1-1-2007

Prior-knowledge based Green's kernel for support vector regression

Tahir Farooq
Ryerson University

PRIOR-KNOWLEDGE BASED GREEN'S KERNEL FOR SUPPORT VECTOR REGRESSION

by

Tahir Farooq
B.Sc. Communication Engineering
University of Engineering and Technology (IIEC)
Taxila, Pakistan, 2004

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2007

Author's Declaration

I hereby declare that I am the sole author of this thesis.
I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.
Signature

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.
Signature

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Surely, your Lord is Allaah Who created the heavens and the earth in six Days and then Istawaa (rose over) the Throne (really in a manner that suits His Majesty), disposing the affair of all things. No intercessor (can plead with Him) except after His Leave. That is Allaah, your Lord; so worship Him (Alone). Then, will you not remember? (Quran 10:3)

Abstract

Prior-Knowledge Based Green's Kernel for Support Vector Regression

© Tahir Farooq, 2007

Master of Applied Science
Department of Electrical and Computer Engineering
Ryerson University

This thesis presents a novel prior knowledge based Green's kernel for support vector regression (SVR) and provides an empirical investigation of SVM's (support vector machines) ability to model complex real world problems using a real dataset. After reviewing the theoretical background such as theory of SVM, the correspondence between kernels functions used in SVM and regularization operators used in regularization networks as well as the use of Green's function of their corresponding regularization operators to construct kernel functions for SVM, a mathematical framework is presented to obtain the domain knowledge about the magnitude of the Fourier transform of the function to be predicted and design a prior knowledge based Green's kernel that exhibits optimal regularization properties by using the concept of matched filters. The matched filter behavior of the proposed kernel function provides the optimal regularization and also makes it suitable for signals corrupted with noise that includes many real world systems. Several experiments, mostly using benchmark datasets ranging from simple regression models to non-linear and high dimensional chaotic time series, have been conducted in order to compare the performance of the proposed technique with the results already published in the literature for other existing support vector kernels over a variety of settings including different noise levels, noise models, loss functions and SVM variations. The proposed kernel function improves the best known results by 18.6% and 24.4% on a benchmark dataset for two different experimental settings.

Acknowledgments

All praise is due to Allaah, the Lord of the Worlds. The Beneficent, the Merciful. Master of the Day of Judgment. Thee do we serve and Thee do we beseech for help.

I would like to sincerely thank my supervisor Dr. Aziz Guergachi and co-supervisor Dr. Sridhar Krishnan for introducing me to the fascinating worlds of statistical learning theory and statistical signal processing. I gratefully acknowledge their support, encouragement, suggestions and feedback during the development of this work.

I would like to thank my colleagues Dr. M. Shahbaz, Dr. K. Umapathy, and S. Zarei for providing a stimulating research environment.

I am grateful to A. Smola for a useful discussion on Green's kernel; M. Rupke, C. Monteith, C. Marshall and F. Basa at Ashbridges Bay Treatment Plant, Toronto for valuable information and the dataset; C. Wu-Tanenbaum for technical help and D. Wright for admin support. The financial support from NSERC, CFI, OIT, OCE Inc and Ryerson University is also highly acknowledged.

I am deeply indebted to my parents for their tremendous love, care, support and patience throughout the years of my life. I will never be able to repay them for their contribution in my life.

Finally, special thanks to my wife Marcelle for bringing all the light to my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Problem of Learning from Data

For a long time, development of machines capable of learning from the available knowledge (historical data) has been an object of both philosophical and technical discussion. One of the major contributions of the philosophical discussion on cognitive science and the artificial learning process is the information processing model of human thinking in which the metaphor of brain as computer is taken somewhat literally. On the other hand, the influx of electronic computing has provided a massive momentum to the technical facet of this discussion. Despite the fact that the boundaries of the artificial learning ability are faraway from being distinct, it has been demonstrated that the machines can show a prominent level of learning ability. Technically, the foundation of the learning approach is based on modeling the learning problem as problem of searching a suitable function in an appropriate hypothesis space. A major advantage of machine learning over classical modeling techniques is the evasion of laborious design and programming involved in classical modeling methods at the expense of collecting some labeled data. The availability of reliable learning machines can revolutionize many aspects of life where no mathematical model of the problem is available and the classical programming techniques can not be applied. The areas that have greatly benefitted from machine learning are medical diagnosis, natural language processing, syntactic pattern recognition, spam email filtering, search engines, bioinformatics and cheminformatics, detecting credit card fraud, stock market analysis, classifying DNA sequences,

speech and handwriting recognition, object recognition in computer vision, game playing and robot locomotion.

1.1.1 Supervised Learning

There are many real world situations where the classical modeling approach cannot be effective. For instance, the system designer cannot exactly specify the method by which the correct input/output relationship can be computed from the available data or the computation may be very expensive. When such situations arise, an alternative approach is to use machine learning methodology. Machine learning methodology is an artificial intelligence approach to construct a model to recognize the patterns or underlying mapping of a system based on a set of training examples consisting of input and output patterns. Within the machine learning methodology, the approach of using examples (input/output pairs) to synthesize models is known as supervised learning. The other types of learning that will not be discussed in this thesis include unsupervised learning, query learning and reinforcement learning. In unsupervised learning the output values are not available and the task of learner is to obtain some understanding of the process that generated the data. Query learning and reinforcement learning provide more complex interaction between the learner and its environment. In query learning the learner is allowed to query the environment about an output associated to a certain input whereas in reinforcement learning the learner can take one of several actions in order to move towards the state with the highest reward. However, there are significant complications since the quality of output can not be assessed until the consequence of an action becomes clear.

The idea of supervised learning is inspired by the fact that the kids are taught by the teacher to learn the real world problems such as classification by looking at some available examples and generalizing to unseen items. The examples of input/output are referred to as the training data. The input/output pairings typically reflect a functional relationship mapping inputs to outputs, though this is not always the case as for example when the outputs are corrupted by noise. When an underlying function from inputs to outputs exists,

it is referred to as the target function. The estimate of the target function which is learnt or output by the learning algorithm is known as the solution of the learning problem. The solution is chosen from a set of candidate functions which map from the input space to the output domain. Usually, a particular set or class of candidate functions known as hypotheses is chosen, before we begin trying to learn the correct function. Hence, the choice of the set of hypotheses (or hypothesis space) is one of the key ingredients of the learning strategy. The second important ingredient is the algorithm which takes the training data as input and selects a hypothesis from the hypothesis space. It is referred to as the learning algorithm. In the case of a learning problem where the output can take one of two possible values (for example, yes/no, true/false, +1/-1, etc.), the learning problem is referred to as binary classification problem. The learning problems with finite number of output categories is known as multi-class classification whereas a learning problem with real valued output is called regression.

1.1.2 Learning and Generalization

An important aspect of learning methodology is based on how the data is generated and presented to the learner. One of the most fundamental types of learning used in the modern machine learning world is probably approximately correct (PAC) learning model. In a PAC learning framework, the learner receives training examples as input. The training data is drawn at random according to an unknown but fixed probability distribution and labeled according to an unknown target function (the function to be learnt). The goal of learning is to return a hypothesis that (with high probability of success) is a close approximation of the target function. In this thesis, the PAC learning model is assumed.

Within the PAC framework, many setting for learning can be realized. One of these settings called batch learning is to provide the learner with all the data at the start of learning process. Another setting called online learning is to provide the learner with one example at a time and asses the quality of hypothesis. The current hypothesis is updated after every example and the quality of hypothesis is assessed by the total number of mistakes

made during the process of learning. In this thesis, only batch learning is considered since support vector machines (SVM) are inherently batch learning machines. The question that needs to be addressed now is, how the quality of the hypothesis generated during batch learning can be assessed? The goal of early learning algorithms was to provide an accurate fit to the data and generate a hypothesis that performs best on training data. This type of hypothesis is known as consistent hypothesis. A consistent hypothesis might work well with simple noise free data. However, this setting gives poor performance outside training data (for unseen data) and is often rendered failed for complex and noisy real world problems. The capability of a hypothesis to perform correct classification of unseen data is known as generalization and this is the criterion that modern machine learning methodology has aimed to optimize. This leads to an important concept of capacity and overfit. The capacity is a measure of complexity and richness of the hypothesis and is related to the ability of a machine to learn a given finite training data without error. Many traditional learning algorithms such as neural networks are capable of learning any given training set, resulting in high capacity hypothesis for difficult training sets. The hypothesis that become too complex during the learning process in order to be consistent are known as overfit. In order to achieve good generalization performance, there has to be a right balance between the accuracy obtained on a given finite training set and the capacity of the hypothesis.

1.2 Support Vector Machines for Learning

Support Vector Machines (SVM) are linear learning machines based on the statistical learning theory given by V. N. Vapnik. In support vector machines, the data in original input space is mapped into a high dimensional space called feature space. The feature space in a non-linear dot product space induced by specialized mapping functions called kernel functions. The goal of basic SVM, designed for the binary classification problem, is to construct the optimal hyperplane (linear solution) in the feature space. The optimal hyperplane is constructed by maximizing the generalization criterion with the help of so-called Vapnik's statistical learning theory, i.e. a balance is achieved between the number of mistakes made during the training

and the capacity of the generated hypothesis. SVM have several benefits compared to the traditional learning machines such as neural networks, decision trees, linear discriminant analysis (LDA), Bayesian analysis, etc. One of the most important characteristics of SVM is that they can handle very high dimensional feature spaces and their generalization ability and computational efficiency are both independent of the dimensionality of the input space. This is a very desirable property of SVM, especially for the problems where the input space has high dimensionality. One of the examples of such problems is the image recognition where the data in input space is high dimensional. Another advantage of SVM compared to the conventional algorithms is that the SVM always result in a globally optimal solution and the convergence is guaranteed for the problem under observation. For the comparison, one could consider the neural networks (NN) which may be trapped in a local minima, leading to a sub-optimal solution or might not converge for a given parameter settings. Further advantage of using SVM is the sparseness of the solution that results in efficient computation. Additionally, SVM have only two adjustable parameters to obtain optimal solution, i.e. regularization parameter of SVM and the kernel parameter as compared to for instance, neural networks that have large number of adjustable parameters such as number of nodes, number of hidden layers, momentum, threshold, number of iterations and learning rate.

1.3 Learning with Kernels

Due to the exposition of Vapnik's SVM during the last decade and the reported success SVM have witnessed, there has been an increased interest in learning systems that express the similarity between two instances (examples) as the dot product of vectors in an appropriate high-dimensional feature space using a non-linear mapping function. Since linear SVM carry limited computational power and complex real world application require more expressive hypothesis than linear functions, SVM are equipped with kernel functions to obtain a linear solution in the feature space to the originally non-linear problem. The dot product operation is often not computed explicitly, but reduced to the computation of the kernel function which

operates on instances of the input space. This provides a way to overcome the so-called curse of dimensionality and handle high-dimensional feature spaces in an efficient manner. The kernel function offers the description language used by the machine for viewing data. Hence, the choice of an appropriate kernel function is an important issue in SVM system design. The standard choices are Gaussian RBF kernel and polynomial kernel. However, situations may arise where more elaborate kernels are needed. The simple kernel functions can be used as building blocks to construct more complicated kernel functions under the set of Mercer conditions. A striking strategy could be to let the data speak of itself and design the appropriate kernel function in response. This leads to the idea of prior knowledge based kernel design.

1.4 Prior Knowledge based Learning

Prior knowledge in machine learning refers to any auxiliary information available about the learning problem in addition to the training data. Although PAC learning framework assumes the existence of an unknown but fixed underlying probability distribution, a certain amount of additional knowledge on the problem is usually available beforehand. For example, in image recognition tasks, if an image is slightly translated or rotated it still represents the same information. The availability of this type of prior knowledge indicates that invariance to translations and rotations should be incorporated into the classifier for image recognition tasks. Other types of knowledge that occur most often in real world data are the knowledge of the degree of smoothness of a function, imbalance of the training set (high proportion of samples of the same class) and variation in the quality of data from one sample to another. If properly incorporated in the learning machine, prior knowledge associated with these cases can improve the quality of the learning. Furthermore, not taking into consideration large imbalance between classes and poor quality of data can mislead the learning machine.

Prior knowledge can be incorporated into learning machines in several ways. Nevertheless, only the methods of prior knowledge incorporation into SVM are considered.

1.4.1 Virtual Examples

The virtual examples approach is usually used to incorporate transformation invariances into SVM for image recognition task. The generalization ability of SVM depends on the number of data examples available for training. The more representative example we have, the better we learn. The basic idea of the virtual examples approach is to incorporate a known transformation-invariance into SVM by applying the transformation to the training examples and generate new virtual examples to enlarge the training set.

1.4.2 Weighting of Examples

This approach is typically used to incorporate knowledge on the data such as imbalance between classes and the relative quality of the examples. This is done by choosing a different SVM regularization parameter C for different classes based on the imbalance between the respective classes. A larger value of C is assigned for the less represented class, thereby penalizing more the errors on this class.

1.4.3 SVM Optimization

The following method achieves the incorporation of prior knowledge into SVM by the addition of new constraints to the original SVM optimization problem. However, the resulting optimization schemes are complex and difficult to handle.

1.4.4 Knowledge based Kernels

This approach corresponds to designing knowledge driven support vector kernel functions. One of the advantages of this approach is that the new kernel function can be used with off the shelf SVM algorithms. In this thesis, the problem of designing knowledge driven kernel functions based on prior knowledge about the smoothness properties of the function to be learnt is considered. Also, it has been showed that the proposed technique is noise tolerant.

1.5 Thesis Organization

This thesis is divided into five chapters and two appendices. Besides the introduction, Chapter 1; Chapter 2 provides the theoretical background about: the learning paradigm of statistical learning theory, support vector method for binary classification and regression, least square support vector machines, the role of kernel functions in SVM, the characteristics of existing kernel functions, the theory of regularization networks for stochastic ill-posed problems, correspondence between regularization networks and SVM as well as the connection between regularization operators and support vector kernels.

Chapter 3 presents the literature review and defines the problem statement for the following subject matter:

- Evaluation of SVM's capability to model complex real world problems.
- Construction of a novel prior knowledge based kernel function for support vector regression.
- Generalization to time series prediction using NARX and NOE models.

Chapter 4 discussed the proposed solution and provides the simulation results. A real dataset from Ashbridges Bay Wastewater Treatment Plant, Toronto has been used to evaluate SVM's performance for modeling complex real world problems. To evaluate the performance of the proposed kernel function, benchmark datasets are used and the results obtained by the proposed kernel function are compared against the results already published in the literature for conventional support vector kernel functions.

The conclusions, recommendations and future work are summarized in Chapter 5.

Appendix A gives a detailed mathematical proof for the dual formulation of SVM quadratic optimization problem. Although very brief versions of this proof are already available in the literature, a detailed step by step illustrative proof is presented.

Appendix B provides the list of publications that have been generated by this research work.

Chapter 2

Theoretical Background: A Learning Paradigm

2.1 Statistical Learning Theory

In contrast to the classical statistics based learning which relies on large datasets and strong priori information about the problem at hand such as underlying probability density function, a new learning paradigm, usually referred to as statistical learning theory (SLT) introduced by Vapnik [4, 9] is gaining more presence. SLT explores the ways of estimating functional dependencies from a finite collection of data mainly relatively small datasets. However, the problems it addresses are general and also covered by classical statistics based learning, i.e. discriminant analysis, regression analysis and density estimation problems.

2.1.1 The Structural Risk Minimization Principle

The principle of structural risk minimization (SRM) is an important derivative of SLT. In contrast to empirical risk minimization principle which seeks to minimize the empirically measured value of a certain risk functional, this principle looks for the optimal relationship between the amount of empirical data available, the quality of approximation of the data by the function chosen from a given set of functions and the value called VC (Vapnik Chervonenkis) dimensions that characterizes the capacity of a set of functions [9]. According to the structural risk minimization principle [4, 9], the generalization error can be upper bounded

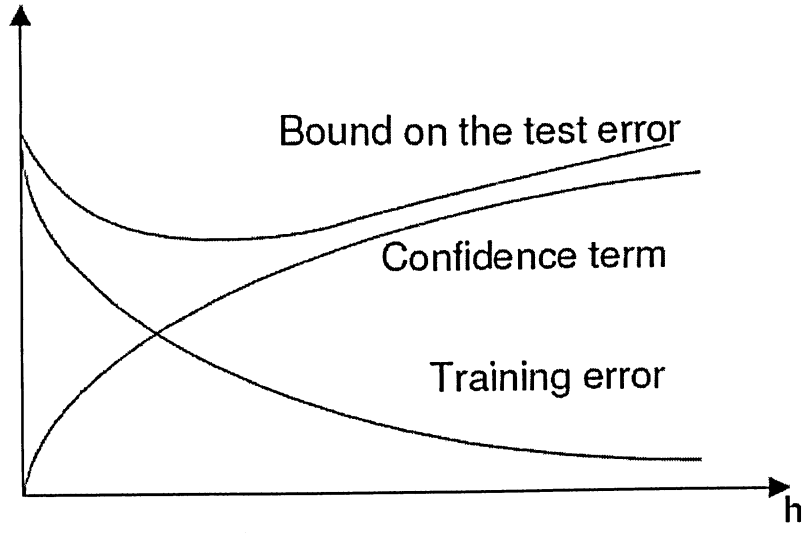


Figure 2.1: The structural risk minimization

in terms of empirical error (the training error) and a confidence term; i.e.

$$R(\alpha) = R_{emp}(\alpha) + \sqrt{\frac{h(\ln(2N/h+1)) - \ln(\eta/4)}{N}} \tag{2.1}$$

where N is the number of training examples and h is VC dimensions. VC dimensions is used to describe the capacity of a given set of functions or in other words the complexity of the learning system [4, 9]. The term on the left side represents the generalization error, whereas the first term on the right side is empirical error calculated from the training data and the second term is called confidence term which is related to the VC dimensions h of the learning machine and size $\grave{N}$ of the training set. The upper bound on generalization error given by (2.1) holds with probability $1 - \eta$. An important result is that the upper bound on the generalization error given by (2.1) is independent of underlying generator of the data characterized by the probability distribution $p(x, y)$, which is usually unknown in practice. Figure 2.1 shows the relationship between the generalization error, training error, confidence term and the VC dimension. Unlike the principle of empirical risk minimization (ERM) which aims to minimize the training error, the structural risk minimization (SRM)

takes both the training error and the complexity of the model (VC dimension) into account and tends to find the minimum of the sum of two terms as a trade-off solution by searching from a given set of functions. One of the most prominent outcomes of the SRM principle is support vector machines.

2.2 The Support Vector Machines (SVM)

Over the last decade, support vector machines (SVM) introduced by Vapnik and co-workers [1, 2, 4] have been reported [6, 9] to be an effective method for the problem of pattern recognition and regression estimation. SVM implement the learning bias derived from statistical learning theory and use a hypothesis space of linear functions in a high dimensional feature space. The simplest SVM were developed for binary classification. However, with continuous extension and advancement, generalizations to functional approximation and time series prediction were developed shortly after.

2.2.1 The Optimal Separating Hyperplane for Binary Classification

Consider a binary classification problem with $\boldsymbol{x}_i \in \mathbb{R}^d$ as the input feature vector and $y_i \in \{-1, +1\}$ the class labels (i.e. $\{(\boldsymbol{x}_i, y_i), \ldots, (\boldsymbol{x}_N, y_N)\}$ are the training set). The hyperplane which separates the two classes is described by

$$f(\boldsymbol{x}) = \text{sgn}(\langle \boldsymbol{w}.\boldsymbol{x} \rangle + b) \tag{2.2}$$

where $\boldsymbol{w}$ is the coefficient vector, b is the bias of the hyperplane and sgn[.] stands for the bipolar sign function. The objective of SVM algorithm is to choose the optimal separating hyperplane that maximizes the margin between the two classes (Figure 2.2) [4, 9]. The hyperplane that has the maximum distance to the closest point is called the optimal separating hyperplane. In view of the fact that the distance from the hyperplane to the closest point is $\frac{1}{\|\boldsymbol{w}\|}$, twofold of this distance is called the margin. The margin provides a measure of the generalization ability of the hyperplane to separate the data into corresponding classes. The

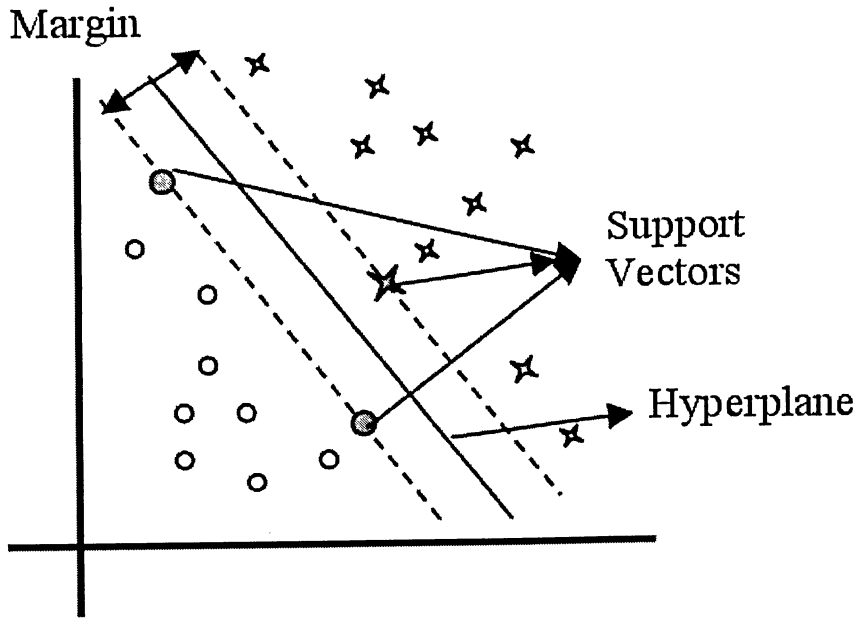


Figure 2.2: A linear SVM classifier. Support vectors are those elements of the training set which are on the boundary hyperplanes of two classes.

larger the margin, the better the generalization ability is expected to be [4, 9, 14]. The optimization problem that yields the hyperplane can be written as

$$minimize_{w,b} \quad \frac{1}{2}\|w\|^2 \tag{2.3}$$

subject to

$$y_i(\langle w.x_i\rangle + b) \geq 1 \qquad i = 1, \ldots, N \tag{2.4}$$

2.2.2 Soft Margin Generalization

In many real-world problems the data is noisy; therefore in general there will be no linear separation. In this case, instead of hard margin, soft margin (the noise tolerant version) is used and slack variables are introduced to relax the constraints as described by [3]. So the

optimization problem would be

$$minimize_{w,b} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\zeta_i^2 \tag{2.5}$$

subject to

$$y_i(\langle w.x\rangle + b) \geq 1 - \zeta_i \qquad \zeta_i \geq 0, \quad i = 1, \ldots, N \tag{2.6}$$

where ζ_i are the slack variables and C is a regularization parameter which helps manage the trade off between the empirical risk (reflected by the second term in (2.5)) and the model complexity (reflected by the first term in (2.5). The positive constant C is required to be set up before solving (2.5). A larger value of C means a higher penalty is assigned to the training errors [4]. This minimization problem is a practical realization of the structural risk minimization principle on a given set of functions.

To solve the convex quadratic optimization problem given by (2.5), the standard lagrange multiplier technique is used [3, 4, 9] and the primal Lagrangian function is constructed as follows:

$$L(w, b, \alpha) = \frac{1}{2}\langle w.w\rangle - \sum_{i=1}^{N}\alpha_i[y_i(\langle w.x_i\rangle + b) - 1] \tag{2.7}$$

where α_i are the Lagrange multipliers. The corresponding dual is found by differentiating with respect to w and b:

$$W(\alpha) = \sum_{i=1}^{N}(\alpha_i) - \frac{1}{2}\sum_{i,j=1}^{n} y_i\, y_j\, \alpha_i\, \alpha_j \langle x_i.x_j\rangle \tag{2.8}$$

subject to

$$\sum \alpha_i\, y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N \tag{2.9}$$

By making use of Karush-Kuhn-Tucker (KKT) conditions as described by [9, 14], the optimal hyperplane takes the following expression known as support vector formulation, i.e.

$$f(x, \alpha^*, b) = \sum_{i=1}^{N_{SV}} y_i\, \alpha_i^* \langle x_i.x\rangle + b \tag{2.10}$$

where N_{SV} is the number of support vectors and α_i^* are non-zero Lagrange multipliers corresponding to the support vectors. This result shows that the data points that are not support vectors have no influence on the solution.

2.2.3 ϵ-Insensitive Support Vector Regression

Initially developed for classification problems, a generalization of SV algorithm known as ϵ-insensitive support vector regression as described in [3, 4] was derived to solve the problems where the function to be estimated belongs to the set of real numbers. Suppose we have $\{(\boldsymbol{x}_i, y_i), \ldots, (\boldsymbol{x}_N, y_N)\}$, as the training set with $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, where y_i are the training targets. The problem of calculating an estimate $f(\boldsymbol{x}_i)$ of y_i for training data $\{|\boldsymbol{x}_i, y_i|_{i=1,\ldots,N}\}$ can be formulated as

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}.\boldsymbol{x} \rangle + b \tag{2.11}$$

The goal of ϵ-insensitive SV algorithm is to calculate an estimate $f(\boldsymbol{x}_i)$ of y_i by selecting the optimal hyperplane $\boldsymbol{w}$ and bias b such that $f(\boldsymbol{x}_i)$ is at the most ϵ distance from y_i while keeping the norm $\|\boldsymbol{w}\|^2$ of the hyperplane minimum. From [4, 5, 9], Vapnik's ϵ-insensitive loss function can be written as

$$|y_i - f(\boldsymbol{x}_i)|_\epsilon = \begin{cases} |y_i - f(\boldsymbol{x}_i)| - \epsilon & for \quad |y_i - f(\boldsymbol{x}_i)| \geq \epsilon \\ 0 & otherwise \end{cases} \tag{2.12}$$

The corresponding quadratic optimization problem can be written in terms of regularized risk functional as described by [6, 7], i.e. to minimize

$$\Re[f] = \frac{\gamma}{2}\|\boldsymbol{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} |y_i - f(\boldsymbol{x}_i)|_\epsilon \tag{2.13}$$

where $\Re$ is the regularized risk functional, γ is the regularization constant such that $\gamma \geq 0$ and the second term on the right hand side of (2.13) is the empirical risk functional. In order to minimize $\Re$, one needs to find the optimal hyperplane $\boldsymbol{w}$ such that the sum of distances, in the sense of Vapnik's ϵ-insensitive loss, from the hyperplane to the data points is minimized while keeping the norm of $\boldsymbol{w}$ minimum. Minimizing $\|\boldsymbol{w}\|^2$ in regression estimation corresponds to imposing flatness in the feature space which relates to maximizing the margin between the data classes in pattern classification [1, 3]. By introducing the slack variables, in the sense of [4, 5, 9] and rewriting the problem in (2.13), i.e. to minimize

$$\Re[f] = \frac{\gamma}{2}\|\boldsymbol{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} (\zeta_i + \zeta_i^*) \tag{2.14}$$

subject to

$$\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + b - y_i \leq \epsilon + \zeta_i \tag{2.15}$$

$$y_i - \langle \boldsymbol{w}.\boldsymbol{x}_i \rangle - b \leq \epsilon + \zeta_i^* \tag{2.16}$$

$$\zeta_i,\ \zeta_i^* \geq 0 \tag{2.17}$$

However, for the sake of simplicity and to be consistent with literature [4, 5, 8], the transformation given by [9]; between γ, N and C; is used and the expression in (2.14) is written as

$$\textit{minimize} \qquad \Re[f] = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\zeta_i + \zeta_i^*) \tag{2.18}$$

with the constraints given by (2.15), (2.16) and (2.17).

In order to obtain the SV expansion of the function $f(\boldsymbol{x})$, the Lagrangian and the corresponding constraints are constructed from the objective function, i.e.

$$\begin{aligned} L = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\zeta_i + \zeta_i^*) - \sum_{i=1}^{N}(\eta_i\zeta_i - \eta_i^*\zeta_i^*) \\ - \sum_{i=1}^{N}\alpha_i(\epsilon + \zeta_i + y_i - \langle \boldsymbol{w}.\boldsymbol{x}_i \rangle - b) \\ - \sum_{i=1}^{N}\alpha_i^*(\epsilon + \zeta_i^* - y_i + \langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + b) \end{aligned} \tag{2.19}$$

where α_i, α_i^*, ζ_i, ζ_i^* are non-negative. Taking partial derivatives of the function in (2.19) with respect to $\boldsymbol{w}$, b, ζ_i, ζ_i^*. The partial derivatives must be zero for the optimal solution.

$$\partial_b L = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \tag{2.20}$$

$$\partial_{\boldsymbol{w}} L = \boldsymbol{w} - \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\boldsymbol{x}_i = 0 \tag{2.21}$$

$$\partial_{\zeta_i} L = C - \eta_i - \alpha_i = 0 \tag{2.22}$$

$$\partial_{\zeta_i^*} L = C - \eta_i^* - \alpha_i^* = 0 \tag{2.23}$$

Substituting the values from (2.20), (2.21), (2.22) and (2.23) into (2.19), yields the following optimization problem (see Appendix A for the detailed derivation)

$$maximize \quad \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\langle \boldsymbol{x}_i.\boldsymbol{x}_j\rangle \\ -\epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)y_i \end{cases} \tag{2.24}$$

subject to

$$\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \tag{2.25}$$

$$0 \leq \alpha_i, \alpha_i^* \leq \frac{1}{\gamma N} \tag{2.26}$$

following the lines of [7] and re-writing the quadratic optimization problem yields

$$minimize \quad \begin{cases} \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\langle \boldsymbol{x}_i.\boldsymbol{x}_j\rangle \\ +\epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)y_i \end{cases} \tag{2.27}$$

subject to the constraints given by (2.25) and (2.26). Using (2.21) gives

$$\boldsymbol{w} = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\boldsymbol{x}_i \tag{2.28}$$

and using (2.11) yields

$$f(\boldsymbol{x}) = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\langle \boldsymbol{x}_i.\boldsymbol{x}\rangle + b \tag{2.29}$$

Equation (2.29) provides the well-known formulation of SV regression. Comparing (2.29) and (2.11), the training examples that lie inside the ϵ-tube contribute to a sparse expansion of $\boldsymbol{w}$ because the corresponding Lagrange multipliers α_i, α_i^* are zero [4, 5, 6, 8, 9]. This is a desirable property of SVR (support vector regression) since only the training examples that turn out to be support vector are required to produce the approximation of the given function with at most ϵ-error chosen a priori.

2.2.4 Least Squares Support Vector Machines for Regression

Since the SVM were introduced by Vapnik, many variations of SVM have been developed by other researchers to leverage the strengths while overcoming the difficulties in algorithmic implementation. One of these variations known as the least squares SVM (LS-SVM)

was introduced by Suykens and co-workers [38]. LS-SVM implements the ridge regression cost function. The inequality constraints in SVM are replaced with equality constraints. As a consequence, the solution follows from solving a set of linear equations instead of a quadratic programming problem which is used in the original SVM formulation of Vapnik [4, 9], obviously resulting in a faster algorithm.

The primal problem of the LS-SVM is defined as

$$minimize_{w,b,e} \quad J_P[\boldsymbol{w}, \boldsymbol{e}] = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\rho}{2}\sum_{i=1}^{N}(e_i^2) \tag{2.30}$$

subject to

$$y_i(\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + b) = 1 \quad i = 1, \ldots, N \tag{2.31}$$

where $\boldsymbol{e}$ is the error variable in the sense of least square minimization and ρ is a parameter analogous to SVM's regularization parameter C. These error variables play a similar role as the slack variables in SVM formulation such that relatively small errors can be tolerated [38]. In the case of linear function approximation, one could easily solve this primal problem as it involves linear equality constraints. But, when the dimension of $\boldsymbol{w}$ becomes very large, the primal problem is difficult to solve. The solution is to derive the dual problem by constructing the Lagrangian for this primal problem:

$$L(\boldsymbol{w}, b, e, \alpha) = J_P(\boldsymbol{w}, e) - \sum_{i=1}^{N} \alpha_i(\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + b + e_i - y_i) \tag{2.32}$$

Taking the condition for optimality of the Lagrangian, i.e., taking partial derivatives of the function in (2.32) with respect to $\boldsymbol{w}$, b, e_i, α_i and setting them equal to zero yields a set of linear equations.

$$\begin{cases} \frac{\partial L}{\partial \boldsymbol{w}} = 0 \quad \rightarrow \quad \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i \\ \frac{\partial L}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^{N} \alpha_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 \quad \rightarrow \quad \alpha_i = \gamma e_i, \quad i = 1, \ldots, N \\ \frac{\partial L}{\partial \alpha_i} = 0 \quad \rightarrow \quad \langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + b + e_i - y_i = 0, \quad i = 1, \ldots, N \end{cases} \tag{2.33}$$

Solving this set of linear equations in α and b, yields the resulting LS-SVM formulation for function approximation

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \langle \boldsymbol{x}.\boldsymbol{x}_i \rangle + b \tag{2.34}$$

The main characteristic of LS-SVM is the low computational complexity comparing to SVM without quality loss in the solution. On the other hand, one of the drawback of LS-SVM is the loss of sparsity in the solution, since the error would not be zero for most of data points. One can overcome the loss of sparsity using special pruning techniques and obtain a sparse approximation [38]. Pruning is a technique that can be used to obtain the sparse approximation by iteratively eliminating a small set (e.g. 5%) of the less relevant support vectors until the user-defined performance index degrades.

2.3 Learning in Feature Space

In complex real world problems, situations often arise where more expressive hypothesis spaces than linear functions are required to learn the target function from data. For example, in the case of classification, the data might not be linearly separable in the input space. The complexity of the target function (i.e. the function to be learnt) depends on the way it is represented, and the difficulty of the learning task can vary accordingly [14]. One way of dealing with non-linearly separable functions is to use a non-linear mapping to project the data into a high dimensional feature space, thereby changing the data representation and extending the computational complexity of the linear machines. A linear machine such as SVM can then be employed in the feature space to solve the original non-linear problem. In SVM, the kernel functions are used to perform the non-linear mapping of data in high dimensional feature space.

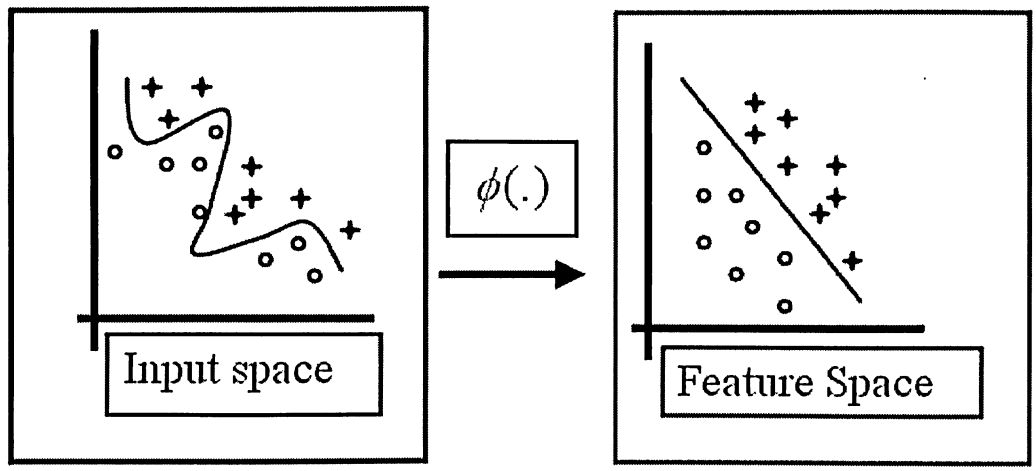


Figure 2.3: Mapping data from input space to a higher dimensional feature space in order to classify them by a linear function

2.3.1 Kernel-Induced Feature Space

In the context of machine learning, the kernel trick was first introduced by Aizerman [10]. Kernel trick involves changing the representation of the data, i.e.

$$\boldsymbol{x} = (x_1, \ldots, x_n) \quad \rightarrow \quad \boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_N(\boldsymbol{x}))$$

where $\boldsymbol{\phi}(.)$ is a non-linear operators mapping from input space $\mathfrak{X}$ into a high dimensional feature space $\mathfrak{F}$, $\boldsymbol{\phi} : \mathfrak{X} \rightarrow \mathfrak{F}$ [10, 11]. This mapping can greatly simplify the learning task [14, 6]. Figure 2.3 shows the mapping of data from input space to a higher dimensional feature space by a non-linear operator $\boldsymbol{\phi}(.)$, in order to classify the data by a linear boundary. However, a problem associated with high dimensional feature spaces is that as the number of features grow, the generalization performance can degrade and the solution can become computationally expensive. This phenomenon is known as curse of dimensionality [14]. Although, dimensionality reduction can be performed by removing the features corresponding to low variance in data, there is no guarantee that these features are not essential for learning. Support vector machines are inherently equipped with a linear combination of the dot

product between the data points that turn out to be support vectors and hence can defy the curse of dimensionality by using the dot product kernel functions.

The dot product kernel function is defined as

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}).\phi(\boldsymbol{x}') \rangle \tag{2.35}$$

$$\int_{\mathcal{X}^2} k(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}) f(\boldsymbol{x}')\, d\boldsymbol{x}\, d\boldsymbol{x}' \geq 0 \quad \forall f \in L_2(x) \tag{2.36}$$

According to Mercer theorem [12], the kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ is any continuous and symmetric function that satisfies the condition of positive semi-definiteness given by (2.36). Such a function defines a dot product in the feature space given by (2.35).

Linear SVM can be readily extended to non-linear SVM by using (2.35) and (2.29) can be written as

$$\begin{aligned} f(\boldsymbol{x}) &= \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) \langle \phi(\boldsymbol{x}_i).\phi(\boldsymbol{x}) \rangle + b \\ &= \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) k(\boldsymbol{x}_i, \boldsymbol{x}) + b \end{aligned} \tag{2.37}$$

It can also be observed from (2.37) that this is a formulation, where the problem of estimating from a subset of the training examples namely the support vectors can be written as the linear combination of the dot products in feature space between the training examples that turn out to be support vectors and there is no need to know the feature map and its properties explicitly. However, the knowledge about this feature map and its properties can provide us with some additional insight about the support vector kernels and might be helpful in answering the question why this mapping usually provides good results [7, 9].

The kernel functions satisfying Mercer condition [34] not only enable implicit mapping of data from input space to feature space but also ensure the convexity of the cost function which leads to the unique and optimum solution.

An important consequence of (2.35) is that the dimensionality of the feature space does not affect the computational cost since the feature vector does not need to be computed explicitly. Another important note at this point is that since the kernel function defines the representation of data in the feature space, ideally a representation that matches the specific

learning problem should be chosen. This leads to the problem of choosing the suitable kernel function for SVM. This problem is addressed in Chapter 4 and it has been shown, how the prior knowledge about the function to be estimated can be used to build admissible support vector kernels.

2.3.2 Examples and Characteristics of Admissible Support Vector Kernels

This section discusses some of the commonly used support vector kernel functions.

Gaussian radial basis function (RBF) is perhaps the most commonly used support vector kernel. The use of Gaussian RBF kernel for SVM was suggested by [1, 2, 4] Gaussian RBF kernel is given by

$$k(x, x') = \exp\left(\frac{\|x - x'\|^2}{2\sigma^2}\right) \tag{2.38}$$

where σ is the kernel width, which is a hyper-parameter and its value is chosen beforehand. Gaussian RBF kernel is a translational invariant kernel, i.e.

$$k(x, x') = k(x + x_0,\, x' + x_0) \quad \forall \;\; x_0 \in \mathfrak{X} \tag{2.39}$$

Inhomogeneous polynomial kernel is also one of the commonly used admissible (positive semi-definite kernel functions that satisfy Mercer conditions) support vector (SV) kernel. Polynomial kernel is given by

$$k(x, x') = (\langle x, x' \rangle + c)^d \tag{2.40}$$

where $d \in \mathbb{N}$ is the degree of polynomial kernel and c is a positive constant. Degree of the polynomial kernel is a hyper-parameter chosen beforehand. Polynomial kernels are rotational invariant, i.e.

$$\langle x, x' \rangle = \langle Rx, Rx' \rangle \tag{2.41}$$

where R is an orthogonal transformation.

Another example of the common SV kernels is Multi-layer Perceptron (MLP) described by

$$k(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2) \tag{2.42}$$

where $\kappa_1, \kappa_2 \in R$ are the kernel hyper-parameters. However, MLP kernel satisfies the condition of positive semi-definiteness only for certain values of κ_1, κ_2.

For a useful discussion on other commonly used admissible support vector kernels and how to make new kernel functions from existing kernels, the reader is referred to [14, 6].

2.4 Theory of Regularization for Stochastic Ill-Posed Problems

The idea of regularization method was first given by Tikhonov [15, 17] for the solution of ill-posed problems. The term "ill-posed" refers to the type of problems where the solution of an operator equation does not exist, is not unique or is unstable.

2.4.1 Regularization Networks and Support Vector Regression

Assume, a given finite data set $\{(\boldsymbol{x}_i, y_i), \ldots, (\boldsymbol{x}_N, y_N)\}$, independently and identically drawn from a probability distribution $p(\boldsymbol{x}, y)$ in the presence of noise. Assume that the probability distribution $p(\boldsymbol{x}, y)$ is unknown. One way of approaching the problem is to estimate a function by minimizing a certain empirical risk functional:

$$\Re_{emp}[f] = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(\boldsymbol{x}_i)|_\epsilon \tag{2.43}$$

In this thesis, the focus is on Vapnik's ϵ-insensitive loss function given by (2.12). For useful discussion on other types of loss functions in the context of regularization theory, the reader is referred to [6, 8, 20, 22]. The problem of approaching the solution through minimizing (2.43) is ill-posed because the solution is unstable [9]. Another problem associated with this technique is that if there is no prior information available about the problem, then there might be infinite possible solutions [18]. As a consequence, the hypothesis space of functions that contains the possible solutions is very rich and has a high capacity. This setting would lead to overfitting and hence poor generalization behavior. Thus, the solution is to utilize the idea proposed by [15, 17] and add a capacity control or stabilizer [19] term to (2.43) and

minimize a regularized risk functional.

$$\Re[f] = \frac{1}{N}\sum_{i=1}^{N}|y_i - f(x_i)|_\epsilon + \frac{\gamma}{2}\|\lambda f\|^2 \tag{2.44}$$

where λ is a linear, positive semi definite regularization operator. The first term in (2.44) corresponds to finding a function that is as close to the data examples as possible in terms of Vapnik's ϵ-insensitive loss function whereas the second term is the smoothness functional and its purpose is to restrict the size of the functional space and to reduce the complexity of the solution [20]. A function is said to be smoother than the other one if the former one has less energy at high frequency in the Fourier sense. The filter properties of λ are given by $\lambda^*\lambda$, where λ^* represents the complex conjugate. The optimal filter can be chosen by utilizing some of the classical model selection techniques such as cross validation, VC theory [16] or AIC [47]. In the following section, the filter properties of some of the commonly used support vector kernels are discussed. However, the purpose here is to illustrate the reasoning that the optimal filter for approximating the function can be obtained from the prior knowledge about data by utilizing the concept of matched filters based on correlation.

The next step is to formulate the problem of minimizing the regularized risk functional given by (2.44) into a quadratic optimization problem and highlight the relationship between support vector (SV) technique and regularization networks. Again, the loss function is chosen to be ϵ-insensitive and following the lines of [20], the problem of minimizing (2.44) is transformed into constrained optimization problem by utilizing the standard Lagrange multipliers technique and a formulation similar to (2.24) is obtained, i.e. to

$$minimize \quad \begin{cases} \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(KD^{-1}K) \\ +\epsilon\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) - \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)y_i \end{cases} \tag{2.45}$$

subject to

$$\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \tag{2.46}$$

$$0 \leq \alpha_i, \alpha_i^* \leq \frac{1}{\gamma N} \tag{2.47}$$

where

$$D_{ij} = \langle(\lambda k)(\boldsymbol{x}_i, .).(\lambda k)(\boldsymbol{x}_j, .)\rangle \tag{2.48}$$

However, this formulation does not lead to a sparse decomposition because of $D^{-1}K$ and cannot be related to support vector method given by (2.37) which offers a sparse solution since many of the coefficients α_i, α_i^* vanish. A necessary and sufficient condition to obtain a relationship between the two methods is to set $D = K$ such that $KD^{-1}K = K$ [7, 8]. Hence, K and $f(\boldsymbol{x})$ can be written as

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle(\lambda k)(\boldsymbol{x}_i, .).(\lambda k)(\boldsymbol{x}_j, .)\rangle \tag{2.49}$$

$$f(\boldsymbol{x}) = \sum_{i=1}^{M}(\alpha_i^* - \alpha_i)k(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{2.50}$$

Comparing (2.37) and (2.50), a direct relationship between the support vector method and the regularization networks can be observed. In other words, training an SVM with a kernel function obtained from the regularization operator λ is equivalent to implementing a regularization network to minimize the regularized risk functional given by (2.44) with λ as the regularization operator.

2.4.2 Green's Functions of Their Corresponding Regularization Operators and Support Vector Kernels

The idea of Green's functions was introduced in the context of solving inhomogeneous differential equations with boundary conditions. However, in this section, the idea of [7, 8, 20] is borrowed and it is demonstrated how, given a regularization operator λ, the concept of Green's functions is utilized to design kernel functions that exhibit the regularization properties given by their corresponding regularization operators, satisfy the Mercer condition and qualify to be support vector (SV) kernels.

The Green's function of a given regularization operator λ can be written as [6]

$$G(x, x') = (2\pi)^{-\frac{1}{2}} \int_{\Omega} e^{\iota\omega(x-x')}|\lambda(\omega)|\, d\omega \tag{2.51}$$

For the sake of simplicity, one-dimensional case is assumed. The generalization to multi-dimensions is straight forward and will be discussed later. The integration is carried over the domain $\Omega = Sup[|\lambda(\omega)|]$, where $|\lambda(\omega)|$ is the magnitude of the Fourier transform of λ, describing its filter properties, satisfying [6, 7]

$$\langle \lambda U.\lambda V \rangle = (2\pi)^{-\frac{1}{2}} \int_{\Omega} \frac{U^*(\omega)V(\omega)}{|\lambda(\omega)|}\, d\omega \tag{2.52}$$

Since $|\lambda(\omega)|$ is a continuous, real valued and non-negative quantity; Bochner's theorem [25], which states that a positive-definite kernel is the Fourier transform of a positive measure, can be used to prove the positive definiteness of 2.51 [6].

Another way of constructing Green's kernel for discrete regularization operators is [23]

$$G(x, x') = \sum_{i=1}^{N} \lambda[n]\phi_n(x)\phi_n(x') \tag{2.53}$$

where $\phi_n\{n = 1, \ldots, N\}$ are the basis of the orthonormal eigenvectors of G corresponding to non-zero eigenvalues $\lambda[n]$ such that $\lambda[n]$ confers the spectrum of G. From [23], it can be easily shown that G satisfies the condition of positive definiteness and the series converges absolutely uniformly since all the eigenvalues of G are positive. As, $G(x_i, x_j) = G(x_j, x_i)$, it also satisfies the symmetry property and Mercer theorem can be applied to prove that G is an admissible support vector kernel [12] and it can be written as a dot product in feature space.

$$G(x_i, x_j) = \langle \phi(x_i).\phi(x_j) \rangle \tag{2.54}$$

Recalling the assumption $D = K$ and utilizing (2.49) leads us to

$$G(x_i, x_j) = \langle (\lambda G)(x_i, .).(\lambda G)(x_j, .) \rangle \tag{2.55}$$

i.e. G is a support vector kernel corresponding to regularization operator λ. To illustrate the rationale behind this discussion, examples of some of the commonly used support vector kernel are taken and the discussion presented in literature on their regularization properties is summarized. Assume the magnitude spectrum of regularization operator [6, 19]

$$|\lambda(\omega)| = |\sigma| e^{(\frac{-\sigma^2\omega^2}{2})} \tag{2.56}$$

Kernel Function	$k(x, x')$	$\lvert\lambda(\omega)\rvert$
Gaussian RBF	$e^{-\lVert x-x'\rVert^2/2\sigma^2}$	$\lvert\sigma\rvert e^{(-\sigma^2\omega^2/2)}$
Exponential RBF	$e^{\lvert x-x'\rvert}$	$1/(1+\omega^2)$
B_n-Spline	$\otimes_{i=1}^{N} I_{[-0.5,0.5]}$	$\prod_{i=1}^{N} sinc(n+1)(\omega_i/2)$
Dirichlet	$\sin(2n+1)\frac{x}{2}/\sin(x/2)$	$\frac{1}{2}\sum_{i=-n}^{n}\delta_i(\omega)$

Table 2.1: Regularization properties of commonly used support vector kernels

from (2.51) Gaussian RBF kernel is obtained, i.e.

$$k(x, x') = e^{(\frac{-\lVert x-x'\rVert^2}{2\sigma^2})} \tag{2.57}$$

It can be inferred that using Gaussian RBF kernel in SV algorithm will have the effect of using a regularization network with the smoothness properties given by (2.56). Similarly a magnitude spectrum of the form [7, 19]

$$|\lambda(\omega)| = \frac{1}{1+\omega^2} \tag{2.58}$$

will lead to exponential RBF kernel

$$k(x, x') = e^{-|x-x'|} \tag{2.59}$$

Table 2.1 shows the regularization properties of some of the commonly used support vector kernels [6].

Equation (2.53) can also be utilized to obtain periodic kernels for given regularization operators. For example, as described in [6], periodic Gaussian kernel can be obtained by using

$$\lVert \lambda f \rVert^2 = (\pi)^{-1} \int_{[0,2\pi]} \frac{\sigma^{2n}}{n! 2^n} (O^n f(x))^2 dx \tag{2.60}$$

as the regularization operator, where $O^{2n} = \Delta^n$ and $O^{2n+1} = \nabla\Delta^n$ such that Δ and ∇ are Laplacian and gradient operators, respectively. Such an operator has eigenvalues $\lambda[n] = e^{n^2\sigma^2/2}$ and taking Fourier basis $\{1/2\pi, \sin(nx), \cos(nx), n \in N\}$ as corresponding eigenvectors the problem is simplified to

$$k(x, x') = \sum_{n=1}^{M} e^{-(n^2\sigma^2/2)} (\sin(nx)\sin(nx') + \cos(nx)\cos(nx'))$$

$$k(x, x') = \sum_{n=1}^{M} e^{-(n^2\sigma^2/2)} \cos(n(x - x')) \tag{2.61}$$

Although the kernel given by (2.61) assumes a general low pass smoothing assumption, further capacity control can be achieved by restricting the summation to different eigensubspaces with different values of M. Excluding the eigenfunctions that correspond to high frequencies would result in increased smoothness, thereby decreasing the system capacity. A general low pass smoothing functional is a good choice if there is no prior information available about the frequency distribution of the signal to the predicted. However, (2.61) can be seen as a reasonable choice for building kernels if there is some prior information available about the magnitude spectrum of the signal to be approximated by utilizing the concept of matched filters [6].

Chapter 3

The Problem Statement and Literature Review

3.1 SVM for Classification and Pattern Recognition

This section presents the literature survey projected towards the application of SVM algorithm for classification and pattern recognition tasks.

3.1.1 State of the Art in Support Vector Classification

For the purpose of classification and pattern recognition, SVM have been applied successfully in a variety of applications. In the literature, SVM have been used for handwritten digit recognition by Cortes et al. [3] on US Postal Service benchmark dataset containing 9300 patterns and the NIST (National Institute for Standards and Technology) benchmark dataset containing 70,000 patterns. The study provides a fair performance comparison of SVM against neural networks and decision trees. Scholkopf et al. [48] used SVM with polynomial kernel, Gaussian RBF kernel and neural network kernel function on US Postal Service database to compare the performance of SVM trained with different kernel functions with that of neural networks. The study shows that SVM solution constitutes less than 4% of the whole training set, thereby highlighting the strong sparseness property of SVM. Burges et al. [50] proposed reduced set support vector method for improving the speed and virtual support vector method for better generalization performance. Experimental results were obtained on NIST dataset of handwritten digit recognition. Blanz et al. [51] used SVM

for view-based object recognition and compared the performance of the proposed technique with that of oriented filters. Schmidt [52] has used SVM for speaker recognition task. The performance of the proposed technique is compared with that of conventional Bayesian classifiers over the standard phone line conversational speech data. Osuna et al. [53] employed SVM for face detection in images. The study proposed a new decomposition algorithm for the solution of SVM quadratic optimization problem which guarantees the global optimality and can handle very large data sets. Joachims [54] applied SVM for text categorization tasks and compared the results obtained by SVM, k-NN classifier, decision trees and Rocchio algorithm. El-Naqa et al. [55] utilized SVM for detection of microcalcifications in digital mammograms for early detection of breast cancer. In most of the above mentioned cases, SVM's generalization performance (i.e. error rates on the unseen test sets) either matches or is significantly better than the other learning machines such as neural networks, decision trees and Bayesian classifiers.

3.1.2 SVM for Environmental Informatics

Environmental informatics is the field of applied computer science that develops and uses the techniques of information processing for environmental protection, research and engineering [56]. Wastewater treatment, i.e. the process of contaminant removal from wastewater is one of the important aspects of environmental informatics. As the regulations for effluent quality are getting more and more stringent throughout North America, advanced wastewater treatment (WWT) modeling techniques are required to achieve better level of nutrient removal in wastewater treatment plants. Wastewater consists of several physical, chemical and biological contaminants. Some of the contaminants of concern in wastewater to be removed are suspended solids, biodegradable organics, pathogens, nutrients, priority pollutants, refractory organics, heavy metals, and dissolved inorganics. Nutrients (i.e. nitrogen and phosphorus) are some of the most problematic contaminants of the wastewater. Both nitrogen and phosphorus are essential nutrients for growth [57, 58]. When discharged in the aquatic environment, these nutrients can lead to the growth of undesirable aquatic life.

When discharged in excessive amount on land, they can also lead to the pollution of groundwater. Phosphorus is essential to the growth of algae and other biological organisms. The usual forms of phosphorus found in aqueous solutions include the orthophosphate, polyphosphate, and organic phosphate [57, 58]. Due to the negative effects of the phosphorus that exists in wastewater, along with the stringent discharge limits imposed on wastewater treatment plants, there has recently been an increasing demand to achieve very low effluent total phosphorus. According to the phosphorus removal requirements that have been imposed (by International Joint Commission's Phosphorus Management Strategies Task Force) in Ontario, the typical effluent concentration limit should be 1.0 mg/L, based on total phosphorus [59].

The process of phosphorus removal in wastewater treatment plants can be performed either biologically or chemically. The dataset is obtained from Ashbridges Bay Treatment Plant, Toronto, which uses the chemical method. Chemicals that are used in chemical phosphorus removal process include metal salts and lime. The most commonly used metal salts are ferric chloride, ferrous chloride and aluminum sulfate. In the aforementioned treatment plant ferrous chloride ($FeCl_2$) is being used for the chemical precipitation of phosphorus in the removal process. The theory of chemical precipitation reactions is very complex. There are many uncertainties that underlie all the chemical reactions. Due to the existence of numerous other particles, concurrent side reactions may happen in wastewater as well [57]. All these uncertainties bring about the necessity of prediction, controlling and therefore some kind of intelligent system.

Previously in the literature, studies have been carried out dealing with applications of artificial neural networks and fuzzy neural networks for modeling biological nutrient removal systems [60], fuzzy-logic based control strategies for biological nitrogen removal and dynamic enhanced biological phosphorus removal [62, 63], fuzzy controller for the level of biogas in the treated wastewater [61], SVM for modeling biological nitrogen removal process [64], whereas no work has been directed towards the chemical processes in wastewater treatment especially chemical phosphorus removal process. In light of the successful application of SVM reported

in literature, this study uses SVM for modeling chemical phosphorus removal process in wastewater treatment plants in order to evaluate SVM's capability for modeling complex real world systems. Another objective of this study is to compare the performance of SVM algorithms trained with different kernel functions.

3.2 SVM for Regression Estimation and Time Series Prediction

The following section gives an overview of the research work published about the use of SVM for regression estimation.

3.2.1 State of the Art in Support Vector Regression

For the case of regression estimation and time series prediction where the time series prediction is treated as a special case of regression, SVM have been compared against the conventional learning machines on benchmark chaotic time series prediction tasks by Muller et al. [37]. The performance of SVM for the problem of chaotic time series prediction is compared with that of radial basis function (RBF) neural networks on Mackey Glass chaotic benchmark time series and Santa Fe Laser chaotic benchmark time series. The comparison on Mackey Glass was carried out over broad experimental settings; for two different loss functions, i.e. ϵ-insensitive loss and Huber's loss; for two different noise models, i.e. Gaussian noise and uniform noise as well as for five different SNR (signal-to-noise ratio) values. Mukherjee et al. [65] utilized SVM for Mackey Glass, the Ikeda map and the Lorenz chaotic benchmark time series datasets. The study compares the results obtained by using SVM with the results obtained by using polynomial, rational, local polynomial, radial basis functions with multiquadrics as basis function and neural networks. Drucker et al. [66] have used SVM for the problem of multivariate regression estimation on Boston housing benchmark dataset where the performance of support vector regression is compared with that of bagging regression trees and ridge regression. Tay et al. [68] have employed SVM for financial time series prediction. Five real futures contracts collated from the Chicago Mercantile Market

are examined in the study. They are the Standard & Poor 500 stock index futures (CME-SP), United States 30-year government bond (CBOT-US), United States 10-year government bond (CBOT-BO), German 10-year government bond (EUREX-BUND) and French government stock index futures (MATIF-CAC40). The daily closing prices are used as the datasets. The performance of SVM is compared with multi-layer back propagation neural networks (BPNN). The prediction performance is evaluated using the following statistical metrics: the normalized mean squared error (NMSE), mean absolute error (MAE), directional symmetry (DS) and weighted directional symmetry (WDS). Yang et al. [64] used RBF kernel with least squares SVM for modeling biological nitrogen removal process in wastewater treatment plants. The wastewater treatment plant simulation and optimization software, GPS-X, is used to create virtual plant layout and simulated data. Chen et al. [69] proposed the use of SVM for electricity load forecasting in 2001 EUNITE competition, where the proposed SVM solution won the competition. For rainfall forecasting, SVM have been utilized by Sivapragasam et al. [71]. The proposed technique is applied to Singapore rainfall data. Liu et al. [70] have used SVM for natural gas load forecasting. The natural gas load data in Xi'an city for 2001 and 2002 are used in this study to demonstrate the forecasting capabilities of SVM. The results are compared with that of neural network based model for 7-day ahead forecasting.

In most of the above mentioned cases, SVM have been reported to perform equal or better than the competing methods.

Regarding extensions of the basic SVM algorithm, the basic SVM contain no prior knowledge of the problem. However, several research studies have focused on incorporating prior knowledge about the problem into SVM to achieve better generalization.

3.2.2 State of the Art in Prior Knowledge Based SVM

The intuition of the prior knowledge based SVM comes from the fact that in many applications, prior knowledge about the properties of the function to be learnt is available and might be helpful in the learning task. In the literature, Scholkopf et al. [49] have presented

a method of incorporating prior knowledge about the transformation invariances of a classification problem into SVM. This is done by applying the appropriate transformation to the data points that turn out to be support vectors, thereby generating virtual examples. The experimental results have been obtained on US Postal Service dataset and the NIST dataset. For US Postal Service dataset an improvement of 4.0% to 3.2% in the error rate and for NIST dataset an improvement of 1.4% to 1.0% was recorded. However, The proposed method amounts to running two training sessions causing the algorithm to significantly slow down. Wang et al. [73] has proposed a knowledge based SVM framework for image retrieval tasks. The knowledge is incorporated into SVM optimization as a constraint and a new knowledge-based target function is formulated. The technique has its advantages in image retrieval tasks where the number of available labeled samples is small. Jeyakumar et al. [74] have presented a framework for incorporating prior knowledge in the form of semidefinite inequalities into linear SVM in an automated way to construct knowledge based semidefinite linear programming classifiers. However, the proposed technique has limited applicability since linear SVM are usually extended to non-linear SVM using kernel functions in order to solve complex problems in high dimensional feature space and linear SVM are seldom practically used. Le et al. [78] have proposed a simple method to incorporate prior knowledge in support vector machines by modifying the hypothesis space rather than the optimization problem. The proposed technique shows some improvement in results. Fung et al. have [75, 76] presented a reformulation of linear and nonlinear support vector classifiers incorporated with prior knowledge in the form of multiple polyhedral sets, each belonging to one of two categories. Numerical tests were carried out on the DNA promoter recognition dataset and the Wisconsin prognostic breast cancer dataset. The proposed technique improved the test results by 44.1% on the DNA promoter recognition dataset and 66% on the Wisconsin prognostic breast cancer dataset. The generalization to regression estimation was derived by Mangasarian et al. [77] which studied linear and non-linear support vector machines with prior knowledge in the form of linear inequalities to be satisfied over multiple polyhedral sets. Despite their strong theoretical foundation and remarkable results these methods ex-

perience some shortcomings such as not all rules may be encoded as polyhedral sets and that the resulting optimization problem is fairly complex and difficult to handle. Macliny et al. [79] proposed a simple mechanism for incorporating "advice "(prior knowledge), in the form of simple rules, into SVM based on introducing inequality constraints associated with datapoints that match the advice. Takuro et al. [80] proposed to construct a nonlinear SVM from a set of available prior knowledge items on the problem domain and to determine their weights by using training data set.

3.2.3 Prior Knowledge Based Kernel design for SV Regression

The research work presented in the previous section focuses on extracting different types of knowledge from data and designing prior knowledge incorporated SVM. It has been shown that if appropriately used, prior knowledge can significantly improve the predictive accuracy of learning machine or reduce the amount of training data required. Keeping in mind the fact that kernel functions in non-linear SVM are responsible for the representation of data in the feature space and that SVM's regularization properties are associated with the choice of kernel function used [6, 7]. No work has been directed towards designing problem specific support vector kernel functions based on the prior knowledge about the smoothness properties of the function to be estimated. Hence, the problem of choosing the optimal regularization operator to construct the corresponding support vector kernel function for a given training set still remains unanswered. Another problem associated with the techniques proposed so far is the lack of noise robustness. Since, most of the real world systems are unavoidably contaminated with noise in addition to their intrinsic dynamics [43, 44] and it has been a time-honored tradition [9, 37, 39, 43, 44, 45, 46] to use additive white noise corrupted dataset to evaluate the performance of the proposed technique in order to imitate the real world conditions. In this thesis, the question of choosing the optimal regularization operator to construct the corresponding support vector kernel function that exhibits suitable regularization properties is answered and a mathematical framework is presented for designing prior knowledge based support vector kernel function by using Green's functions

and the concept of matched filters. Apart from its optimal regularization properties, one of the significant features of the proposed technique is its noise robustness since matched filters are known to the be the optimal tool for recovering a signal from additive white noise. The focus is on support vector regression (SVR). Time series prediction is considered as a special case of regression estimation.

Chapter 4

Proposed Scheme and Simulation Results

Having known the weaknesses, for instance non-existence of a unique solution, of many of the existing machine learning technologies such as neural networks and fuzzy logic; the introduction of an innovative machine learning approach by Vapnik [4] provided answers for many of the shortcomings. Support Vector machine (SVM) emerged as a natural consequence of the statistical learning theory and the structural risk minimization principle. Statistical learning theory provided the theoretical foundation for SVM whereas the structural risk minimization offers a structured way to determine the complexity of the model so as to avoid overfitting. The solution of SVM has the properties such as global optimum, sparseness and an upper bound on generalization error. SVM have been reported by several studies [4, 6, 9] to perform equally good or better than other techniques in many applications. Although, the theoretical foundation of SVM seems very appealing, there is a need to carry out more investigations and enhancements (empirical and theoretical) to evaluate and further improve the performance of SVM. This chapter provides answers to the two important questions described in the previous chapter, i.e. to evaluate the performance of SVM for complex real world systems and enhance SVM's performance as well as incorporate noise robustness by designing a novel prior knowledge based SVM kernel. The performance evaluation of SVM also includes assessment of different kernel functions used for the same problem.

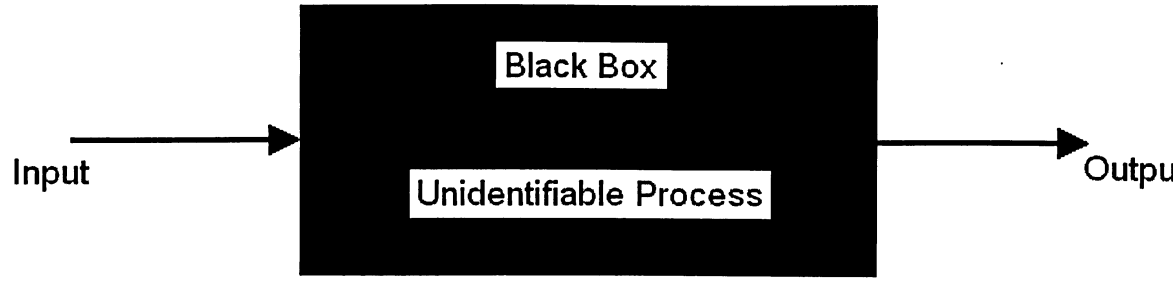


Figure 4.1: A black box system: The internal process is unidentifiable

4.1 SVM for Environmental Informatics

Modeling chemical phosphorus removal process in wastewater treatment is a complex real world problem [57] and no research work has been directed towards addressing this issue. This section is intended towards empirical investigation of SVM by applying them to the modeling of chemical phosphorus removal process in wastewater treatment plants. The performance comparison of different support vector kernel functions is also presented.

4.1.1 Black-box Modeling of Phosphorus Removal Process

This study uses SVM for black box modeling of phosphorus removal process at Ashbridges Bay wastewater treatment plant, in order to evaluate SVM's ability to model complex real world systems. Black box modeling technique refers to the estimation of both the functional form of relationships between the variables and the numerical parameters in those functions from the available data with the assumption that there is no a priori information available about the system (Figure 4.1).

Due to the algorithmic advantages such as faster training and lesser computational complexity, LS-SVM is used (a variation of SVM) in this study. The goal of LS-SVM algorithm is to correctly model the dynamics and the underlying uncertainty in phosphorus removal process at Ashbridges Bay wastewater treatment plant and classify whether or not the concentration of total phosphorus as P in effluent will exceed the limit of 1 mg/L for a given

set of yet to be seen patterns.

4.1.2 Dataset Preparation

The dataset used in this study was obtained from Ashbridges Bay Wastewater Treatment Plant, Toronto. This dataset consists of 123 records. Each record is an observation of the input, control and output variables. Every record represents the average values of the variables for three different measurements taken over a period of one day. The input and control variables used in this study were selected after consultation with senior plant management. Total daily volume treated, peak flow rate, carbonaceous biochemical oxygen demand (CBOD), suspended solids (SS) and total phosphorus as P in influent are used as input variables. Ferrous chloride is used as control variable and is included in the input feature vector for training and testing of LS-SVM classifier. Concentration of total phosphorus as P in effluent is used as the output variable. The dataset was randomly divided into two separate subsets. One of the subsets having 62 examples was used exclusively for training purpose and the other one having 61 examples was used exclusively for testing. Any example from the training set was never used during testing phase and vice versa. A class label $y_i \in$ {-1,+1} was assigned to every output value based on the threshold value of 1.0 mg/L. If the output variable exceeds the threshold, +1 class label is assigned to the output value, otherwise -1 class label is assigned. Class label assignment was done for both of the training and testing datasets before training the LS-SVM classification algorithm.

4.1.3 Simulation Results

The objective of the LS-SVM classification algorithms is to correctly classify whether or not the concentration of total phosphorus as P in effluent will exceed the threshold for a given set of test input patterns. Classification rate was used as a figure of merit. The classification rate was defined as the total number of correctly classified examples divided by the total number of examples classified times one hundred. The results of LS-SVM classification have been obtained using three different kernel functions: Polynomial kernel

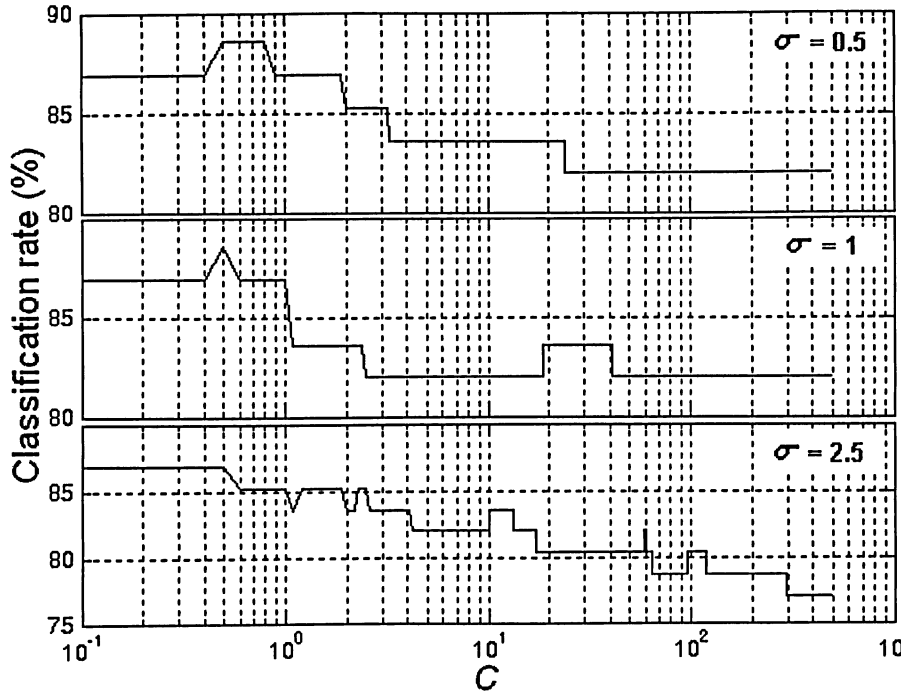


Figure 4.2: Plot of LS-SVM classification rate versus regularization parameter C using RBF kernel with $\sigma = 0.5$, 1 and 2.5

$K(x, x') = (\langle x.x' \rangle + 1)^d$ where d is the degree of polynomial kernel, radial basis function (RBF) kernel $K(x, x') = e^{\|x-x'\|^2/2\sigma^2}$ where σ is the width of RBF kernel and multilayer perceptron kernel (MLP) $K(x, x') = \tanh(k\langle x.x' \rangle + \theta)$. MLP kernel does not satisfy Mercer condition for all k and θ. Figure 4.2 shows the estimated classification rate achieved by LS-SVM classifier using RBF kernel with kernel width $\sigma = 0.5$, 1 and 2.5 against SVM regularization parameter C. The best classification rate achieved was 88.5% when $\sigma = 0.5$ and C is between 0.5 and 0.8. A similar classification rate was achieved when $\sigma = 1$ and C = 0.5. The classification rate dropped to 86.8% when the value of σ was changed to 2.5 and 0.1. For polynomial kernel a consistent classification rate of 86.8% was achieved for a wide range of parameter settings. Although polynomial kernel did not perform as well as RBF kernel, its performance was insensitive for a very wide range of parameter settings. Figure 4.3 represents

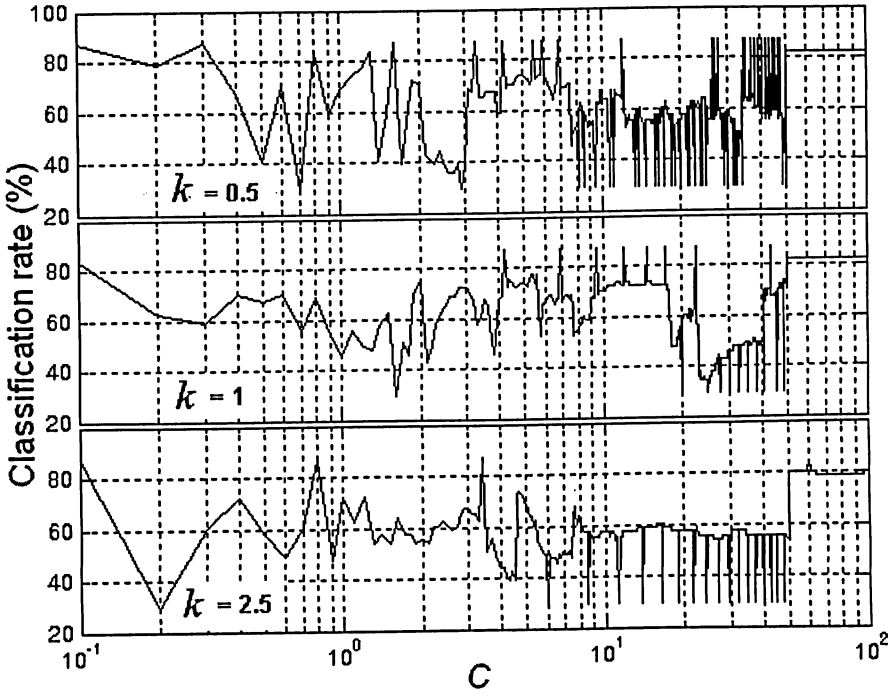


Figure 4.3: Plot of LS-SVM classification rate versus regularization parameter C using MLP kernel with $k = 0.5$, 1 and 2.5

the classification rate obtained by MLP kernel with $k = 0.5$, 1 and 2.5. The value of θ was kept constant at 1. MLP kernel achieved the best classification rate of 86.8% for all the three values of k at different values of C. However, the results obtained by MLP kernel were very sensitive to the parameter settings. Hence, polynomial kernel could be a better choice over MLP kernel. Although all the kernel functions showed a good generalization performance, RBF kernel achieved the highest classification rate. These results are consistent with [6] which shows that SVM's regularization properties are associated with the kernel function used and that the generalization performance of SVM can be affected by the choice of kernel function used for training.

4.2 Prior Knowledge Based Green's Kernel for SV Regression

In order to design the prior knowledge based Green's kernel function that exhibits optimal regularization properties and performs best in the high noise regime, the idea of matched filters along with Green's function is used. Such that, the resulting kernel function's regularization properties match the smoothness properties of the function to be estimated, thereby providing optimal regularization.

4.2.1 Matched filters

Matched filter, maximum signal-to-noise ratio (SNR) filter or sometimes called North filter after its discoverer D. O. North [28], is the optimum time invariant filter among all linear or non-linear filters to recover a known signal from additive white Gaussian noise under the maximum SNR, the likelihood ratio or inverse probability criterion [26]. However, in order to design a matching kernel from prior knowledge, it is sufficient to have an estimate of the magnitude spectrum of the signal to be predicted as prior knowledge about the signal as opposed to the theory of matched filters where complete knowledge of the signal is required to recover the signal from noise. Assume the input signal $f(x)$ in the presence of additive white noise $n(x)$ passing through the matched filter with transfer function $h(x)$. The output

of the filter is given by

$$y(x) = h(x) \otimes (f(x) + n(x)) \tag{4.1}$$

where $\otimes$ denotes the convolution operation. Our aim is to obtain the conditions for which SNR at the filter output takes its maximum value since it is understandable that the probability of recovering a signal from noise is high when SNR is maximum [29]. Maximum SNR is defined to be the ratio of output power due to desired signal $f(x)$ to total mean output noise power

$$SNR = \frac{|y_f(x)|^2}{E\{|y_n(x)|^2\}} \tag{4.2}$$

Assume that $|y_f|^2$ takes its maximum value at $x = x_1$. Then (4.2) can be written as [26, 31]

$$SNR = \frac{|\int_{-\infty}^{\infty}(x_1 - \alpha)h(\alpha)|^2 d\alpha}{\frac{N_0}{2}\int_{-\infty}^{\infty} h^2(x)dx} \tag{4.3}$$

where N_0 is the noise power density. By making use of the Schwarz inequality [30, 31]

$$\left|\int A(x)B(x)dx\right|^2 \leq \int |A(x)|^2 dx . \int |B(x)|^2 dx \tag{4.4}$$

expression in (4.3) can be written as

$$SNR \leq \frac{2}{N_0}\frac{\int_{-\infty}^{\infty} h^2(\alpha)d\alpha \int_{-\infty}^{\infty} f^2(x_1 - \alpha)d\alpha}{\int_{-\infty}^{\infty} h^2(x)dx} \tag{4.5}$$

SNR in (4.5) takes its maximum value when the equality sign holds in (4.4) such that $A(x) = GB^*(x)$, i.e.

$$(SNR)_{max} = \frac{2}{N_0}\int_{-\infty}^{\infty} f^2(x_1 - \alpha)d\alpha \tag{4.6}$$

and

$$h(x) = Gf(x_1 - x) \tag{4.7}$$

Where G is the filter gain constant. For simplicity unity gain is assumed. It is noteworthy in (4.7) that the filter impulse response is independent of noise power density N_0 with the prior assumption of white noise. Secondly, maximum SNR (4.6) is a function of signal energy and is independent of signal shape [26]. This leads to the conclusion that matched filter is the optimal solution for any signal corrupted by any level of white noise.

4.2.2 Mathematical Framework for Prior Knowledge Based Green's Kernel

In order to design a matching kernel based on prior knowledge, it is sufficient to have an estimate of the magnitude spectrum of the signal to be predicted as prior knowledge about the signal, as opposed to the theory of matched filters where complete knowledge of the signal is required to recover the signal from noise. From (4.7) it can be seen that the impulse response of the optimum filter is time reversed signal $f(x)$ with x_1 delay. Nevertheless, in order to obtain matching kernel, the only object of interest is the magnitude spectrum of the matched filter which can be obtained by taking the Fourier transform of $h(x)$ in (4.7) and multiplying it with its complex conjugate

$$H(e^{j\omega}) = \int_{-\infty}^{\infty} h(x)e^{-j\omega x}\,dx = \int_{-\infty}^{\infty} f(x_1 - x)e^{-j\omega x}\,dx$$

$$= e^{-j\omega x_1}\int_{-\infty}^{\infty} f(x_1 - x)e^{j\omega(x_1 - x)}\,dx = F^*(e^{j\omega})e^{-j\omega x_1} \tag{4.8}$$

and

$$|H(\omega)|^2 = H(e^{j\omega})H^*(e^{j\omega}) = |F(\omega)|^2 \tag{4.9}$$

$$|H(\omega)| = |F(\omega)| \tag{4.10}$$

where $H(e^{j\omega})$ is the frequency response of matched filter, $|H(\omega)|$ is the magnitude response and $|F(\omega)|$ is the magnitude spectrum of the matched filter and the signal $f(x)$, respectively. An important note at this point is that (4.10) does not depend on delay x_1 whereas in the case of matched filters it is necessary to have a delay to make the impulse response realizable. Hence, the matching kernel can be obtained by simply calculating the magnitude spectrum of $f(x)$ and utilizing (2.53). As $f(x)$ is the signal to be predicted, it is assumed that its magnitude spectrum does not significantly change from the training targets $y(x)$ in (2.13) and this is the prior knowledge that is acquired from $y(x)$ about $f(x)$ to obtain the Green's kernel. This is a weak condition since many signals with completely different characterization in time domain share the similar magnitude spectrum. Figure 4.4 shows the time and the frequency domain representation of two different signals. Signal in Figure

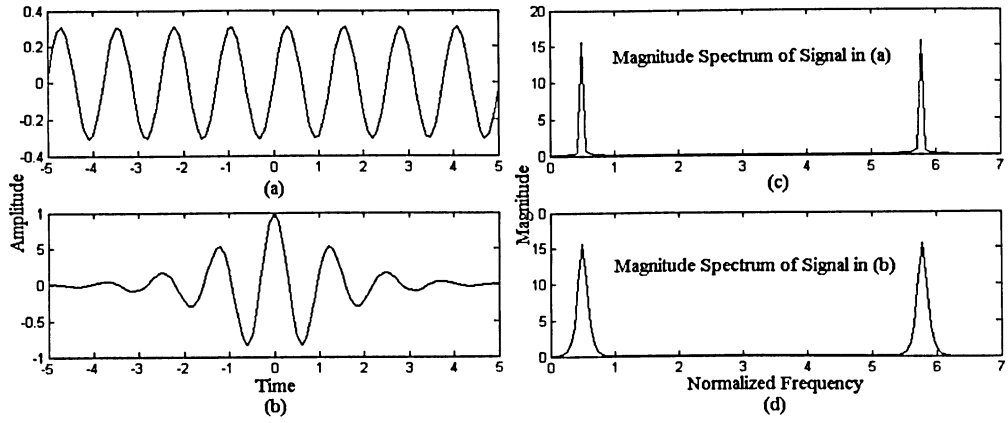


Figure 4.4: Time and frequency domain representation of two different signals.

4.4-a is a sinusoid whereas signal in Figure 4.4-b is the modified Morlet wavelet function. Despite their completely different time domain characterization they share similar frequency localization given by Figure 4.4-c and Figure 4.4-d, respectively.

In order to be capable of using (2.53) to obtain the desired kernel function, its eigenvalues are needed and Fourier basis $\{1/2\pi, \sin(nx), \cos(nx), n \in N\}$ is used as the corresponding eigenfunctions since complex exponentials are the eigenfunctions of any linear time-invariant (LTI) system that includes matched filters and sinusoids can be expressed as linear combination of complex exponentials using Eulers formula [32]. The eigenvalues of an LTI system are given by frequency response $H(e^{j\omega})$ which is a complex-valued quantity [33]. Frequency response can however be written as

$$H(\omega) = |H(\omega)|e^{j\theta(\omega)} \tag{4.11}$$

namely as a product of magnitude response $|H(\omega)|$ and phase response $\theta(\omega)$ [37]. Since the object of interest is the smoothness properties of the kernel and not the phase response, it is adequate to take the magnitude response $|H(\omega)|$ as eigenvalues of the system. Another reason for this is that in order to have positive definite Green's kernel function the eigenvalues need to be strictly positive [23]. Hence, matching Green's kernel function can be obtained

by using (2.53)

$$G(x,x') = \sum_{n=0}^{N-1} |H(\omega_n)|(\sin(\omega_n x)\sin(\omega_n x') + \cos(\omega_n x)\cos(\omega_n x'))$$

$$= \sum_{n=0}^{N-1} |H(\omega_n)| \cos(\omega_n(x-x')) \tag{4.12}$$

where ω_n is the discrete time counterpart of continuous frequency variable ω such that $\omega_n = \frac{2\pi n}{N}$, $0 \leq n \leq N-1$, i.e. normalized to have a range of $0 \leq \omega_n \leq 2\pi$. By making use of (4.10) and ignoring the constant eigenfunction with $n = 0$:

$$G(x,x') = \sum_{n=1}^{N-1} |F(\omega_n)| \cos(\omega_n(x-x')) \tag{4.13}$$

which is a positive definite SV kernel that exhibits matched filter regularization properties given by $|F(\omega)|$. From the algorithmic point of view, only magnitude of the discrete Fourier transform of the training targets need to be computed, with the assumption that the function $f(x)$ to be predicted takes a similar magnitude spectrum with additive noise. To control the model complexity of the system, two variables are introduced to restrict the summation calculation to desired eigensubspaces and (4.13) can be written as

$$G(x,x') = \sum_{n=i}^{j} |F(\omega_n)| \cos(\omega_n(x-x')) \tag{4.14}$$

where i and j are the kernel parameters for Green's kernel similar to the kernel parameters of other SV kernels such as kernel width σ in the case of Gaussian RBF kernel or degree of the kernel d in the case of polynomial kernel. Similar to other SV kernels, an optimal value for i and j is required to achieve the best results.

Analogous to the conventional Gaussian kernel that exhibits Gaussian low pass filter behavior, i.e. $\lambda(\omega) = \exp[\sigma^2\|\omega\|^2/2]$ [6, 7] (recall that the Fourier transform of a Gaussian function is also a Gaussian function) the knowledge based Green's kernel obtained from the eigenvalues of the matched filter exhibits the matched filter properties. This property makes the knowledge based Green's kernel an optimal choice for noise regime since matched filters are the optimal filters for noise corrupted data regardless of the signal shape and the noise

level [26]. Since most of the real world systems are unavoidably contaminated with noise in addition to their intrinsic dynamics [43, 44, 46], it is intended to keep up with the long-established tradition [9, 37, 39, 43, 44, 45, 46] of using benchmark datasets, with additive white noise, to evaluate the performance of the proposed techniques and conduct several experiments, on mostly benchmark datasets ranging from simple regression models to chaotic and non-linear time series with additive white noise, in order to compare the performance of our technique with that of existing support vector (SV) kernels. Nevertheless, the advantage of knowledge based Green's kernel comes at the cost of slightly increased computational complexity. However, for most of the practical signals only a small portion of the whole eigensubspace turns out to be non-zero, thereby lessening the computational load. Another way to overcome this problem is the efficient algorithmic implementation.

A generalization of the kernel function given by (4.14) to N dimensions can be easily made by

$$K(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{d} k_i(x^i, y^i) \tag{4.15}$$

(see [9] for proof of the theorem). Alternatively,

$$K(\boldsymbol{x}, \boldsymbol{y}) = k(\|\boldsymbol{x} - \boldsymbol{y}\|) \tag{4.16}$$

can also be used [6].

4.2.3 Time Series Prediction Models

In this thesis, time series prediction is treated as a special case of function estimation and support vector regression is used to find the underlying relationship between the previous values and the next value of a time series. To predict the future values of the time series, NARX (Nonlinear Autoregressive with eXogenous input) and NOE (Nonlinear Output Error) models are used [38]. NARX is a feedforward model and can be described as

$$\hat{y}_{k+1} = f(y_k,\ y_{k-1}, \ldots,\ y_{k-q}) \tag{4.17}$$

where y_k denotes the true output at time instant k, $\hat{y}_k$ the estimated output at time k and q is the system order. However, NARX model can not be used in the long term prediction

stage since the true output values y_k are not available in the long term prediction. Hence, for the long term prediction of future values, NOE model is used

$$\hat{y}_{k+1} = f(\hat{y}_k,\ \hat{y}_{k-1}, \ldots,\ \hat{y}_{k-q}) \tag{4.18}$$

In contrast to NARX, NOE is a recurrent model with recursion on estimated output variable $\hat{y}_k$. In NOE model, the future output values are iteratively predicted using previously predicted values as input to the model.

4.3 Green's Kernel: Simulation Results

The simulation results presented in this thesis are intended to serve the following purposes:

- To examine the ability of Green's kernel to control the complexity of the system
- To evaluate the performance of Green's kernel and perform a faithful comparison against existing support vector kernel functions.

The datasets used for simulations range from simple regression model to non-linear and high dimensional chaotic time series. The datasets and experimental procedures used in different experiments reflect a wide range of diverse settings; i.e. different noise models, noise levels, loss functions, SVM variations, etc.; in order to perform comparison in a broad perspective. Wherever applicable, results already published in literature for existing support vector kernels have been used as reference for comparison.

4.3.1 Experiment No. 1: Model Complexity Control

The purpose of this experiment is to examine the ability of Green's kernel to control the complexity of an SV model trained with Green's kernel. Sinc function is used as training and testing data. The training data is approximated with different models built only by reducing the size of eigensubspace in kernel matrix computation, i.e. by reducing the value of kernel parameter j while keeping the other SV parameters (C, ϵ), and kernel parameter i constant throughout the experiment. In other words, the complexity of the model is reduced by reduc-

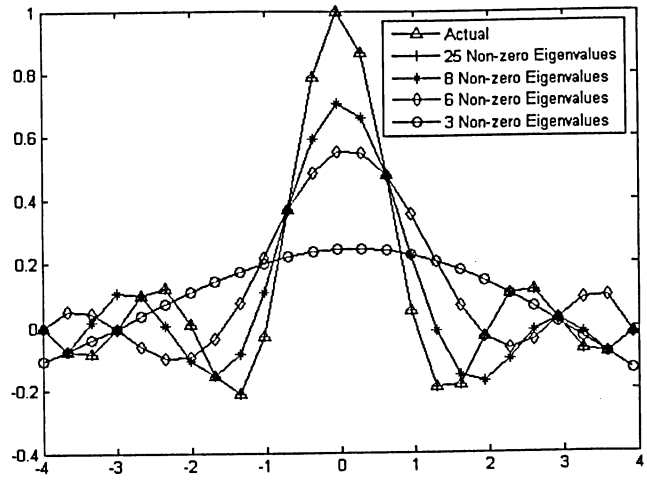


Figure 4.5: SV regression using Green's kernel with the value of $j = 25, 8, 6, 3$.

ing the number of nonzero eigenvalues, thereby removing the high capacity eigenfunctions to obtain a smoother approximation. The value of $i = 1$ was used for all the models. Figure 4.5 shows the regression results obtained for different values of j and highlights the fact that restricting the kernel computation to lower eigensubspace results in a smoother approximation. This shows the ability of Green's kernel to control the regularization properties of SVM.

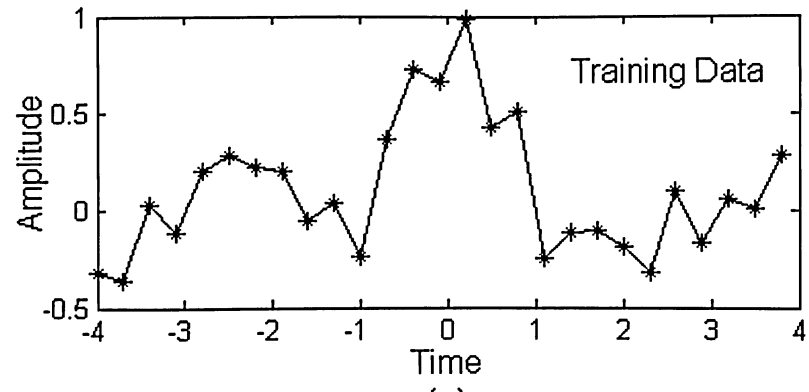
4.3.2 Experiment No. 2: Regression on Sinc Function

In order to validate the performance of our proposed technique sinc function is used. Sinc function given by

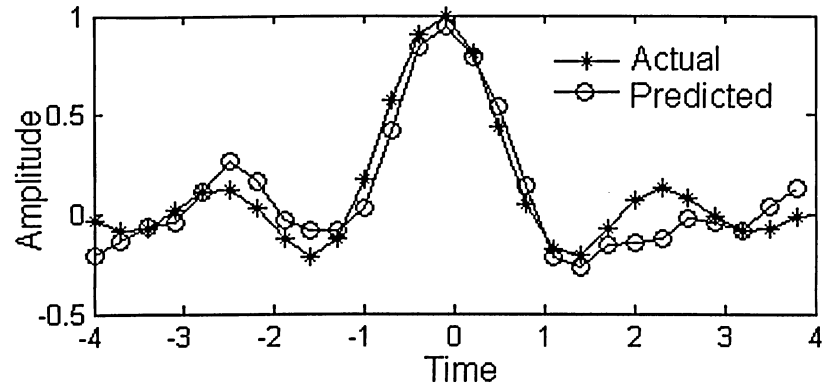
$$\text{Sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{4.19}$$

has become a benchmark to validate the results of SV regression [5, 6, 7, 8, 9, 37, 38]. The training function contains 27 data points with 0 mean, 0.2 variance additive Gaussian white noise. For SVM hyper-parameter (ϵ and regularization parameter C) selection, [21] is followed. Mean squared error (MSE) was used as the figure of merit.

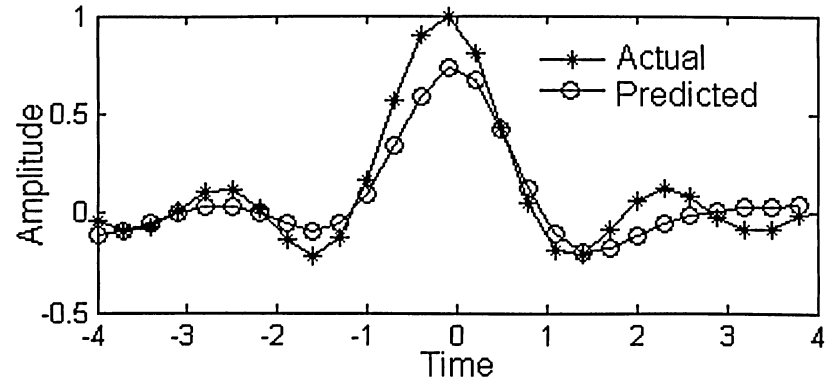
Figure 4.6 shows the training data and the regression results obtained by Green's kernel



(a)



(b)



(c)

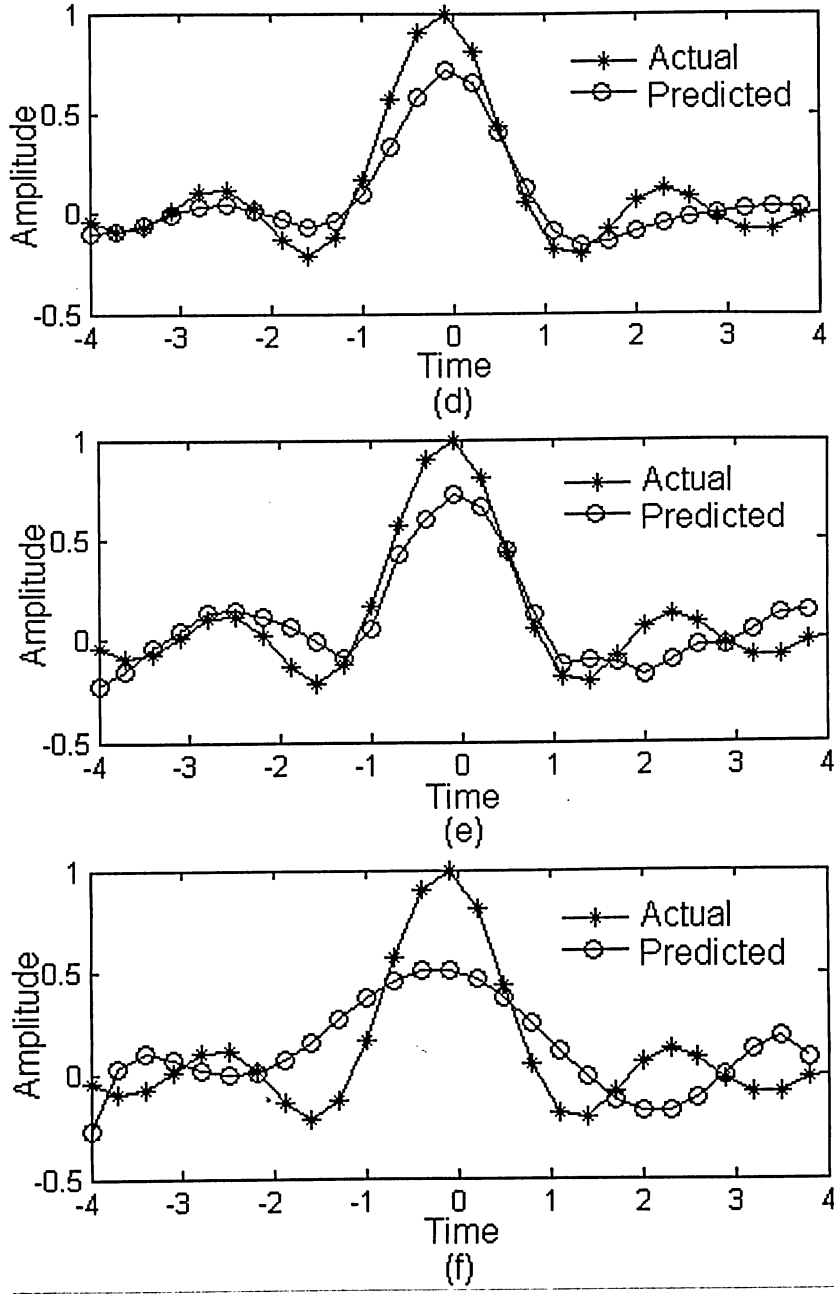


Figure 4.6: SV regression using Green's kernel, (a) Training function, (b) Green's kernel, (c) RBF kernel, (d) Bspline kernel, (e) Exponential RBF kernel, (f) Polynomial kernel.

	Kernel Function	MSE	No. of SV	CPU Time (Sec.)
1	Green's kernel	0.0126	22	0.007
2	Gaussian RBF	0.0152	22	0.034
3	Bspline	0.0163	23	0.17
4	Exponential RBF	0.0214	24	0.035
5	Polynomial	0.0559	24	0.033

Table 4.1: Performance comparison of different kernels for sinc function

and other commonly used SV kernels. Although the results obtained by Gaussian RBF and B-spline kernel are very similar, it is preferred to use Gaussian RBF because only B-spline of odd order are admissible support vector kernels [6] and this restricts the model complexity control.

Figure 4.7 shows the magnitude spectrum of the training signal and the actual sinc function. Magnitude spectrum of the training signal is used as the prior knowledge about the actual signal, i.e. the signal to be predicted and used to construct the matching Green's kernel.

Table 4.1 shows the regression results obtained with different kernel functions. Results indicate that Green's kernel achieved better performance than any other support vector kernel for the given function. The CPU time is the kernel matrix computation time in seconds on an Intel(R) 2.8 GHz, 2 GB Memory system using Matlab 7. The CPU time for other kernel functions was computed using [35]. The lesser computational time of knowledge based Green's kernel is owed to efficient algorithmic implementation.

4.3.3 Experiment No. 3: Regression on Modified Morlet Wavelet Function

Modified Morlet wavelet function is described by

$$\text{Modified Morlet Wavelet Function}, \psi(x) = \frac{\cos(\omega_0 x)}{\cosh(x)} \tag{4.20}$$

This function was selected because of its complex model. A signal of 101 data points with zero mean, 0.3 variance white noise was used as the training set.

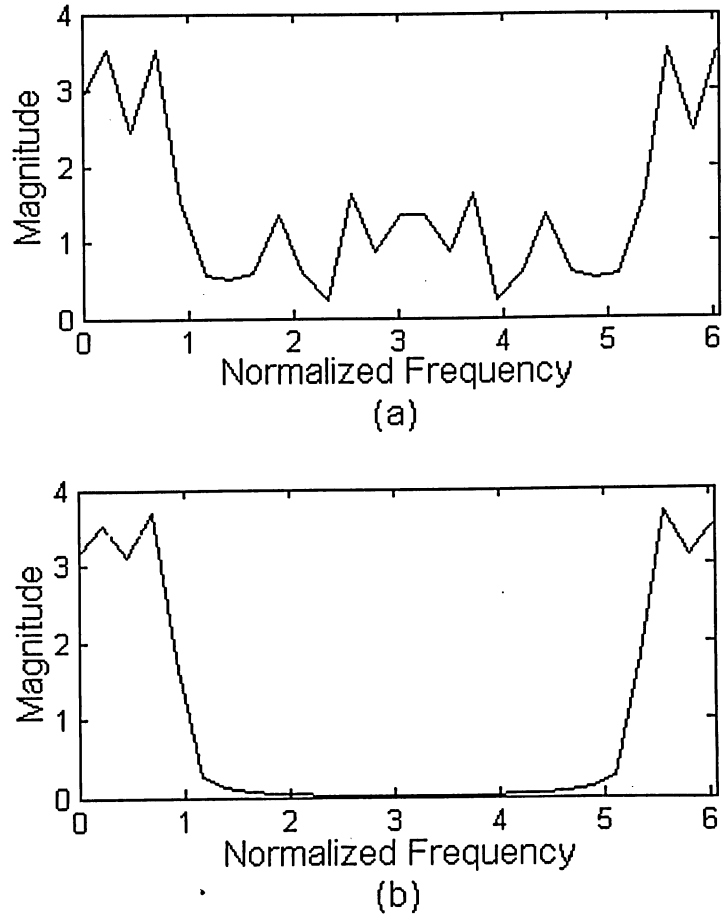
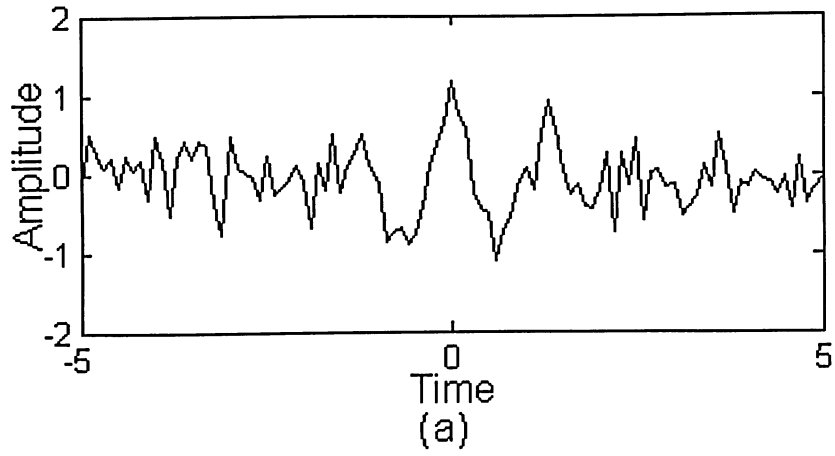
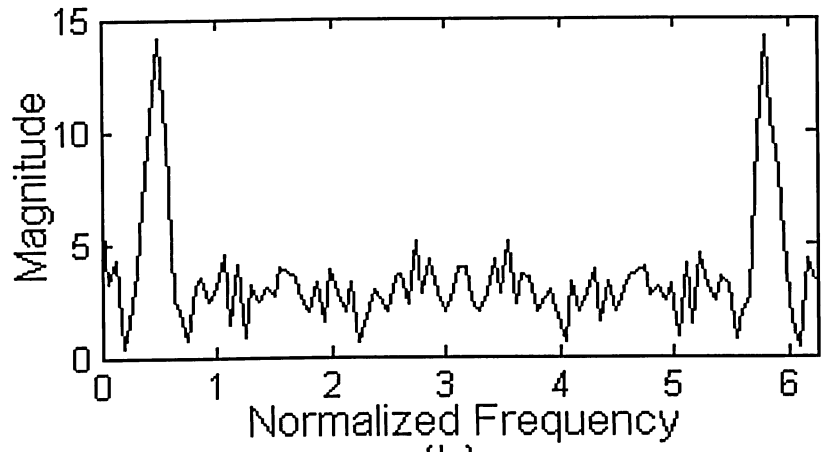


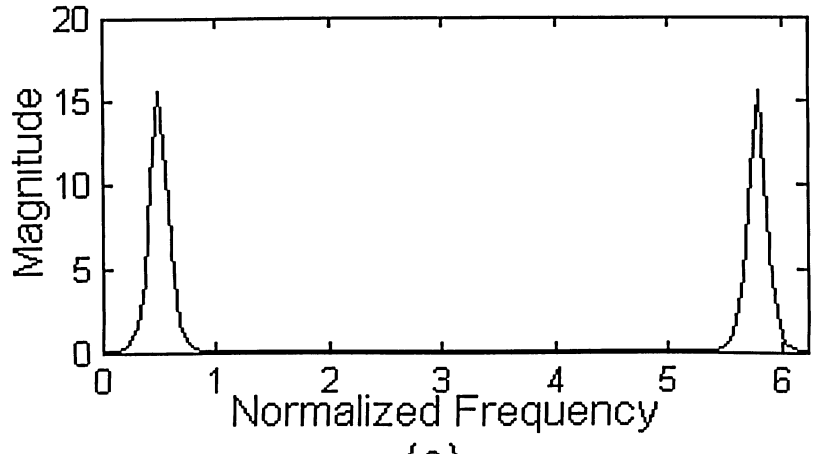
Figure 4.7: Magnitude spectrum of (a) Training signal, (b) Actual sinc function.


Amplitude
2
1
0
-1
-2
-5
0
5
Time

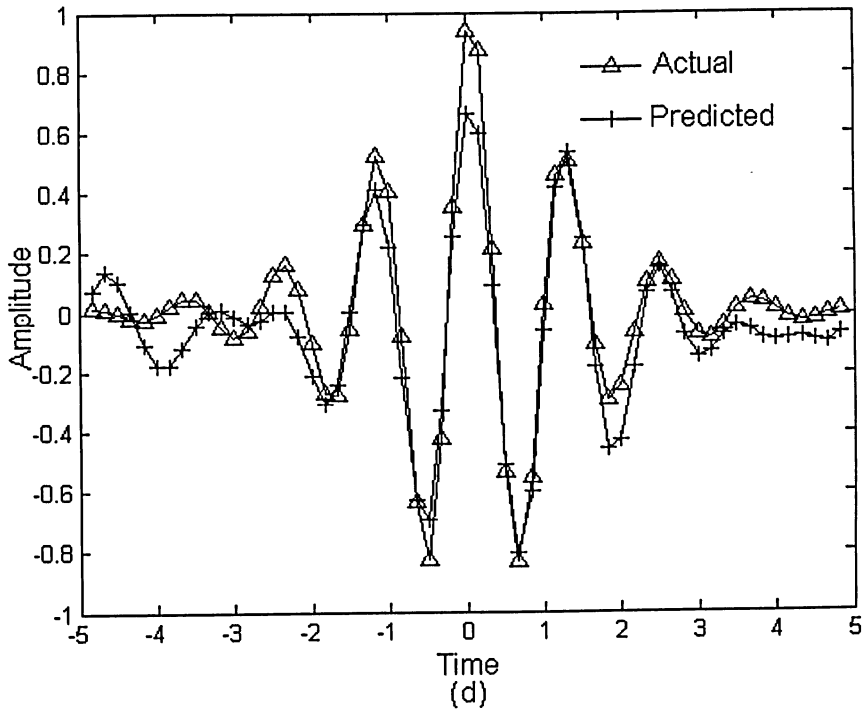
(a)


Magnitude
15
10
5
0
0
1
2
3
4
5
6
Normalized Frequency

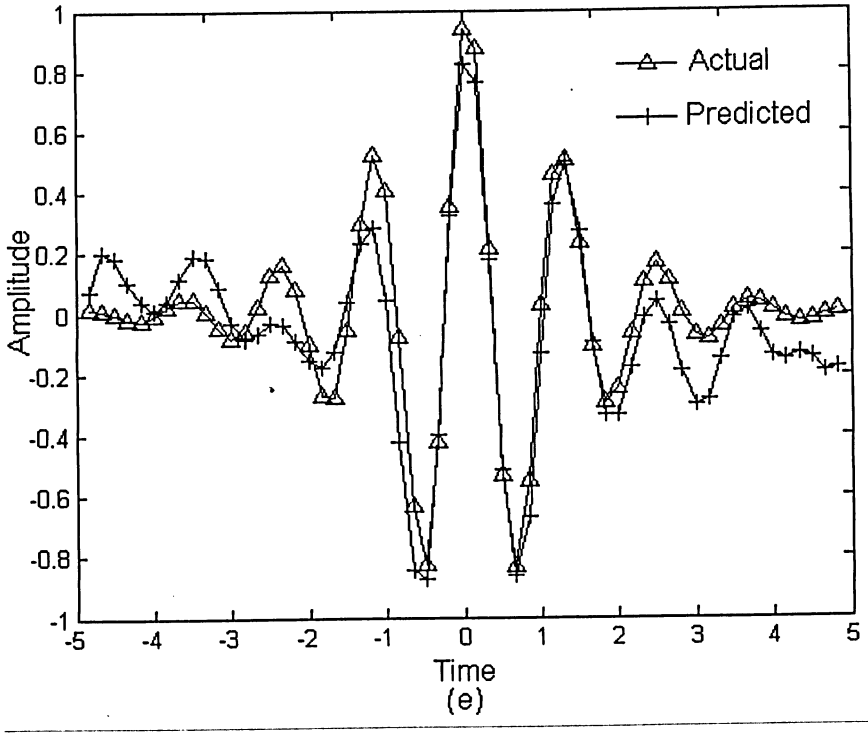
(b)


Magnitude
20
15
10
5
0
0
1
2
3
4
5
6
Normalized Frequency

(c)


Actual
Predicted
Amplitude
Time

(d)


Actual
Predicted
Amplitude
Time

(e)

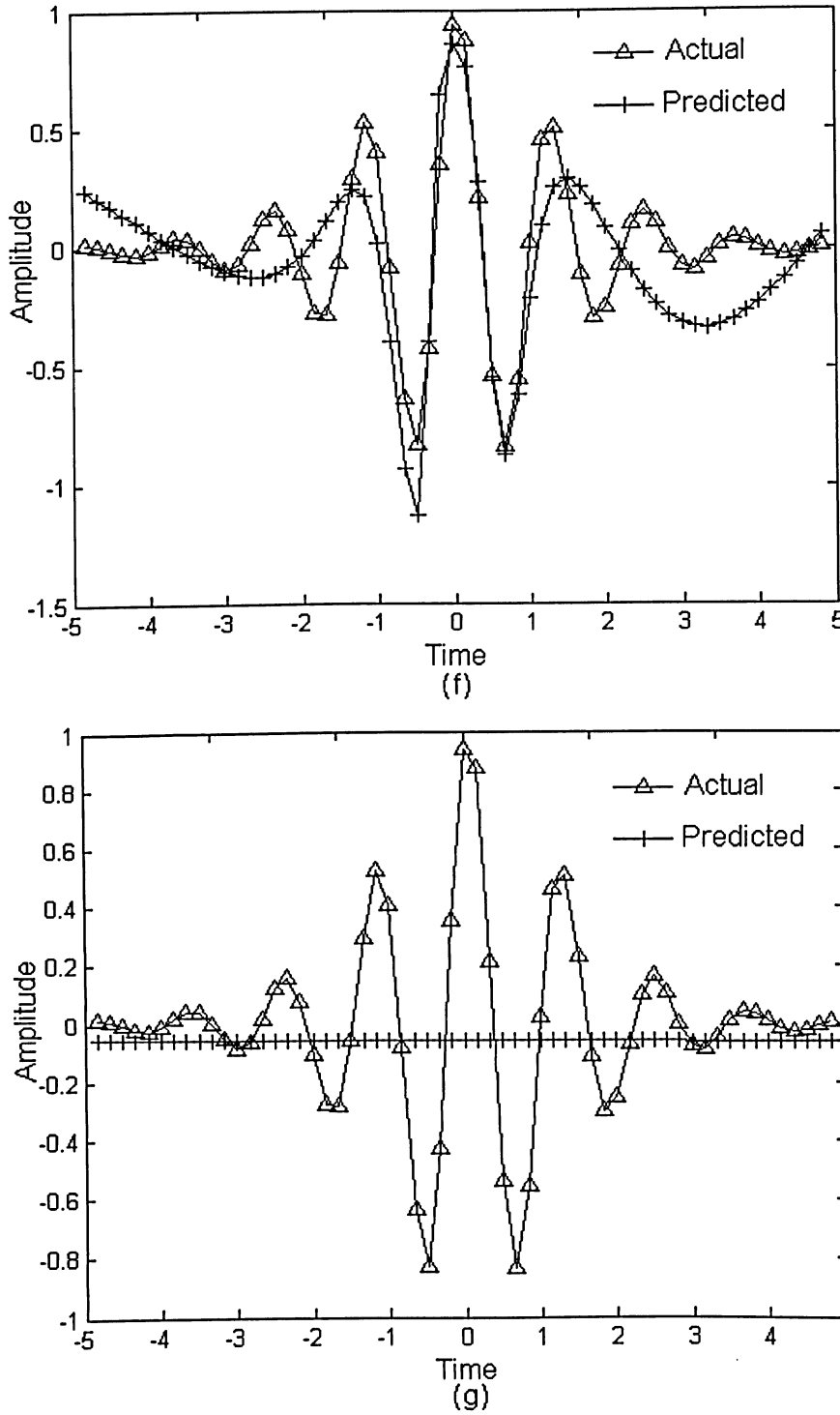


Figure 4.8: (a) Training signal; magnitude spectrum of (b) training signal, (c) actual signal; regression results (d) Green's kernel, (e) RBF kernel, (f) B-spline kernel, (g) polynomial kernel

	Kernel Function	MSE	No. of SV	CPU Time (Sec.)
1	Green's kernel	0.0087	53	0.095
2	Gaussian RBF	0.0272	54	0.45
3	Bspline	1.0173	59	2.37
4	Polynomial	1.0358	59	0.447

Table 4.2: Performance comparison of different kernels for modified Morlet wavelet function

Fig. 4.8 shows the performance of the different SV kernels for modified Morlet wavelet function and the magnitude spectrum of training and actual signals. Although the training function is heavily corrupted with noise, there is still some similarity between the magnitude spectrum of two functions and this similarity is used as the prior knowledge about the problem.

As shown in Table 4.2, again, the Green's kernel performed better than any other kernel for heavily noise corrupted data.

4.3.4 Experiment Nos. 4 & 5: Chaotic Time Series Prediction using SVM and LS-SVM

The purpose of next two experiments is to evaluate the performance of the proposed kernel function against the conventional Gaussian kernel in a broader perspective, i.e. across different noise models, noise levels, prediction steps (short term and long term prediction for time series) and different variations of SVM that use different loss functions and optimization schemes. To perform a faithful comparison, it is intended to use the results already published in literature on a benchmark dataset as the reference point and use the same noise models, noise levels and loss functions as suggested by the corresponding authors. Long term and short term prediction of chaotic time series is considered as a special case of regression. Mackey-Glass; a high-dimensional chaotic benchmark time series, originally introduced as a model of blood cell regulation [36]; is used. Mackey Glass is generated by the following delay

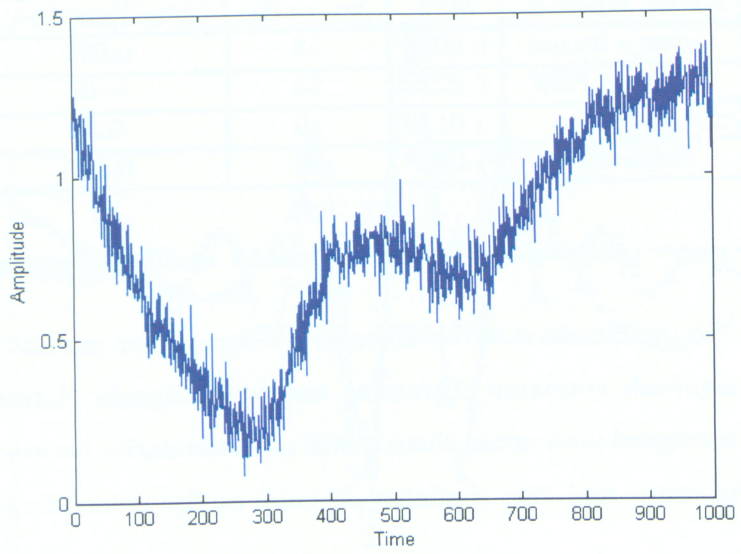


Figure 4.9: Training data with 22.15% additive Gaussian noise.

differential equation [37]:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t) \tag{4.21}$$

with $a = 0.2$, $b = 0.1$ and $\tau = 17$.

The results obtained by the proposed kernel function are compared with: Muller et al. [37] that uses SVM with Gaussian RBF kernel and Zhu et al. [39] that uses LS-SVM with Gaussian RBF kernel for short term (1 step) and long term (100 step) prediction of Mackey-Glass system for different noise models and noise levels. The definitions; experimental and dataset settings; and procedures that have been set by the corresponding authors, are used as closely as possible.

Figure 4.9-4.23 show the performance of Green's kernel using SVM and LS-SVM for different noise levels of Gaussian and uniform noise.

Table 4.3 shows the root mean square error obtained by Green's kernel using SVM and LS-SVM over different settings in comparison to the results reported by [37, 39]. 1S and 100S denotes the 1 step and 100 step prediction of time series. The same definition of SNR

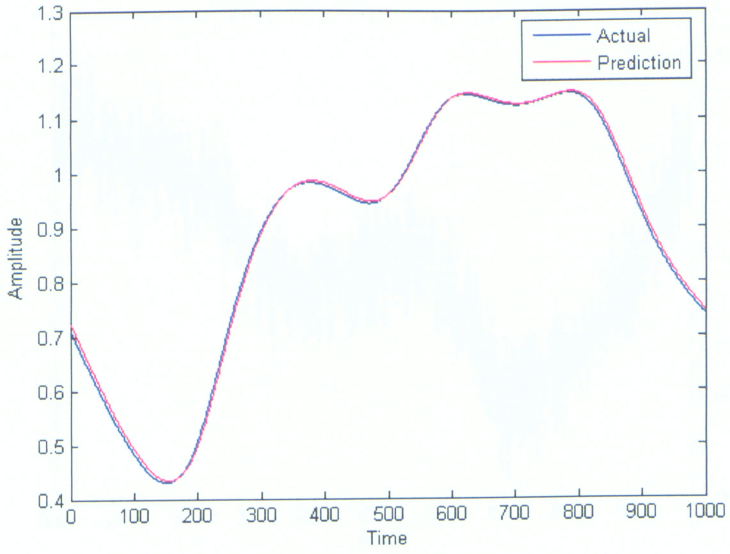


Figure 4.10: SVM one step ahead prediction for data with 22.15% additive Gaussian noise.

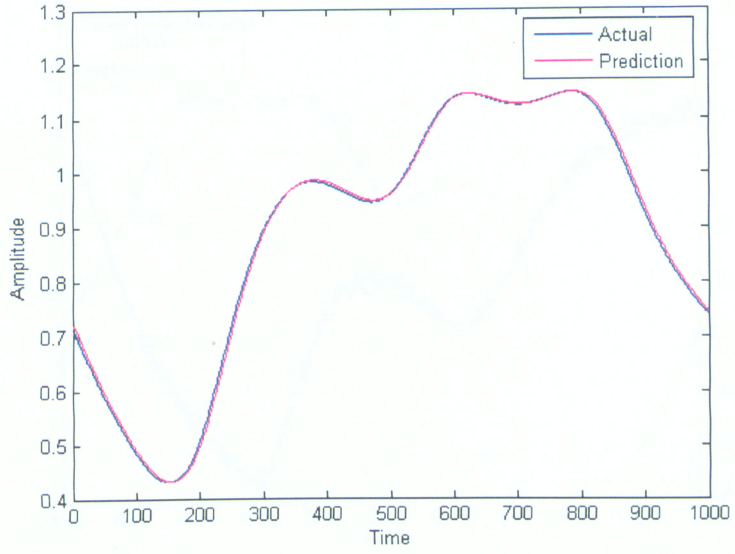


Figure 4.11: LS-SVM one step ahead prediction for data with 22.15% additive Gaussian noise.

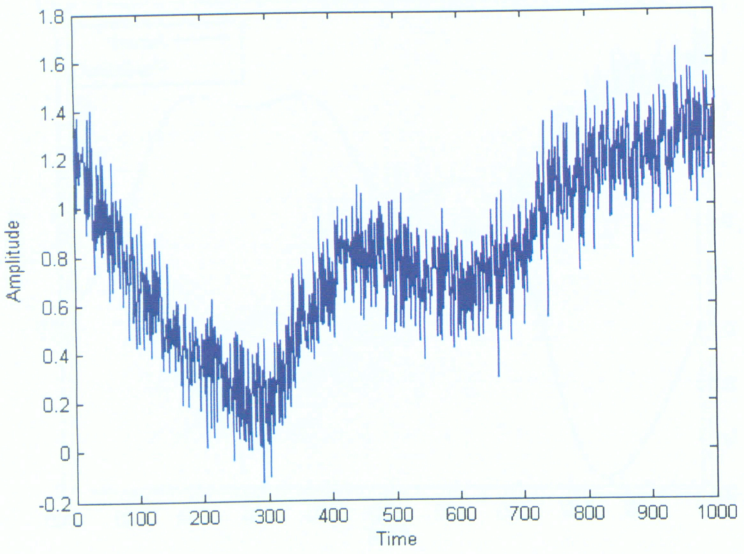


Figure 4.12: Training data with 44.3% additive Gaussian noise.

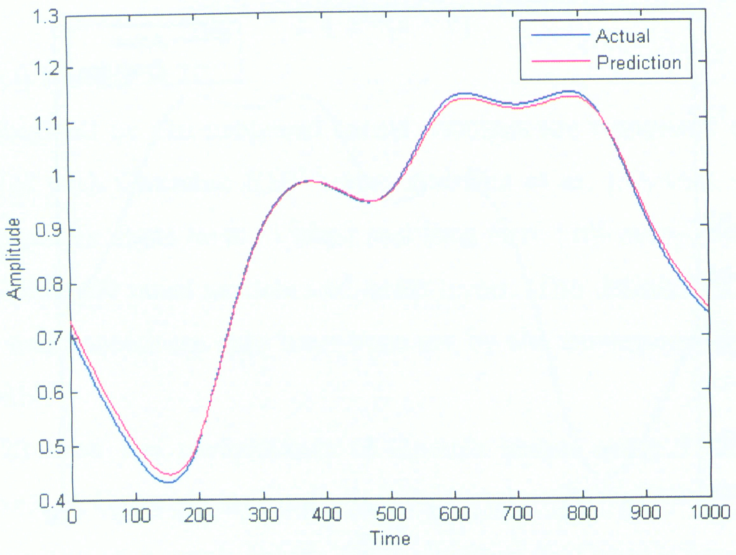


Figure 4.13: SVM one step ahead prediction for data with 44.3% additive Gaussian noise.

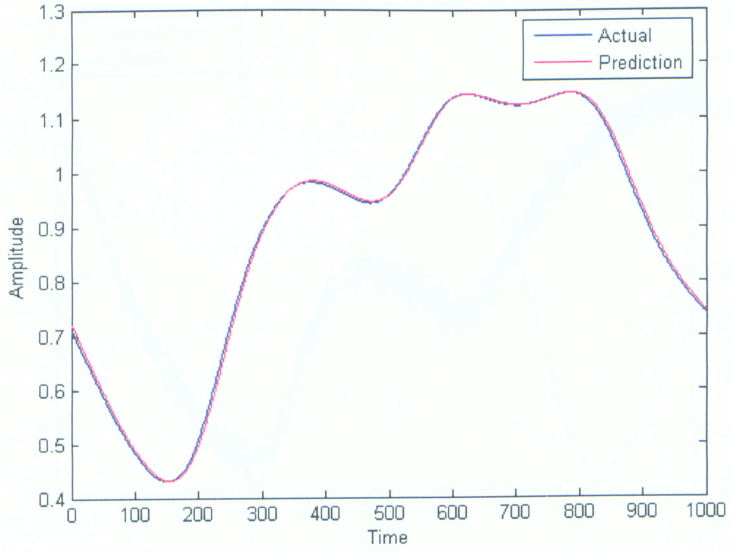


Figure 4.14: LS-SVM one step ahead prediction for data with 44.3% additive Gaussian noise.

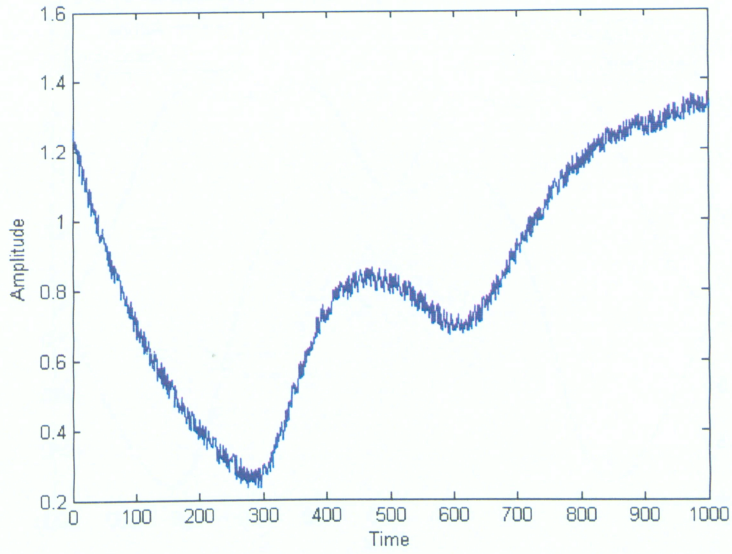


Figure 4.15: Training data with 6.2% additive uniform noise.

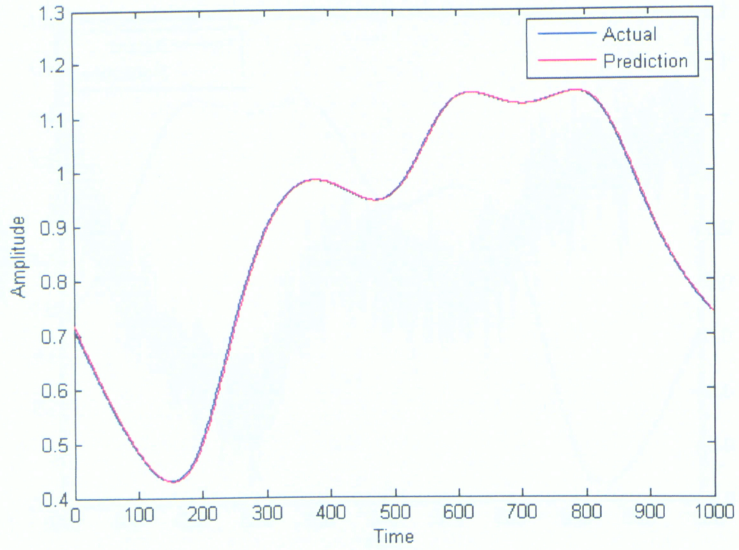


Figure 4.16: SVM one step ahead prediction for data with 6.2% additive uniform noise.

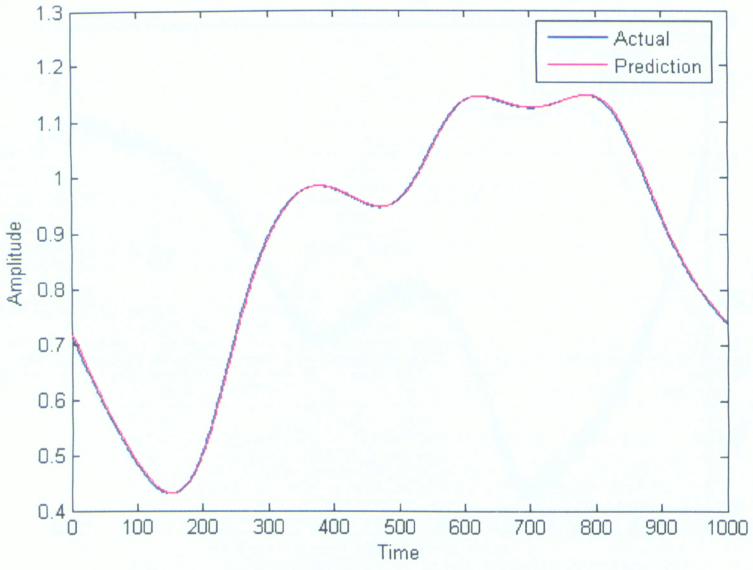


Figure 4.17: LS-SVM one step ahead prediction for data with 6.2% additive uniform noise.

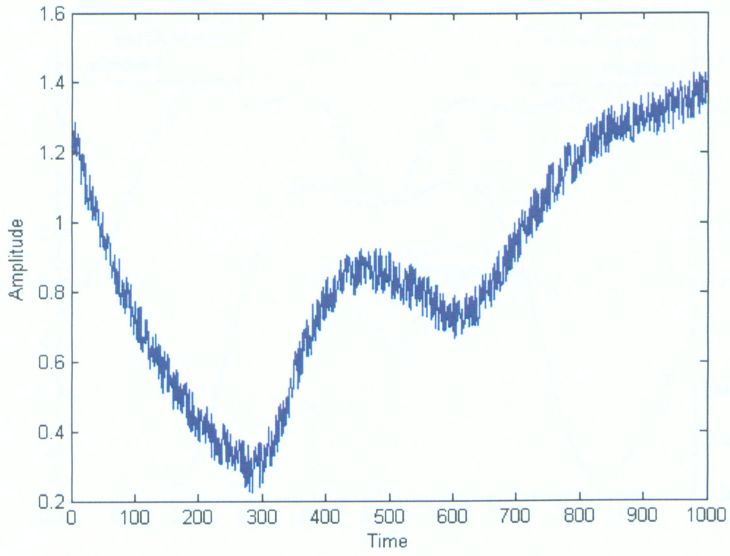


Figure 4.18: Training data with 12.4% additive uniform noise.

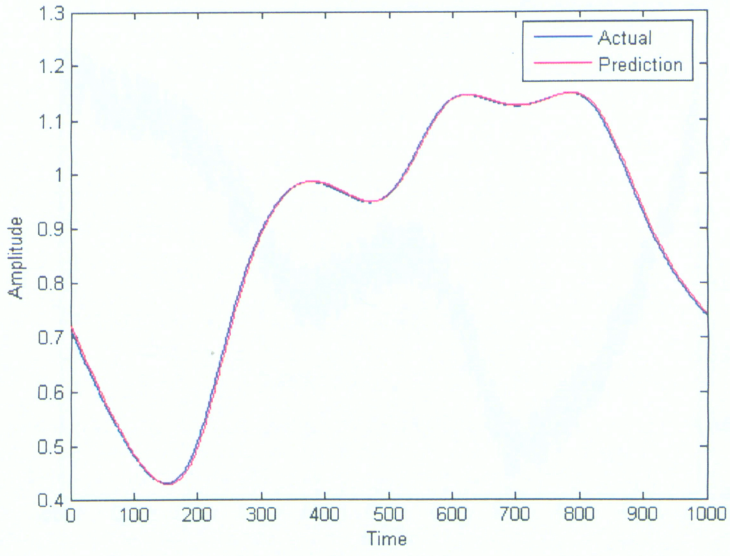


Figure 4.19: SVM one step ahead prediction for data with 12.4% additive uniform noise.

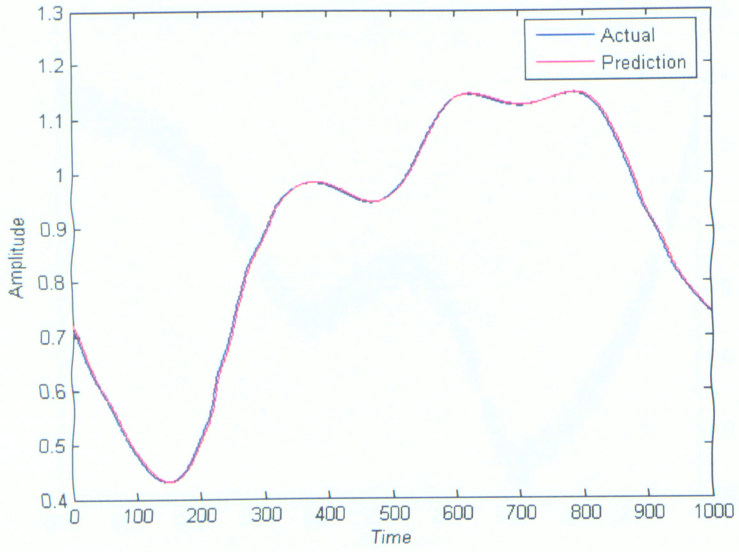


Figure 4.20: LS-SVM one step ahead prediction for data with 12.4% additive uniform noise.

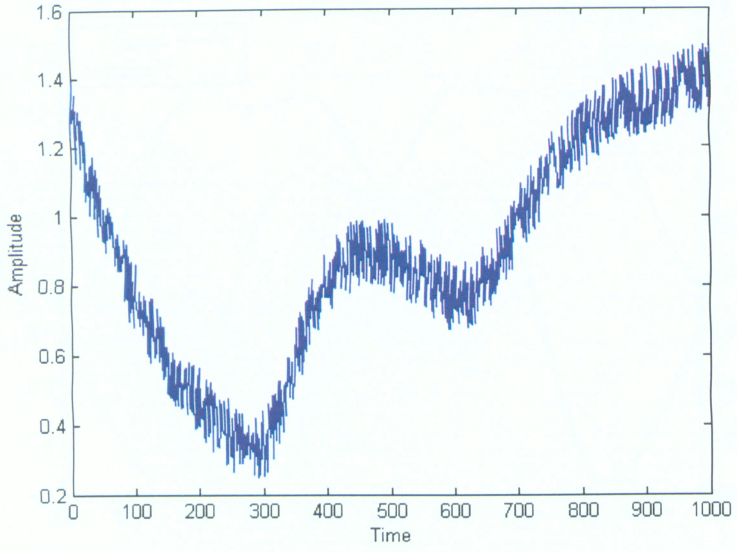


Figure 4.21: Training data with 18.6% additive uniform noise.

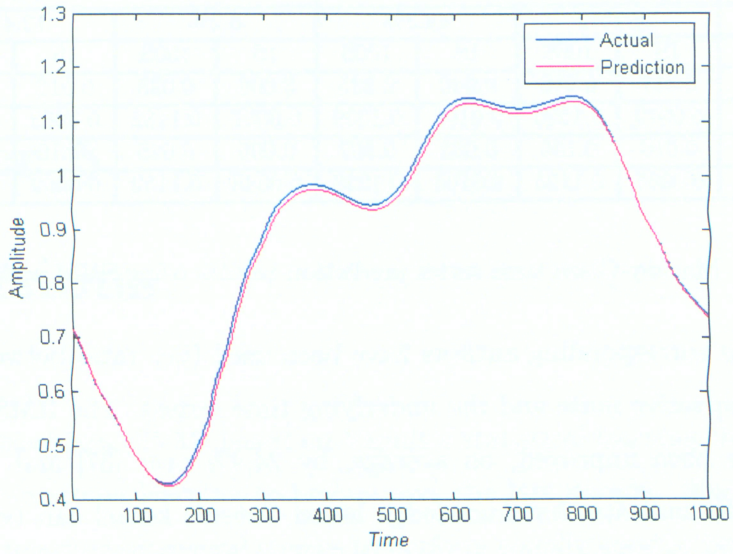


Figure 4.22: SVM one step ahead prediction for data with 18.6% additive uniform noise.

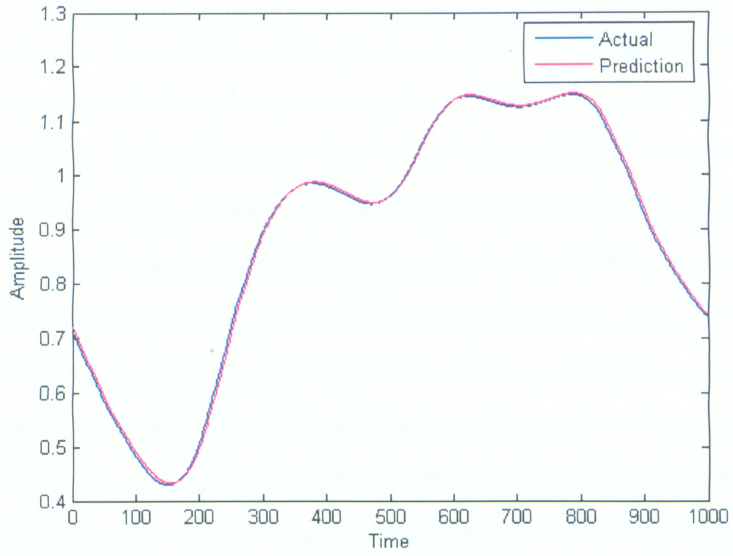


Figure 4.23: LS-SVM one step ahead prediction for data with 18.6% additive uniform noise.

Noise	Normal				Uniform					
SNR	22.15%		44.3%		6.2%		12.4%		18.6%	
Step	1S	100S	1S	100S	1S	100S	1S	100S	1S	100S
RBF (SVM) [37]	0.017	0.218	0.040	0.335	0.006	0.028	0.012	0.070	0.017	0.142
Green's (SVM)	0.0072	0.1220	0.0106	0.1339	0.0057	0.1182	0.0063	0.1229	0.0081	0.1342
RBF (LS-SVM) [39]	0.016	0.165	0.032	0.302	0.005	0.026	0.010	0.064	0.018	0.136
Green's (LS-SVM)	0.0067	0.1195	0.0105	0.1226	0.0048	0.1152	0.0062	0.1189	0.0067	0.1193

Table 4.3: Mackey-Glass time series prediction results using SVM and LS-SVM

as suggested by the corresponding authors have been used (i.e. ratio between the standard deviation of the respective noise and the underlying time series). The RMS errors for SVM and LS-SVM have been improved, on average, by 24.4% over [37] and 18.6% over [39]. Experimental results indicate that knowledge based Green's kernel can be seen as a good kernel choice specially for high noise regime.

Chapter 5

Conclusions

Support vector machines (SVM) based on Vapnik's statistical learning theory have emerged as an innovative machine learning technique over the last decade. Since its introduction, SVM have been reported by several studies to perform equally good or better than the other traditional machine learning methods. SVM based solutions provide many advantages compared to the conventional machine learning approaches, such as global optimum, sparseness of solution and fewer tuneable parameters. SVM's simple geometric interpretation offers fertile grounds for further empirical and theoretical investigations. Given that, basic SVM do not assume any prior knowledge of the learning problem at hand and it has been shown that if properly used, prior knowledge can significantly improve the predictive accuracy of the learning algorithm, one of the important expansions of SVM to enhance the predictive accuracy is the incorporation of prior knowledge into basic SVM. Another imperative facet is the design of a suitable kernel function, since kernel functions are responsible for representation of data in the feature space and SVM's regularization properties are characterized by the choice of kernel function. In this thesis, before addressing the theoretical issues, an empirical investigation of SVM for modeling complex real world problems is presented. The thesis also provides an introduction to the problem of learning from data, reviews the fundamental concepts and mathematical formulations of SVM, regularization networks and kernel functions, presents the literature review, defines the problem statement, sketches the proposed solution and discusses the simulation results.

5.1 Summary of Contributions

This thesis contributes to the area of machine learning in the following aspects:

- A novel prior knowledge based Green's kernel for support vector regression is presented.
- SVM's capability for modeling complex real world problems such as phosphorus removal in wastewater treatment plants is investigated.
- Generalization of Green's kernel to high dimensional time series prediction is derived using NARX and NOE models.

5.1.1 Prior Knowledge based Green's Kernel for Support Vector Regression

The motivation of prior knowledge based Green's kernel comes from the following facts:

- SVM are regularized risk functional and their regularization properties are associated with the kernel function used to map data into feature space.
- A suitable kernel function can provide effective regularization, thereby significantly improving the performance of SVM.
- SVM assume no additional (other than the class of target function, e.g. classification or regression) prior knowledge of the learning problem and if appropriately used, additional prior knowledge can significantly enhance the predictive accuracy of an SVM system.
- Previously, no work has been directed towards designing prior knowledge based kernel functions that exhibit optimal regularization properties for the given training data.
- The existing techniques for prior knowledge incorporation into SVM lack noise robustness.
- Many of the existing support vector kernels such as B-spline and MLP kernels are conditionally admissible.

This thesis presents a mathematical framework for constructing problem specific admissible support vector kernel functions based on the prior knowledge about the problem by incorporating the domain knowledge of the magnitude spectrum of the signal to be predicted into support vector kernels to achieve desired regularization properties using the matched filter theorem and Green's functions. It has been shown that the knowledge based matching Green's kernel exhibits the matched filter behavior. Since the matched filters are known to be the optimal choice for noise corrupted data, one of the key contributions of the proposed technique is its noise robustness, which makes it suitable for many real world system. Experimental results show that the knowledge based Green's kernel has the ability to control the complexity of the system and shows better generalization performance compared to other existing support vector kernels.

For regression estimation, Green's kernel achieved an improvement in error of 17.1% for since function and 68% for modified Morlet wavelet function in comparison to the conventional Gaussian RBF kernel. For chaotic time series prediction, Green's kernel achieved an improvement in error of 24.4% for SVM (average of 10 trials) and 18.6% for LS-SVM (average of 10 trials) over the results already published in literature for Mackey-Glass benchmark system.

5.1.2 SVM for Complex Real World Problems

One of the complex real world problems in the field of environmental informatics is the modeling of chemical phosphorus removal process in wastewater treatment plants. As the regulations on effluent quality with regards to phosphorus are becoming more and more stringent. There has been an increasing demand to develop more sophisticated modeling strategies in order to achieve very low effluent total phosphorus. Previously, no work has been directed towards the solution of this problem by using the machine learning methodology. The objectives of this study are as follows:

- To carry out an empirical investigation of SVM's capability to model complex real world systems.

- To contribute to the field of environmental informatics by constructing a reliable solution to the problem of modeling chemical phosphorus removal process in wastewater treatment plants.

- To compare the performance of different existing kernel functions used for the same problem.

A real dataset from Ashbridges Bay wastewater treatment plant, Toronto has been used in the study. The goal of LS-SVM based learning algorithm is to correctly model the dynamics and the underlying uncertainty in phosphorus removal process at Ashbridges Bay wastewater treatment plant and classify whether or not the concentration of total phosphorus as P in effluent will exceed the limit of 1 mg/L (imposed by International Joint Commission's Phosphorus Management Strategies Task Force) for a given set of test patterns. Simulation results were obtained using Gaussian RBF kernel, Polynomial kernel and MLP kernel. Gaussian RBF kernel achieved the highest classification rate of 88.5%. The results indicate good generalization performance of SVM.

5.1.3 High-dimensional Time Series Prediction

Generalization of Green's kernel to high-dimensional time series prediction is derived using NARX and NOE models. NARX is a feed forward model and can only be used in the training stage. In testing stage, NOE model is used, such that the future output values are iteratively predicted using previously predicted values as input to the model.

5.2 Future Work

Some potential future work directions to enhance the performance of prior knowledge based Green's kernel are as follows:

1. For more complex signals such as non-stationary signals, other transformations besides Fourier transform can be used to incorporate suitable knowledge into support vector kernels.

2. Develop a method to extract information from the past data and compute the memory order of NARX model for time series prediction tasks.

3. Study the generalization performance of SVM kernel functions for situations where training and testing functions have different smoothness properties.

4. Design kernel functions with more meaningful distance calculation criterion than Euclidian distance.

Appendix A

Proof for the Dual Formulation of SVM Primal Quadratic Optimization Problem

This section provides mathematical proof for dual formulation of SVM primal quadratic optimization problem. Although, very brief versions of this proof are already available in the literature, a detailed step by step illustrative proof is presented.

SVM primal objective function (from Equation 2.18) is given by

$$\textit{minimize} \qquad \Re[f] = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\zeta_i + \zeta_i^*) \tag{A.1}$$

subject to the constraints

$$y_i - \langle \boldsymbol{w}.\boldsymbol{x}_i \rangle - b \leq \epsilon + \zeta_i \tag{A.2}$$

$$\langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + b - y_i \leq \epsilon + \zeta_i^* \tag{A.3}$$

$$\zeta_i,\, \zeta_i^* \geq 0 \tag{A.4}$$

This problem can be solved either in the primal space or the dual space by using the Lagrangian function. However, there are several key advantages in solving the problem in dual space. Firstly, the problem is simplified and easier to solve in the dual space. Secondly, due to the fact that primal space does not provide dot product representation between data points, kernel functions can only by used in dual space. Thirdly, the problem formulation in dual space provides support vector representation.

The corresponding Lagrangian is given by

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\zeta_i + \zeta_i^*) - \sum_{i=1}^{N}(\eta_i\zeta_i - \eta_i^*\zeta_i^*)$$

$$-\sum_{i=1}^{N}\alpha_i(\epsilon + \zeta_i + y_i - \langle \boldsymbol{w}.\boldsymbol{x}_i\rangle - b)$$

$$-\sum_{i=1}^{N}\alpha_i^*(\epsilon + \zeta_i^* - y_i + \langle \boldsymbol{w}.\boldsymbol{x}_i\rangle + b) \tag{A.5}$$

Where $\alpha_i^{(*)}$, $\zeta_i^{(*)}$, $\eta_i^{(*)}$ are non-negative. We take partial derivatives of the function in (A.5) with respect to $\boldsymbol{w}$, b, ζ_i, ζ_i^* and put them equal to zero for optimality.

$$\partial_b L = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \tag{A.6}$$

$$\partial_w L = \boldsymbol{w} - \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\boldsymbol{x}_i = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\boldsymbol{x}_i \tag{A.7}$$

$$\partial_{\zeta_i} L = C - \eta_i - \alpha_i = 0 \Rightarrow \eta_i = C - \alpha_i \tag{A.8}$$

$$\partial_{\zeta_i^*} L = C - \eta_i^* - \alpha_i^* = 0 \Rightarrow \eta_i^* = C - \alpha_i^* \tag{A.9}$$

Putting the value of η_i, η_i^* in A.5

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\zeta_i + \zeta_i^*) - \sum_{i=1}^{N}(C - \alpha_i)\zeta_i - \sum_{i=1}^{N}(C - \alpha_i^*)\zeta_i^*$$

$$-\sum_{i=1}^{N}\alpha_i(\epsilon + \zeta_i + y_i - \langle \boldsymbol{w}.\boldsymbol{x}_i\rangle - b)$$

$$-\sum_{i=1}^{N}\alpha_i^*(\epsilon + \zeta_i^* - y_i + \langle \boldsymbol{w}.\boldsymbol{x}_i\rangle + b) \tag{A.10}$$

Rearranging

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i(\zeta_i + \zeta_i^*) - C\sum_i(\zeta_i + \zeta_i^*) + \sum_i \alpha_i\zeta_i + \sum_i \alpha_i^*\zeta_i^*$$

$$-\sum_i \alpha_i\epsilon - \sum_i \alpha_i\zeta_i - \sum_i \alpha_i y_i + \sum_i \alpha_i\langle \boldsymbol{w}.\boldsymbol{x}_i\rangle + \sum_i \alpha_i b$$

$$-\sum_i \alpha_i^* \epsilon - \sum_i \alpha_i^* \zeta_i^* + \sum_i \alpha_i^* y_i - \sum_i \alpha_i^* \langle \boldsymbol{w}.\boldsymbol{x}_i \rangle - \sum_i \alpha_i^* b \tag{A.11}$$

Canceling the common terms

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_i \alpha_i \epsilon - \sum_i \alpha_i^* \epsilon - \sum_i \alpha_i y_i$$

$$+ \sum_i \alpha_i \langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + \sum_i \alpha_i^* y_i - \sum_i \alpha_i^* \langle \boldsymbol{w}.\boldsymbol{x}_i \rangle + b \sum_i (\alpha_i - \alpha_i^*) \tag{A.12}$$

Using (A.6) and factoring

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 - \epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i^* - \alpha_i) - \boldsymbol{w} \sum_i (\alpha_i^* - \alpha_i)\boldsymbol{x}_i \tag{A.13}$$

Using (A.7)

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 - \epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i^* - \alpha_i) - \|\boldsymbol{w}\|^2 \tag{A.14}$$

Simplifying

$$L = -\frac{1}{2}\|\boldsymbol{w}\|^2 - \epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i^* - \alpha_i) \tag{A.15}$$

Using (A.7)

$$L = -\frac{1}{2}\| \sum_{i=1}^{N} (\alpha_i^* - \alpha_i)\boldsymbol{x}_i \|^2 - \epsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i (\alpha_i^* - \alpha_i) \tag{A.16}$$

The final dual formulation of SVM quadratic optimization problem is reached, i.e.

$$maximize \quad \begin{cases} -\frac{1}{2} \sum_{i,j=1}^{N} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \langle \boldsymbol{x}_i.\boldsymbol{x}_j \rangle \\ -\epsilon \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) + \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) y_i \end{cases} \tag{A.17}$$

subject to

$$\sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0 \tag{A.18}$$

$$\alpha_i^{(*)} \in [0, C] \tag{A.19}$$

It is noteworthy that the resulting dual formulation is a function of only the Lagrange multipliers α_i, α_i^*, namely the support vectors and the dual variables have been eliminated. Also, this formulation provides the dot product representation of input data points such that the linear SVM can be readily extended to non-linear SVM using kernel functions.

Appendix B

List of Publications

This section presents the list of publications generated from this research work.

Refereed Journal Papers

- T. Farooq, A. Guergachi, S. Krishnan,"Knowledge Based Green's kernel for Support Vector Machines," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Submitted, 2007.

Refereed Conference Papers

- T. Farooq, A. Guergachi, S. Krishnan,"Chaotic Time Series Prediction using Knowledge Based Greens Kernel and Least Squares Support Vector Machines," in press, *IEEE International Conference on Systems, Man and Cybernetics*, Montreal, October 2007.

- T. Tabatabaei, T. Farooq, A. Guergachi, and S. Krishnan,"Support Vector Machines based approach for Chemical Phosphorus removal process in Wastewater Treatment Plant," *Canadian Conference on Electrical and Computer Engineering (CCECE)*, Ottawa, May 2006.

Bibliography

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *in Proc. Fifth Ann. ACM Workshop Computational Learning Theory*, D. Haussler, ed., pp. 144-152, 1992.

[2] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," *Advances in Neural Information Processing Systems*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993, vol. 5, pp. 147-155.

[3] C. Cortes and V. Vapnik, "Support vector networks, " *Machine Learning*, 20:1-25, 1995.

[4] V. Vapnik, The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995.

[5] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," *in Advances in Neural Information Processing Systems 9*, M. Mozer, M. Jordan, and T. Petsche, Cambridge, MA: MIT Press, 1997.

[6] B. Schlkopf and A. J. Smola, Learning With Kernels. Cambridge, MA: MIT Press, 2001.

[7] A. J. Smola, B. Schlkopf, and K.-R. Mller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, pp. 637-649, 1998.

[8] A. J. Smola and B. Schlkopf, "A tutorial on support vector regression," Royal Holloway College, Neuro COLT Tech. Rep. TR-1998-030, 1998.

[9] V. Vapnik, Statistical Learning Theory. New York: Wiley, 1998.

[10] M. A. Aizerman, E. M. Braver, L. I. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control*, vol. 25, pp. 821-837, 1964.

[11] N. J. Nilsson, Learning Machines: Foundations of Trainable Pattern Classifying Systems. New York: McGraw-Hill, 1965.

[12] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equation," *Philos. Trans. R. Soc. London*, vol. A-209, pp. 415-446, 1909.

[13] V. N. Vapnik. "An overview of statistical learning theory," *IEEE Trans. on Neural Networks*, vol. 10, pp. 988-999, 1999.

[14] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and other Kernel-based Learning Methods. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[15] A. N. Tikhonov and V. Y. Arsenin, Solutions of Ill Posed Problems. New York: Wiley, 1977.

[16] V. Vapnik, Estimation of Dependencies Based on Empirical Data. Moscow: Nauka, 1979, in Russian; English translation: New York: Springer-Verlag, 1982.

[17] A. N. Tihonov, "Solution of incorrectly formulated problems and the regularization method," *Sov. Math. Dokl.*, vol. 4, pp. 1035-1038, 1963.

[18] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Comput.*, vol. 7, pp. 219-269, 1995.

[19] F. Girosi, M. Jones, and T. Poggio. "Priors, stabilizers and basis functions: From regularization to radial, tensor and additive splines," AI Memo No: 1430, MIT AI Lab, 1993.

[20] A. J. Smola and B. Schlkopf, "On a kernel-based method for pattern recognition, regression, approximation and operator inversion," *Algorithmica*, vol. 22, pp. 211-231, 1998.

[21] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression" *Neural Networks,* 17(1), pp 113-126, 2004.

[22] T.'Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Adv. Comput. Math.*, vol. 13, pp. 1-50, 2000.

[23] G. F. Roach, Green's functions introductory theory with applications. London: Van Hostrand Reinhold Company, 1970.

[24] M. D. Greenberg, Application of Green's functions in science and engineering. New Jersey: Prentice-Hall, 1971.

[25] S. Bochner, Lectures on Fourier integral. Princeton. NJ: Princeton University Press, 1959.

[26] H. J. Blinchikoff, A. I. Zverev, Filtering in the time and frequency domains. New York: John Wiley & Sons, 1976.

[27] J. F. James, A Student's Guide to Fourier transforms with Applications in Physics and Engineering. New York: Cambridge University Press, 1995.

[28] D. O. North, "An analysis of the factors which determine signal/noise discrimination in pulsed-carrier systems," RCA Lab Rep., PTR-6c, Jun. 1943. Reprinted *in Proc. IEEE*, vol. 51, pp. 1016-1027, Jul. 1963.

[29] V. Olshevsky, L. Sakhnovich, "Matched filtering for generalized stationary processes," *IEEE Trans. Information Theory*, vol. 51, pp. 3308-3313, 2005.

[30] G. L. Turin, "An introduction to matched filters," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 311-329, 1960.

[31] B. V. Kumar, A. Mahalanobis, R. D. Juday. Correlation Pattern Recognition. New York: Cambridge University Press, 2005.

[32] J. O. Smith (2006, August). Introduction to digital filters with audio applications. [Online]. Available: http://ccrma.stanford.edu/ jos/filters/

[33] O. Akay, "Linear fractional shift invariant (LFSI) systems," *In Proc. Seventh International Symposium on Signal Processing and Its Applications*, pp. 585-588, 2003.

[34] S. K. Mitra, Digital Signal Processing: A Computer Based Approach. 3rd ed., New York: McGrawHill, 2006.

[35] S. Gunn, "Support Vector Machines for Classification and Regression," Technical Report, Image Speech and Intelligent Systems Research Group, University of Southampton, 1997.

[36] M. C. Mackey and L. Glass. "Oscillation and chaos in physiological control systems," *Science*, Vol. 197, pp. 287-289, 1977.

[37] K. R. Mller, A. J. Smola, G. Rtsch, B. Schlkopf, J. Kohlmorgen, and V. Vapnik, Using Support Vector Machines for Time Series Prediction, In: B. Scholkopf, J. Burges, A. Smola, ed., Advances in Kernel Methods: Support Vector Machine, MIT Press, 1999.

[38] J. A.K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, Least Squares Support Vector Machines. Singapore: World Scientific, 2002.

[39] J. Y. Zhu, B. Ren, H. X. Zhang and Z. T. Deng, "Time series prediction via new support vector machines," *in Proc. First Intl. Conf. of Machine Learning and Cybernetics*, ICMLC, pp. 364-366, 2002.

[40] T. Poggio and F. Girosi, A Theory on Networks For Approximation and Learning. Cambridge, MA: MIT, 1989.

[41] T. Poggio and F. Girosi. "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481-1497, 1990.

[42] R. C. Williamson, A. J. Smola, B. Schlkopf, "Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators," *IEEE Trans. Information Theory*, vol. 47, pp. 2516-2532, 2001.

[43] R. Metzler, Y. Bar-Yam, and M. Kardar"Information flow through a chaotic channel: Prediction and postdiction at finite resolution " *Phys. Rev. E*, vol. 70:026205, 2004.

[44] G. Noltel, A. Ziehe and K. R. Muller, "Noise robust estimates of correlation dimension and K_2 entropy " *Phys. Rev. E*, vol. 64:016112, 2001.

[45] E. S. Chng, S. Chen and B. Mulgrew, "Gradient radial basis function networks for nonlinear and nonstationary time series prediction " *IEEE Trans. Neural Networks,* vol. 7, No. 1, 1996.

[46] D. Lowe and A. R. Webb, "Time series prediction by adaptive networks: A dynamical systems perspective," *IEE Proceedings, Part F: Radar and Signal Processing*, vol 138, pp 17-24, 1991.

[47] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Appl. Comp.*, vol. AC-19, pp. 716-723, 1974.

[48] B. Scholkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings, First International Conference on Knowledge Discovery & Data Mining*, AAAI Press, Menlo Park, CA, 1995.

[49] B. Scholkopf, C. Burges, and V. Vapnik, "Incorporating invariances in support vector learning machines," In C. Malsburg, W. Seelen, J. C. Vorbruggen, and B. Sendhoff, editors, *Artificial Neural Networks ICANN96*, pp. 47-52, Berlin, 1996.

[50] C. Burges, and B. Scholkopf, "Improving the accuracy and speed of support vector learning machines," In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pp. 375-381, MIT Press, Cambridge, MA, 1997.

[51] V. Blanz, B. Scholkopf, H. Bulthoff, C. Burges, V. Vapnik and T. Vetter, "Comparison of viewbased object recognition algorithms using realistic 3d models," In C. Malsburg, W. Seelen, J. C. Vorbruggen and B. Sendhoff, editors, *Artificial Neural NetworksI-CANN'96*, pp. 251256, Berlin, 1996.

[52] M. Schmidt,"Identifying speaker with support vector networks," *In Interface '96 Proceedings*, Sydney, 1996.

[53] E. Osuna, R. Freund and F. Girosi, "Training support vector machines: an application to face detection," *In IEEE Conference on Computer Vision and Pattern Recognition*, pp. 130136, 1997.

[54] T. Joachims, "Text categorization with support vector machines," Technical report, LS VIII Number 23, University of Dortmund, 1997.

[55] I. El-Naqa, Y. Yang, N. Wernick, N. P. Galatsanos and R. M. Nishikawa "A Support vector machine approach for detection of microcalcifications," *IEEE Transactions on Medical Imaging*, vol. 21, No. 12, 2002.

[56] N. M. Avouris and B. Page, Environmental Informatics: Methodology and Applications of Environmental Information Processing. Springer, 2004.

[57] Metcaf and Eddy, Wastewater Engineering-Treatment and Reuse. NewYork: McGraw-Hill, 1991.

[58] M. J. Hammer and M. J. Hammer Jr., Water and Wastewater Technology. New Jersey, Columbus: Prentice Hall, 2003.

[59] N. W. Schmidtke and Assoc. Ltd. And D. I. Jenkins and Assoc. Inc., Retrofitting municipal wastewater treatment plants for enhanced biological phosphorus removal. Canada: Minister of supply and services Canada, 1986.

[60] D. S. Lee amd J. M. Park, "Neural networks modeling for on-line estimation of nutrient dynamics in a sequentially-operated batch reactor," *Journal of Biotechnology*, vol. 75, pp. 229-239, 1999.

[61] O. C. Pires, C. Palma, J. C. Costa, I. Moita, M. M. Alves, and E. C. Ferreira, "Knowledge-based fuzzy system for diagnosis and control of an integrated biological wastewater treatment process," *the 2nd IWA conference on instrumentation, control, and automation*, 2005.

[62] S. T. Yordaova, "Fuzzy two-level control for an aerobic wastewater treatment," *proc. 2nd international IEEE conference*, vol. 1, pp. 348-352, 2004.

[63] S. Marsili and L Giunti, "Fuzzy predict control for nitrogen removal in biological wastewater treatment," *IWA conference on water science technology*, vol. 45, pp. 37-44, 2002.

[64] Y. H. Yang, A. Guergachi and G. N. Khan, "Support Vector Machines for Environmental Informatics: Application to Modelling the Nitrogen Removal Processes in Wastewater Treatment Systems " *Journal of Environmental Informatics*, vol. 7, No. 1, pp. 14-25, March 2006.

[65] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using a support vector machine," *In Proceedings of the IEEE Workshop on Neural Networks for Signal Processing 7*, pp. 511519, Amelia Island, FL, 1997.

[66] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, 9:155161, 1997.

[67] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approxi-

mation, regression estimation, and signal processing," *Advances in Neural Information Processing Systems*, 9:281287, 1996.

[68] F. E. H. Tay, and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega: The International Journal of Management Science*, vol. 29, pp. 309-317, 2001.

[69] B. J. Chen, M. W Chang, and C. J. Lin, "Load forecasting using support vector Machines: a study on EUNITE competition 2001," *IEEE Trans. on Power Systems*, vol. 19, pp. 1821-1830, 2004.

[70] H. Liu, D. Liu, G. Zheng, Y. Liang, and Y. Ni, "Research on natural gas load forecasting based on support vector regression," *WCICA 2004. Fifth World Congress on Intelligent Control and Automation*, pp. 3591A-3595, 2004.

[71] C. Sivapragasam, S. Y. Liong, and M. F. K. Pasha, "Rainfall and runoff forecasting with SSASVM approach, " *Journal of Hydroinformatics*, 3:141-152, 2001.

[72] B. Scholkopf,C. Burges, V. Vapnik, "Incorporating Invariances in Support Vector Learning Machines," *Lecture Notes In Computer Science, Springer Verlag*, Issue 1112, pp. 47-52, 1996.

[73] L. Wang, P. Xue, and K. L. Chan, "Incorporating prior knowledge into SVM for image retrieval," *In Proc. of the 17th International Conference on Pattern Recognition, ICPR 2004*, vol. 2, pp. 981-984, 2004.

[74] V. Jeyakumar, J. Ormerod, and R. S. Womersley, "Knowledge-based semidefinite linear programming classifiers," *Optimization Methods and Software, Taylor & Francis*, vol. 21, pp. 693-706, 2006.

[75] G. M. Fung, O. L. Mangasarian, and J. W. Shavlik, "Knowledge-based support vector machine classifiers," *Advances in Neural Information Processing Systems*, Issue 15, pp. 537-544, 2003.

[76] G. M. Fung, O. L. Mangasarian, and J. W. Shavlik, ". Knowledge-based nonlinear kernel classifiers (Technical Report)," Data Mining Institute, 2000.

[77] O. L. Mangasarian, J. W. Shavlik, and E. W. Wild, "Knowledge-based kernel approximation," *Journal of Machine Learning Research*, vol. 5, pp.1127-1141, 2004.

[78] Q. V. Le, A. J. Smola, and T. Gartner, "Simpler knowledge-based support vector machines," *In Proc. of the 23rd International Conference on Machine Learning*, pp. 521-528, 2006.

[79] R. Macliny, J. Shavlikz, T. Walkerz, and L. Torreyz, "A simple and effective method for incorporating advice into kernel methods," *In Proc. of the 21st National Conference on Artificial Intelligence(AAAI 2006)*, Boston, MA, 2006.

[80] F. Takuro, and I. Toshihide, "Knowledge based support vector machines," *Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 13, pp. 259-267, 2005.