMP as the model selection criterion to compare different kernel functions and choose the optimal subset of the independent variables. The numerical results indicate that ICOMP is as good as the cross-validation method in finding the optimal kernel parameters. ICOMP is computationally more efficient compared with cross-validation. Therefore, it is more efficient to use ICOMP to select kernel functions besides selecting the parameters given the form of a kernel function. The proposed method is successfully applied to both the relevance vector regression and the relevance vector logistic regression for the classification. It is also demonstrated that variable subsetting may increase the predictive ability of the model tremendously. In the current literature, η-RVM is used for the Friedman's data to achieve smaller prediction error. It is very time consuming because η-RVM uses a nonlinear gradient search procedure. This dissertation uses ICOMP as the model selection criterion and GA as the searching technique to choose the optimal subset of the independent variables. The selected model has a comparable prediction error and the searching procedure is very efficient.

9.2 Recommendations of Future Work

The model selection criterion used in this dissertation for the univariate kernel-based regression analysis applications can be extended to the multivariate applications following Bozdogan's ICOMP forms for the linear multivariate regression (Bozdogan, 1990, 2004a).

The ICOMP form for KPCA used in this dissertation depends on the technique of approximating pre-images. The pre-image approximation method applied in this dissertation uses different formulas for different kernel functions. Furthermore, the pre-images of some kernel functions are not available using this approximating method. We hope to develop a measure of lack-of-fit (for computing ICOMP) in the future that does not require the approximating of pre-images.

The Gaussian noise has been assumed in this dissertation. It can be generalized to the non-normal distributions such as Power Exponential (PE) and family of elliptically contoured (EC) error distributions (Liu, 2006).

Bibliography

Bibliography

Aizermann, M., Bravermann, E., and Rozonoer, L. (1964). Theoretical foundations of potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837.

Akaho, S. (2001). A kernel method for canonical correlation analysis. In *International Meeting of Psychometric Society*.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In Petrov, B. and Cski, F., editors, *Second international symposium on information theory*, pages 267–281, Acadmiai Kiad, Budapest.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Akaike, H. (1987). Factor analysis and aic. *Psychometrika*, 52:317–332.

Bao, X. and Bozdogan, H. (2004). Subsetting kernel regression models using genetic algorithm and the information measure of complexity. In *The 2004 BI-ANNUAL MEETING OF THE INTERNATIONAL FEDERATION OF CLASSIFICATION SOCIETIES*, CHICAGO, USA.

Bellman, R. (1961). *Adaptive control processes: A guided tour*. Princeton University Press, Princeton, New Jersey.

Berger, J. O. (1985). *Statistical decision theory and Bayesian analysis*. Springer, 2 edition.

Boltzmann, L. (1877). Uber die beziehung zwischen dem hauptsatze der mechanischen warmetheorie und der wahrscheinlicjkeitsrechnung respective den satzen uber das warmegleichgewicht. *Wiener Berichte*, 76:373–435.

Boyce, D. E., Farhi, A., and Weischedel, R. (1974). *Optimal subset selection: multiple regression, interdependence, and optimal network algorithms.* Springer-Verlag, New York.

Bozdogan, H. (1988). Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Annals of Eugenics*, 52(3):345–370.

Bozdogan, H. (1990). On the information-based measure of covariance complexity and its application to the evaluation of multivariate linear models. *Communications in Statistics, Theory and Methods*, 19:221–278.

Bozdogan, H. (1998). Information complexity criteria for regression models. *Computational Statistics & Data Analysis*, 28:51–76.

Bozdogan, H. (2000). Akaike's information criterion and recent developments in information complexity. *Journal of Mathematical Psychology*, 44:62–91.

Bozdogan, H. (2003). Multivariate statistical modeling.

Bozdogan, H. (2004a). Intelligent statisticial data mining with information complexity and genetic algorithms. In Bozdogan, H., editor, *Statistical Data Mining and Knowledge Discovery*, pages 15–55. CRC Press.

Bozdogan, H. (2004b). Misspecification-resistant model selection using information complexity. Invited pater under review for the Journal of Econometrics.

Bozdogan, H. (2005). Model selection under misspecification using information complexity. Submiited to the Journal of Multivariate Analysis.

Bozdogan, H. (2006). A new generation mixture-model cluster analysis using information complexity and genetic algorithms. Keynote Lecture Presented at the International Workshop on Knowledge Extraction and Modeling.

Bozdogan, H. (2007). Information complexity and multivariate learning in high-dimensional data mining. Forthcoming Book.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *Annals of Statistics*, 24:2350–2380.

Breiman, L. and Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80(391):580–598.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behav. Res.*, 1:245–276.

Cawley, G. C. (2002). Redueced rank kernel ridge regression. *Neural Processing Letters*, 16(3):292–302.

de Jong, S. (1993). Simpls: an alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Engle, H. W., Hanke, M., and Neubauer, A. (2000). *Regularization of inverse problems*. Kluwer Academic Publishers.

Friedman, J. H. (1991). Multivariate adpative regression splines. *Annals of Statistics*, 19(1):1–141.

Fukumizu, K., Bach, F. R., and Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99.

Geladi, P. and Kowalski, B. R. (1986a). An example of 2-block predictive partial least-squares regression with simulated data. *Analytica Chimica Acta*, 185:19–32.

Geladi, P. and Kowalski, B. R. (1986b). Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185:1–17.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis*. Chapman & Hall/CRC, 2 edition.

Gil, J. A. and Romera, R. (1998). On robust partial least squares (pls) methods. *Journal of Chemometrics*, 12:365–378.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, New York.

Good, I. J. and Gaskins, R. A. (1971). Nonparametric roughness penalties for probability densities. *Biometrika*, 58:255–277.

Goutte, C. (1997). Note on free lunches and cross-validation. *Neural Computation*, 9:1211–1215.

Gower, J. (1968). Adding a point to vector diagrams in multivariate analysis. *Biometrika*, 55(3):582–585.

Han, L., Embrechts, M. J., Szymanski, B., Sternikel, K., and Ross, A. (26–28 April 2006). Random forest feature selection with k-pls: Detecting ischemia from magnetocardiograms. In *ESAANN'2006 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium)*, pages 26–28. d-side publi.

Hansen, P. C. (1998). *Rank-deficient and discrete ill-posed problems*. SIAM monographs on mathematical modeling and computation. Philadelphia.

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5:81–102.

Hartley, H. (1958). Maximum likelihood estimation from incomplete data. *Biometrics*, 14:174–194.

Helland, I. S. (1988). On structure of partial least squares regression. *Communications in Statistics - Elements of Simulation and Computation*, 17:581–607.

Hilbert, D. (1927). Über die grundlagen der quantenmechanik. *Mathematische Annalen*, 98:1–30.

Hocking, R. R. (1976). The analysis and selection variables in linear regression. *Biometrics*, 32:1044.

Hocking, R. R. (1983). Delelopments in linear regression methodology: 1959-1982. *Technometrics*, 25:219–230.

Hoerl, A. and Kennard, R. W. (1970a). Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12(1):69–82.

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

Hoerl, A. E. and Kennard, R. W. (1976). Ridge regression: Iterative estimation of the biasing parameter. *Communications in statistics*, A5:77–88.

Hoerl, A. E., Kennard, W., and Baldwin, K. F. (1975). Ridge regression: Some simulations. *Communications in statistics*, 4:105–123.

Höskuldsson, A. (1988). Pls regression methods. *Journal of Chemometrics*, 2:211–228.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.*, 24:417–441, 498–520.

Jolliffe, I. T. (2002). *Principal components analysis*. Springer.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educ. Psychol. Meas.*, 20:141–151.

Kowalski, B., Gerlach, R., and Wold, H. (1982). Chemical systems under indirect observation. In Jöreskog, K. and Wold, H., editors, *Systems under Indirect Observation*, pages 191–209, Amsterdam. North-Holland.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86.

Kwok, J. T. and Tsang, I. W. (2003). The pre-image problem in kernel methods. In *Proceedings of ICML 2003*, pages 408–415.

Leonov, A. S. and Yagola, A. G. (1997). The l-curve method always introduces a nonremovable systematic error. *Moscow University Physics Bulletin*, 52(6):20–23.

Lewi, P. J. (1995). Pattern recognition, reflection from a chemometric point of view. *Chemometrics and Intelligent Laboratory Systems*, 28(1):23–33.

Liu, M. (2006). *Multivariate Nonnormal Regression Models, Information Complexity, and Genetic Algorithms: A Three Way Hybrid for Intelligent Data Mining*. PhD thesis, The University of Tennessee.

MacKay, D. J. C. (1992a). Bayesian interpolation. *Neural Computation*, 4(3):415–447.

MacKay, D. J. C. (1992b). The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736.

Magnus, J. R. and Neudecker, H. (1988). *Matrix differential calculus*. Wiley, New York.

Mallows, C. L. (1973). Some comments on c_p. *Technometrics*, 15(4).

Manne, R. (1987). Analysis of two partial least squares algorithms for multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 2:187–197.

Mantel, N. (1970). Why stepdown procedures in variables selection. *Technometrics*, 12:591–612.

Marquardt, D. W. (1975). Ridge regression is practice. *American Statistician*, 29(1):3–20.

Martens, H. and Naes, T. (1989). *Multivariate Calibration*. John Wiley, New York.

Mehdi, J.-H. and Kyani, A. (2007). Application of genetic algorithm-kernel partial least square as a novel nonlinear feature selection method: Activity of carbonic anhydrase ii inhibitors. *European Journal of Medicinal Chemistry xx*, pages 1–11.

Mercer, J. (1909). *Functions of positive and negative type and their connection with the theory of integral equations*. Philos. Trans. Roy. Soc, London.

Michalewicz, S. D. (1992). *Genetic algorithms + data structures = evoluation programs*. Springer-Verlag, New York.

Mika, S., Rätsch, G., and Weston, J. (1999). Fisher discriminant analysis with kernels. In Hu, Y.-H., Larsen, J., Wilson, E., and Douglas, S., editors, *Neural Networks for Signal Processing IX*, pages 41–48.

Mika, S., Schölkopf, B., Smola, A., Müller, K., Scholz, M., and Rätsch, G. (1998). Kernel pca and denoising in feature spaces. In *Advances in Neural Information Processing Systems*, volume 11, San Mateo, CA. Morgan Kaufmann.

Montgomery, D. C. . E. A. P. . G. G. V. (2001). *Introduction to linear regression analysis*. Wiley-Interscience, 3rd edition.

Morozov, V. A. (1984). *Methods for solving incorrectly posed problems*. Springer-Verlag New York Inc.

Mose, L. E. (1986). *Think and explain with statistics*. Addison-Wesley, Reading, MA.

Nabney, I. T. (1999). Eifficient training of rbf networks for classification. In *Proceeding of Ninth International Conference on Artificial Neural Networks (ICANN99)*, pages 210–215. IEE.

Qin, S. J. and McAvoy, T. J. (1992). Nonlinear pls modeling using neural networks. *Computers and Chemical Engineering*, 7(4):379–391.

Rissanen, J. (1976). Minmax entropy estimation of models for vector processes. In Mehra, R. K. and Lainiotis, D. G., editors, *System identification*, pages 97–119. Academic Press, New York.

Rosipal, R. and Trejo, L. J. (2001). Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2(2):97–123.

Schmidt, G., Mattern, R., and Schüler, F. (1981). Biomechanical investigation to determine physical and traumatological differentiation criteria for the maximum load capacity of head and vertebral column with and without protective helmet under effects of impact. Eec research program on biomechanics of impacts, Institut für Rechtsmedizin, Universität Heidelberg, Germany.

Schölkopf, B., Smola, A., and Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464.

Sclove, S. L. (1973). Least squares problem with random regression coefficient. Technical Report 87, Institute for Mathematical Studies in the Social Sciences, Standford University, CA.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press, Cambridge.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):1–52.

Theil, H. and Goldberger, A. S. (1961). On pure and mixed statistical estimation in economics. *International Economic Review*, 7(1):65–71.

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244.

Urmanov, A. M. (2002). Information complexity-based regularization parameter selection for solution of ill conditioned inverse problems. *Inverse Problems*, 18:L1–L9.

van Emden, M. H. (1971). *An analysis of complexity*, volume 35. Mathematical Centre Tracts, Amsterdam.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer Verlag, New York.

Vogel, C. R. (1996). Non-convergence of the l-curve regularization parameter selection. *Inverse Problems*, 12:535–547.

Wahba, G., Golub, G. H., and Health, C. G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223.

Wahba, H. (1990). *Spline Model for Observational Data*, volume Society for Industrial and Applied Mathematics. Philadelphia and Pennsyvania.

Watanabe, S. (1985). *Pattern Recognition: Human and Mechanical*. John Wiley and Sons, New York.

Wilkinson, L. (1989). *SYSTAT: The system for statistics*. Evanston: SYSTAT.

Williams, C. (2002). On a connection betweeen kernel pca and metric multidimensional scaling. *Machine Learning*, 46:11–19.

Wold, H. (1966). *Nonlinear Estimation by Iterative Least Squares Procedures*. John Wiley, New York.

Wold, S. (1983). Pattern recognition: Finding and using regularities in multi-variate data. In Martens, H. and Russwurm, H., editors, *Food Research and Data Analysis*, pages 147–188. Applied Science Publishers, London.

Wold, S., Kettanech-Wold, N., and Skagerberg, B. (1989). Nonlinear pls modeling. *Chemometrics and Intelligent Laboratory Systems*, 7:53–65.

Wold, S., Ruhe, H., Wold, H., and III, W. J. D. (1984). The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverse. *SIAM Journal of Scientific and Statistical Computations*, 5:735–743.

Appendix

Appendix

A1 Description of Sinc Function

The sinc function has been widely used as the simulation data in the nonlinear machine learning area. The sinc function is defined as

$$y = \text{sinc}(x) = \frac{\sin(x)}{x} \tag{A1.1}$$

In each simulation, 100 observations were generated within the x range of [-10, 10] as the training data. The Gaussian noise $N(0, 0.1^2)$ has been added to the response. Additional 1000 noisy observations were generate as the testing data. This testing data is treated as the validation data when cross-validation is used for the model selection.

A2 Description of Friedman's Data

The true function of the Friedman's function Friedman (1991); Tipping (2001) is defined as

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \sum_{i=6}^{10} 0 \times x_i \tag{A2.2}$$

The 10 independent variables $x_1 - x_{10}$ are randomly generated from the unit hypercube. Variables x_6 through x_{10} have no contribution to the response y. The Gaussian noise $N(0, 1^2)$ is added to the response.

A3 Description of Boston Housing Dataset

Source: StatLib library, Carnegie Mellon University

Original Publication: Harrison, D. and Rubinfeld, D.L. "Hedonic prices and the demand for clean air", J. Environ. Economics & Management, vol.5, 81-102, 1978.

Data Description: Concerns housing values in suburbs of Boston.

Number of Observations: 506

Number of Independent Variables: 13 continuous variables (including "class" attribute "MEDV"), 1 binary variable (CHAS).

Response: MEDV: Median value of owner-occupied homes in 1000's

Independent Variable Description:

1. CRIM: per capita crime rate by town
2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS: proportion of non-retail business acres per town
4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX: nitric oxides concentration (parts per 10 million)
6. RM: average number of rooms per dwelling
7. AGE: proportion of owner-occupied units built prior to 1940
8. DIS: weighted distances to five Boston employment centers
9. RAD: index of accessibility to radial highways
10. TAX: full-value property-tax rate per $10,000$
11. PTRATIO: pupil-teacher ratio by town
12. B: $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
13. LSTAT: lower status of the population

Missing Values: None.

Vita

Rui Zhang was born and grew up in Beijing, China. After graduating in 1992 from No. 9 High School in Beijing, he attended Shanghai Jiaotong University in Shanghai where he received a Bachelor of Engineering degree in 1996, majored in Nuclear Engineering.

He joined China Zhongyuan Engineering and worked as a nuclear engineer for four years. He participated in biding and managing an international nuclear medicine project and served as a nuclear power plant startup engineer in an overseas site. He is a certified construction project manager since 1999. He also earned the certificate in Quality Assurance Systems and ISO 9000 Series Standards.

He returned to the academic world as a graduate student at the Department of Nuclear Engineering in the University of Tennessee, Knoxville in 2000 where he started learning statistical modeling techniques. He transferred to the Department of Statistics as a graduate student with the desire of being more professional in statistics.

In 2003 and 2005 he received his master of science degrees in Statistics and Nuclear Engineering respectively. He became a doctoral student in Business Administration with Statistics concentration since summer 2003.

In summer 2004, he worked as an intern in General Electric Global Research Center at Niskayuna, NY, helping engineering plastic formulation using experimental design techniques. He is the primary author of two conference publications. The author is the member of American Statistical Association. He enjoys teaching statistics.