DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PIECEWISE AFFINE AND SUPPORT VECTOR MODELS FOR ROBUST AND LOW COMPLEX REGRESSION

by

Ömer KARAL

May, 2011

İZMİR

PIECEWISE AFFINE AND SUPPORT VECTOR MODELS FOR ROBUST AND LOW COMPLEX REGRESSION

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Electrical and Electronics Engineering, Department of Electrical and Electronics Engineering

by
Ömer KARAL

May, 2011
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**PIECEWISE AFFINE AND SUPPORT VECTOR MODELS FOR ROBUST AND LOW COMPLEX REGRESSION**" completed by **ÖMER KARAL** under supervision of **PROF. DR. CÜNEYT GÜZELİŞ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

[illegible]

Prof. Dr. Cüneyt GÜZELİŞ

Supervisor

[illegible]	[illegible]
Prof. Dr. F. Acar SAVACI	Assoc. Prof. Dr. Halil ORUÇ
Thesis Committee Member	Thesis Committee Member
[illegible]	[illegible]
Prof. Dr. Ömer MORGÜL	Assistant Prof. Dr. Güleser K.DEMİR
Examining Committee Member	Examining Committee Member

[illegible]

Prof. Dr. Mustafa SABUNCU
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my profound gratitude and appreciation to my advisor Prof. Dr. Cüneyt GÜZELİŞ, for his consistent help and attention that he devoted throughout the course of this work. He was always kind, understanding and sympathetic to me. His valuable suggestions and useful discussions made this work interesting for me. I am deeply grateful to him.

Thanks are also due to my thesis committee members, who are Prof. Dr. F. Acar SAVACI and Assoc. Prof. Dr. Halil ORUÇ, for their interest, cooperation, and constructive advice. And also, thanks Prof. Dr. Ömer MORGÜL for his valuable contributions and directions. I would like to thank to Assist. Prof. Dr. Sinan KIVRAK whose guidance, inspiration and prayers, led to this accomplishment.

I am grateful to the staff of Graduate School of Natural and Applied Scienses for in Dokuz Eylül Universty, their moral support and good wishes.

I would like to give thanks to my all colleagues. In particular, I would like to thank my close friend Aykut KOCAOĞLU for his friendship and supports.

Finally, I also wish to express my deepest gratitude to my family being with me all my life and in this situation their sacrifices, prayers, and understanding.

Ömer KARAL

PIECEWISE AFFINE AND SUPPORT VECTOR MODELS

FOR ROBUST AND LOW COMPLEX REGRESSION

ABSTRACT

Function representations defined with a small set of parameters are desirable not only for data and model reduction but also for obtaining signal and system models which work well under the real test data. Function approximation and regression (Both will be used in the thesis interchangeably.) provide function representations which are usually designed based on a given finite set of domain-range samples by a learning algorithm and are aimed to possess good generalization performances for the test data not used in the learning phase. The thesis proposes four different classes of regression models which are based on piecewise affine representations and/or support vector methods.

The first class of the developed models is the support vector regression model class employing ℓ_p with $p \leq 1$ “norms” for model parameter cost in order to reduce model complexity and saturating or linear loss functions for rejection or limiting the contributions of outliers in determination of model parameters. The second proposed class is the ε-insensitive least square support vector regression model which is introduced as an extension of the least square support vector regression for reducing excessive number of support vectors appearing in the support vector approach. The third class of the developed regression models is the piecewise affine support vector regression models which are derived by exploiting the canonical representations of piecewise affine functions and first order B-spline basis functions. The fourth class is the piecewise affine regression models which are designed by input-output clustering minimizing an unsupervised clustering error instead of regression error, i.e. the loss function in support vector regression based models. The proposed models are analyzed in a qualitative way and also in a numerical way, and also compared with the known support vector regression models for test functions and real data.

Keywords: Function representation, support vector regression, loss functions, least square support vector regression, optimization methods, piecewise affine function, piecewise affine kernel, input-output clustering.

GÜRBÜZ VE YALIN REGRESYON İÇİN PARÇA PARÇA DOĞRUSAL VE DESTEK VEKTOR TABANLI MODELLER

ÖZ

Az sayıda parametre ile tanımlanan fonksiyon gösterilimleri, sadece veri veya modellerin karmaşıklığını azaltmak için değil, aynı zamanda gerçek test verileri altında oldukça iyi çalışan işaret ve sistemlerin elde edilmesi için de arzu edilir. Fonksiyon yaklaşımı ve regresyon (Her iki terim de tezde eşanlamlı olarak kullanılacaktır.), genellikle, sonlu sayıda giriş-çıkış örnek verilerinden bir öğrenme algoritması yardımı ile tasarlanırlar ve öğrenme sürecinde kullanılmayan test örnekleri için iyi bir genelleme yeteneği olan fonksiyon gösterilimleri sağlarlar. Tezde, parça parça doğrusal ve/veya destek vektör yöntemlerine dayalı dört farklı regresyon model sınıfı önerilmiştir.

Geliştirilen model sınıfından ilki, model parametrelerinin belirlenmesinde, model karmaşıklığını azaltmak üzere model parametre maliyeti için $p \leq 1$ olacak biçimde ℓ_p "normu" ve model parametrelerinin belirlenmesinde aykırı verilerin katkısını yok etmek veya sınırlamak için doymalı veya doğrusal hata fonksiyonu kullanmaktadır. İkinci olarak önerilen model sınıfı, en küçük karesel destek vektör modelinde karşılaşılan aşırı sayıda destek vektör oluşması problemini gidermek için önerilen ve en küçük karesel destek vektör modelinin bir uzantısı olan ε-duyarsız en küçük karesel destek vektör regresyon model sınıfıdır. Geliştirilen fonksiyon yaklaşım modellerinin üçüncüsü, parça parça doğrusal fonksiyonların yalın gösterilimleri ve B-spline taban fonksiyonlarından esinlenerek türetilen parça parça doğrusal destek vektör modelleridir. Geliştirilen dördüncü sınıf regresyon model sınıfı, destek vektör yaklaşım tabanlı modellerdeki yaklaşım hatası fonksiyonu yerine bir eğiticisiz öbekleme hatasını azaltan giriş-çıkış öbekleme algoritması ile tasarlanan diğer bir parça parça model sınıfıdır. Önerilen yöntemler nitel ve sayısal olarak incelenmiş ve gerçek veri ile bazı test fonksiyonları için bilinen destek vektör regresyon modelleri ile karşılaştırılmıştır.

Anahtar kelimeler: Fonksiyon gösterimi, destek vektör yaklaşımı, hata fonksiyonları, en küçük karesel destek vektör regresyon, eniyileme yöntemleri, parça parça doğrusal fonksiyon, parça parça doğrusal kernel, giriş-çıkış öbekleme.

CONTENTS

CHAPTER ONE
INTRODUCTION

Function representation problem for a function specified by a finite number of samples can be defined as finding an approximate function $f(\cdot): X \to Y$ which fits to a given set $\{(\mathbf{x}^s, \mathbf{y}^s)\}_{s=1}^{L}$of domain-range sample pairs where $\mathbf{x}^s \in X$ denotes the samples of the independent variable and $\mathbf{y}^s \in Y$ denotes the samples of the dependent variable. The first step in the function representation is to choose a model $f_w(\cdot): R^n \to R^m$, more precisely building blocks so called basis functions and the type of combination of building blocks. Model selection is realized based on a priori information about the structure of the function to be approximated. The second step is to determine the parameters $w \in R^p$ of the chosen function model. The determination of the model function parameters is usually done based on minimization of an approximation error measure.

Recently, Support Vector Regression (SVR) for function approximation has been developed by Vapnik (1996) and has been applied to solve regression problems (Vapnik, Golowich, & Smola 1996; Müller & Smola, 1997; Mukherjee, Osuna, & Girosi 1997). The superiority of SVR over ANN models is due to their better generalization ability which is achieved by minimizing not only the training error but also a norm of the model parameters to obtain less complex models. SVR solution is found by minimizing a convex quadratic cost in terms of dual variables corresponding to Lagrange multipliers (Smola & Scholkoph, 1998). The optimal function is represented by the combination of kernel functions and a small subset of all training data called Support Vectors (SV).

A function representation defined with a relatively small number of parameters is needed especially when a large number of data is involved requiring large memory allocation and also time consumption. In other words, function representation is a way of data and model reduction.

In recent years, the sparse representation, which means that the number of basis function of the model is small, in the primal space, is studied for robust function approximation by many researchers (Tibshirani, 1996; Chen, 1995; Chen, Donoho & Saunders, 1995; Olshausen & Field, 1996; Daubechies, 1992; Mallat & Zhang, 1993; Coifman &Wickerhauser, 1992). The sparsity of the function is obtained by using ℓ_1 norm unlike ℓ_2 norm of coefficients in the cost function. As an application of ℓ_1 norm, Zhu, Rosset, Hastie, & Tibshirani (2003) is realized the classification problem by using Support Vector Machine (SVM) with ℓ_1 norm for the cases of redundant features. The thesis presents novel robust and low complex regression models by introducing new kind of linear and saturating or linear loss functions for rejecting or limiting outliers and noises, and ℓ_p with $p \leq 1$ "norms" for model parameters in order to reduce model complexity. Herein, the quotation marks on the norm mean that ℓ_p with $p \leq 1$ is not a norm actually which violates positive homogeneity condition for $p = 0$ and triangle inequality condition for $0 < p < 1$.

Least Square SVR (LS-SVR) which is a modified version of standard SVR introduced by Saunders & Gammerman, (1998) and extended to the weigthed version by Suykens, Brabanter, Lukas, & Vandewalle (2002). In conventional SVR, the ε-insensitive loss function is used as the cost function and it is represented by the inequality constraints. In the LS-SVR, the squared loss function is used as the cost function and the errors terms are represented as the equality constraints and the minimization problem is eventually converted to solving a linear algebraic equation system. Nonlinear identification and modeling, function approximation and optimal control are among the numerous applications of LS-SVR (Goethals, et all., 2005; Espinoza, Suykens, & De Moor, 2004; Espinoza, et al., 2005; Suykens, Lukas, & Wandewalle, , 2000; Jiang, Song, Zhao, Wu, & Liang, 2009; Suykens,et. all 2001; Espinoza, et. all., 2006; Wu, 2006; Pelckmans, Suykens, & De Moor, 2005). However, the LS-SVR case does not provide a sparse representation. To solve this problem, data set is partitioned by Hoegaerts, Suykens, Vandewalle and De Moor, (2005) or hierarchical modeling strategy is applied to data (Pelkmans, Suykens & De Moor, 2005). To provide sparsity in the dual space, the thesis proposes ε-insensitive version of LS-SVR and provides its associated solutions. The thesis further compares the solutions of the proposed ε-insensitive LS-SVR with conventional Least Square Solution (LSR) and SVR solution in a qualitative way.

The outline and contributions of the thesis are summarized in the sequel.

In Chapter 2, a background on function approximation and regression is given. A taxonomy of function representations defined on continuous domains is presented. Several interpolation and approximation models and their associated design procedures are presented in a comparative way.

Chapter 3 presents novel robust and low complex regression models by introducing new loss functions for rejecting outliers, and ℓ_p with $p \leq 1$ “norms” for model parameters in order to reduce model complexity. The chapter begins with a description of support vector regression in the most general case ever known and then presents thesis’s contributions providing sparseness in the primal and dual space.

In Chapter 4, ε-insensitive version of least square support vector regression is developed and its associated solution is compared with standard least square regression and support vector regression in a qualitative way.

In Chapter 5, a new type of kernel which is called piecewise linear kernel where feature space is explicitly given with a piecewise affine mapping from the input space is developed. Chapter 5 also presents how SVR with piecewise affine kernel can be formulated for function approximation. The newly proposed PWA kernel is implemented and compared to the other kernel functions for benchmark data.

In Chapter 6, for PWA function representation, an input-output clustering based design method is proposed and applied to the real ECG data.

Finally, a summary of the contributions of the current work and two possible future research directions are presented in Chapter 7.

CHAPTER TWO
BACKGROUND

A function $f(\cdot): X \to Y$ is a specific relation assigning a unique element from the range set Y for each element of the domain set X. Functions provide a useful mathematical framework for signals and systems in which analysis and design methods are based on a functional form. A function used in defining a signal or a system is, in rare cases, obtained as an analytical expression in terms of the known functions by derivations in the context of underlying physical laws. In most of the cases, functions are given a set of domain-range data pairs $\{(\mathbf{x}^s, \mathbf{y}^s)\}_{s=1}^{L}$ where $\mathbf{x}^s$ denotes the samples of the independent variable $\mathbf{x}$ and $\mathbf{y}^s$ denotes the samples of the dependent variable $\mathbf{y}$. The data pairs are obtained by measurements realized in an experiment or in an observation or by sampling an already known function.

Representing a function, which might be given as a set of data pairs or in any way, in terms of a set of known functions can be called as function representation problem. A function representation defined with a relatively small number of parameters is needed especially when a large number of data is involved requiring large memory allocation and also time consumption. In other words, function representation is a way of data reduction and compression. The noise and outliers which are unavoidable in any data generation process should be taken into account in the function representation. This can be done by employing smooth and less complex functions ignoring outliers and suppressing noise as by not trying to fit all of the given data. Another important point in function representation is to obtain a function which predicts acceptable range values for the data not available in the function representation design phase. That is, the obtained function should have good generalization ability.

Figure 2.1 gives taxonomy of function representations defined on continuous domains in terms of; i) discreteness of the domain set in the original form of the given function, ii) finiteness of the domain set in the original form of the given function, iii) exactness of the resulting function representation iv) the orthogonality of the basis functions used in the resulting function representation, v) locality of the definition region for the basis functions used in the resulting function representation, vi) type of basis functions used in the resulting function representation and vii) type of the error functions used in the approximation.

Function representations such as Fourier, wavelet and Taylor series defined on continuous, i.e. real, domain sets have a great impact on the analysis and design of continuous time/space signals and also systems due to their decomposition properties describing signals and systems as weighted sums of simple signal/system building blocks.

One of the most widely used continuous variable representations is Taylor series expansion. It is a local representation valid for infinitely many continuously differentiable functions and provides a polynomial representation in the neighborhood of a point. The truncated version of Taylor series, so called Taylor formula, gives an exact representation with a remainder for the k th order continuously differentiable functions. k terms other than the remainder in Taylor formula constitute a k th order polynomial of the independent variable and coefficients of the polynomial are the derivatives of the functions evaluated at the considered point and the remainder term is negligible compare to the polynomial terms in the vicinity of the point. The conceptual importance of Taylor series is due to the fact that any smooth function can be represented by a polynomial function around a point of interest. Linearization which is indeed a first order special case of Taylor expansion allows exploiting linear techniques for analyzing systems which are in fact nonlinear.

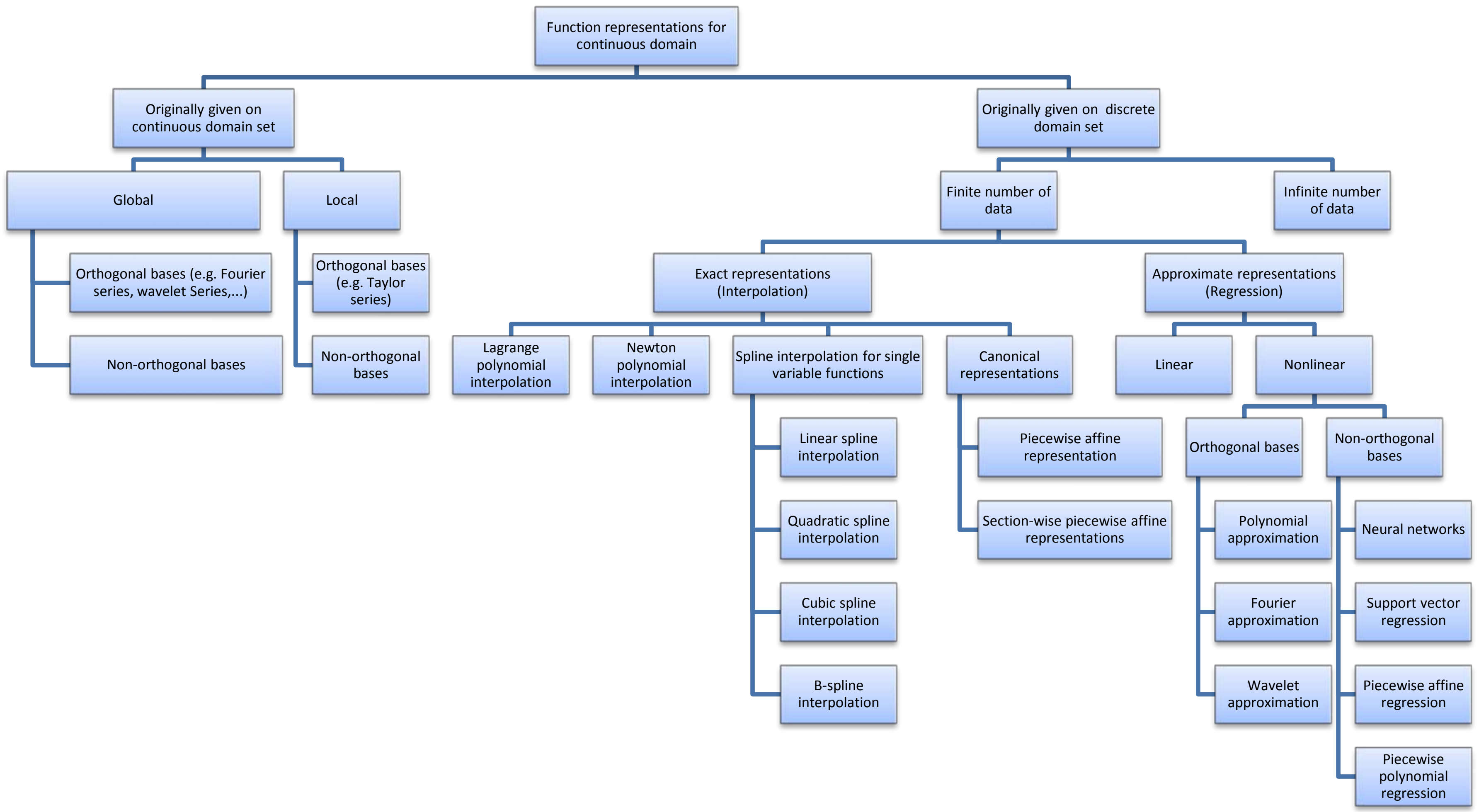

Function representations for continuous domain
Originally given on continuous domain set
Originally given on discrete domain set
Global
Local
Finite number of data
Infinite number of data
Orthogonal bases (e.g. Fourier series, wavelet Series,...)
Non-orthogonal bases
Orthogonal bases (e.g. Taylor series)
Non-orthogonal bases
Exact representations (Interpolation)
Approximate representations (Regression)
Lagrange polynomial interpolation
Newton polynomial interpolation
Spline interpolation for single variable functions
Canonical representations
Linear
Nonlinear
Linear spline interpolation
Quadratic spline interpolation
Cubic spline interpolation
B-spline interpolation
Piecewise affine representation
Section-wise piecewise affine representations
Orthogonal bases
Non-orthogonal bases
Polynomial approximation
Fourier approximation
Wavelet approximation
Neural networks
Support vector regression
Piecewise affine regression
Piecewise polynomial regression

Figure 2.1 A taxonomy of function representations

Fourier series expansion, which is valid for periodical, piecewise continuous and square integrable functions, gives a description in terms of the sinusoidal functions whose frequencies are integer multiple of the frequency of the periodic function. It reveals the frequency content of the function as providing the amplitude and phase information of the constituting frequency components. The spectral coefficients, i.e. the Fourier coefficients, can be found easily by using the orthogonality of the bases functions, i.e. the complex exponential functions associated to the harmonics, in the inner product space defined by an integral. Although Fourier series is a very useful tool for understanding which frequency components exists and what their amplitudes and phases are, it does not give the information related to the time evolution of the frequency content. This insufficiency is removed by the wavelet series expansion, which is another global representation defined by orthogonal bases. The wavelet series provides time-frequency spectra carrying information not only on the frequency spectra but also on its change in time.

The above continuous representations require knowing an analytical representation or a compact form for the original function. However, their truncated versions can well be used for the cases where functions are given by data pairs obtained by measurements in experiments/observations or by sampling from a continuous function for some purposes such as for computer simulations.

Function representation for a given discrete and finite set of data is a problem of finding a function which is defined in terms of a set of suitable functions with some desirable features and fits to the given data together with good prediction ability for data unseen beforehand. Fitting to data might be in an exact sense, i.e. the graph of the function may be required to pass all of the data points. In the other case, fitting is non-exact, i.e. the graph of the function is not required to pass all of the data rather it is required to be as close as possible to all of the data. The latter case is called as function approximation while the former is called as interpolation.

The question of existence of an interpolator for single variable real data has a positive answer: One can always construct a k th order polynomial with real coefficients passing through any given set of $(k + 1)$ data points. As described in Section 2.1.1, such a polynomial can be found by solving the linear algebraic equation system defined by Vandermonde matrix which is usually ill conditioned yielding erroneous polynomial coefficients when taking its inverse. The mentioned numerical inefficiency could be overcome by employing methods not requiring taking inverse of the Vandermonde matrix. Newton and Lagrange interpolations which are presented in Section 2.1.2 and Section 2.1.3, respectively, are two interpolation methods to mention.

Determination of polynomial coefficients is subject to round off and over flow errors when calculating them in any interpolation method. The errors in the coefficients related to the high-order terms yields polynomials too much away from the original function. A solution to overcome this problem is described in Section 2.1.4 which employs piecewise polynomial interpolation techniques including linear, quadratic and cubic splines and also canonical representations for piecewise linear functions.

On the other hand, the interpolation representations are not suitable when there exist noise and outliers and also when there exist large numbers of data requiring large memory allocation and time consumption. Another important point in the exact function representation is to obtain a function which predicts acceptable range values for the data not available in the function representation design phase. That is, the obtained function should have good generalization ability. A more appropriate strategy for these cases is to employ smooth and less complex functions ignoring outliers and suppress noise as by not trying to fit all of the given data. One way to achieve this is to derive a known functional form that minimizes the error between the finite set of data points and the known functional form which is not required to pass all of the data but rather desired to be as much as possible close to the data. This known functional form is called function approximation to the given finite set of data and is detailed in section 2.2.

A part of the representations described above for finite number of data can be extended for infinite number of data case. Linear regression expressed in terms of auto-correlation and cross-correlation functions defined either for deterministic or random but both infinite duration signals constitutes an example in this direction. Infinite number of data case is out of the scope of this thesis which actually focuses on the optimization theory in a deterministic framework as the main mathematical tool. However, the finite number of data restriction is quite acceptable in most applications since signals and systems realized on a machine with finite word length such as today digital computers are defined on discrete and finite domain sets if no transformation is applied to map infinite data into a finite representation in a way.

2.1 Exact representations (interpolation)

Interpolation is the process of finding a function that passes directly through a given finite set of domain-range data pairs. It is well known that one can always construct an k th order polynomial with real coefficients passing through any given set of $(k+1)$ real data points. It should be noted that a polynomial of order less than k could be sufficient when some of the point coincide. There are many different interpolation methods (equivalently saying exact representations) differing from each other either in the calculation of defining coefficients or in the kind of chosen basis function. This section gives the most common ones of these exact representations: Vandermonde Matrix polynomial representation, Newton polynomials, Lagrange polynomials, Spline interpolation and piecewise linear canonical representation.

2.1.1 Determining Coefficients of an interpolating polynomial (Vandermonde matrix)

Polynomial interpolation can be realized for a real function $f_k(\cdot): R \to R$ in a way described by the following theorem.

Theorem 2.1 For any given set of $(k+1)$ data pairs $\{(x^s, y^s)\}_{s=1}^{k+1}$, with $x^s, y^s \epsilon R$ there exist a unique k th order polynomial $f_k(\cdot)$ satisfying

$$f_k(x^s) = y^s, \quad s \in \{1,2,\dots,k+1\} \tag{2.1}$$

$$f_k(x) = w_0 + w_1 x + \cdots + w_k x^k, \tag{2.2}$$

where, the real coefficients of the polynomial in the Equation (2.2) can be calculated by taking the inverse of the *Vandermonde matrix* $\boldsymbol{W}$ given in the Equation (2.3).

$$\boldsymbol{W} = \begin{bmatrix} 1 & (\boldsymbol{x}^1) & \cdots & (\boldsymbol{x}^1)^k \\ 1 & (\boldsymbol{x}^2) & \cdots & (\boldsymbol{x}^2)^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (\boldsymbol{x}^{k+1}) & \cdots & (\boldsymbol{x}^{k+1})^k \end{bmatrix} \tag{2.3}$$

Proof

The conditions in (2.1) lead to the following system of linear algebraic equations in terms of the w_i coefficients:

$$\begin{aligned} &f(x^1) = y^1 = w_0 + w_1(x^1) + \cdots + w_k(x^1)^k \\ &f(x^2) = y^2 = w_0 + w_1(x^2) + \cdots + w_k(x^2)^k \\ &\vdots \\ &f(x^{k+1}) = y^k = w_0 + w_1(x^{k+1}) + \cdots + w_k(x^{k+1})^k \end{aligned} \tag{2.4}$$

It can easily be seen that the system of linear equations in (2.4) has a unique solution if and only if the Vandermonde matrix in (2.3) is invertible and that the polynomial coefficients w_i's can be calculated in a unique way by taking the inverse of $\boldsymbol{W}$**.** The necessary and sufficient condition for the invertibility of the Vandermonde matrix is the distinctness of the data points of x^s's. When some of the data points are identical the system of linear equation of in (2.4) becomes underdetermined and still solvable. Herein the solvability follows from the fact that a function assigns a unique image for two identical points in the domain space. That is, $x^s = x^r \rightarrow y^s = y^r$ for any $s, r \in \{1,2,\cdots,k+1\}$ yielding the s'th and r'th equations in (2.4) are identical. The proof is completed by the observation of that distinctness of x^s is necessary and sufficient to the invertibility of $\boldsymbol{W}$ matrix.

□

2.1.2 Newton polynomial interpolation

Taking inverse of the Vandermonde matrix is numerically an inefficient way of determining coefficients of polynomial interpolation especially for large data sets. This problem could be overcome by employing Newton and Lagrange interpolation methods not requiring taking inverse of the Vandermonde matrix.

In order to get a polynomial interpolation to $(k+1)$ data pairs $\{(x^s, y^s)\}_{s=0}^{k}$ with $x^s, y^s \epsilon R$, one can prefer to use the k th order polynomial form in (2.5) as an alternative to the k th order polynomial $f_k(x) = w_0 + w_1 x + \cdots + w_k x^k$.

$$f_k(x) = b_0 + b_1(x - x^0) + \cdots + b_k(x - x^0)(x - x^1)\cdots(x - x^{k-1}) \tag{2.5}$$

The equivalence of these two polynomial forms of the same degree can be easily seen by observing the solvability of a_i's in terms of b_j's and vice versa:

$$\begin{aligned} w_0 &= b_0 - b_1(x^0) + \cdots + (-1)^k b_k(x^0)(x^1)\cdots(x^{k-1}) \\ &\vdots \\ w_{k-1} &= b_{k-1} - b_k(x^0 + x^1 \cdots + x^{k-1}) \\ w_k &= b_k \end{aligned} \tag{2.6}$$

For a given set of $(k+1)$ data points $\{(x^s, y^s)\}_{s=1}^{k+1}$ with $x^s, y^s \epsilon R$, the coefficients $b_0, b_1, \cdots, b_k$ can be obtained by the following recursive procedure so called “finite divided difference”. The first order finite divided difference is described by:

$$f[x^i, x^j] := \frac{f(x^i) - f(x^j)}{x^i - x^j} \tag{2.7}$$

The following second order finite divided difference represents the difference of two first divided differences.

$$f[x^i, x^j, x^k] := \frac{f[x^i, x^j] - f[x^j, x^k]}{x^i - x^k} \tag{2.8}$$

Similarly, the k th finite divided difference is:

$$f[x^k, x^{k-1}, \cdots, x^0] := \frac{f[x^k, x^{k-1}, \cdots, x^1] - f[x^{k-1}, x^{k-2}, \cdots, x^0]}{x^k - x^0} \tag{2.9}$$

These differences can be used to evaluate the interpolation coefficients of (2.5):

$$\begin{aligned} b_0 &= f(x^0) \\ b_1 &= f[x^1, x^0] \\ b_2 &= f[x^2, x^1, x^0] \\ b_k &= f[x^k, x^{k-1}, \cdots, x^0] \end{aligned} \tag{2.10}$$

Then, the pol ynomial i nterpolation c an be f ound by substituting the obtained coefficients in (2.10) into (2.5):

$$\begin{aligned} f_k(x) = f(x^0) &+ (x - x^0) f[x^1, x^0] \\ &+ (x - x^0)(x - x^1) f[x^2, x^1, x^0] \cdots \\ &+ (x - x^0)(x - x^1) \cdots (x - x^{k-1}) f[x^k, x^{k-1}, \cdots, x^0] \end{aligned} \tag{2.11}$$

2.1.3 Lagrange polynomial interpolation

The Lagrange pol ynomial i nterpolation i s a reformulation of t he N ewton polynomial interpolation formulation. Observe that the first divided difference:

$$f[x^1, x^0] = \frac{f(x^1) - f(x^0)}{x^1 - x^0} \tag{2.12}$$

can be reformulated as:

$$f[x^1, x^0] = \frac{f(x^1)}{x^1 - x^0} + \frac{f(x^0)}{x^0 - x^1} \tag{2.13}$$

First order Newton interpolation polynomial can be given as follows:

$$f_1(x) = f(x^0) + \frac{f(x^1) - f(x^0)}{x^1 - x^0}(x - x^0) \tag{2.14}$$

Using (2.13) in (2.14) yields:

$$f_1(x) = f(x^0) + \frac{x - x^0}{x^1 - x^0} f(x^1) + \frac{x - x^0}{x^0 - x^1} f(x^0) \tag{2.15}$$

Observe that (2.15) can be rewritten as in the so called Lagrange form:

$$f_1(x) = \frac{x - x^1}{x^0 - x^1} f(x^0) + \frac{x - x^0}{x^1 - x^0} f(x^1) \tag{2.16}$$

Then, the second order Lagrange form is obtained as follows.

$$\begin{aligned} f_2(x) = {} & \frac{(x - x^1)(x - x^2)}{(x^0 - x^1)(x^0 - x^2)} f(x^0) + \frac{(x - x^0)(x - x^2)}{(x^1 - x^0)(x^1 - x^2)} f(x^1) \\ & + \frac{(x - x^0)(x - x^1)}{(x^2 - x^0)(x^2 - x^1)} f(x^2) \end{aligned}$$

The k <u>th</u> order version is finally given in (2.17)

$$f_k(x) = \sum_{s=0}^{k} L_s(x) f(x^s) \qquad \text{with} \qquad L_i(x) := \prod_{\substack{j=0 \\ j \neq i}}^{k} \frac{x - x^j}{x^i - x^j} \tag{2.17}$$

2.1.4 Spline interpolation for single variable functions

Calculation of polynomial coefficients is highly sensitive to round off and over flow errors in the above cited interpolation methods. So, the resulting polynomial interpolation may deviate too much away from the original function due to the errors especially in the coefficients related to the high order terms. To overcome this problem, one may prefer to employ a set of locally defined low order polynomials which are connected to each other in a smooth way. In other words, one may interpolate to each subset of the considered data set by a locally defined low order polynomial and concatenate these local interpolators in such a way providing a sufficiently smooth global interpolator. Corresponding a straight line to each pair of successive points yielding eventually a piecewise linear and continuous function is the simplest type of such smooth piecewise polynomial interpolators so called as spline functions. These interpolators differ from the other types of piecewise polynomial interpolators with the property that, for a $k+1$ th order spline interpolator, they are smooth in a certain degree, say k, at the data points where two splines meet.

2.1.4.1 Linear spline interpolation

The simplest form of spline interpolation is the linear spline interpolation employing first order local polynomials which is equivalent to piecewise linear interpolation and also to piecewise linear canonical representation (Chua & Kang 1978). In this interpolation, two successive points define a linear function. The resulting spline function is necessarily a continuous function since each point is shared by two successive local regions and also two successive local linear functions. Given a finite set of data pairs $\{(x^s, y^s)\}_{s=1}^{k+1}$ with $x^s, y^s \epsilon R$ and with the order $x^1 \leq x^2 \leq \cdots \leq x^k \leq x^{k+1}$, the first order splines $f_{(s)}(x)$ can be defined as:

$$f_{(s)}(x) = f(x^s) + \frac{f(x^{s+1}) - f(x^s)}{x^{s+1} - x^s}(x - x^s), \qquad s \in \{1,2,\cdots,k+1\} \tag{2.18}$$

The linear spline is continuous at each data point:

$$f_{(s)}(x^s) = f_{(s+1)}(x^s), \qquad s \in \{1,2,\cdots,k+1\} \tag{2.19}$$

To see (2.19), one can evaluate (2.18) at the s th and s+1 th sample points:

$$f_{(s)}(x) = f(x^s) + \frac{f(x^{s+1}) - f(x^s)}{x^{s+1} - x^s}(x - x^s) = y^s \tag{2.20}$$

$$f_{(s+1)}(x) = f(x^{s+1}) + \frac{f(x^{s+2}) - f(x^{s+1})}{x^{s+2} - x^{s+1}}(x - x^{s+1}) = y^s \tag{2.21}$$

Although linear spline interpolators are simple and continuous, they are not sufficient when differentiability is necessary. A point where two splines meet is called knot or junction point or break point. For the linear spline case, any point is also a knot. However, this is not true for higher order splines. In general, the slopes of the lines defining local linear functions change at the so called knot resulting discontinuity of the first order derivative of the function. This problem can be overcome by introducing higher order polynomial splines.

2.1.4.2 Quadratic spline interpolation

Quadratic splines associate a second order polynomial for each interval defined by four successive data points. Let the second order polynomial be represented as:

$$f_{(s)}(x) = a_s + b_s(x) + c_s(x)^2, \qquad s \in \{1,2,\cdots,k+1\} \tag{2.22}$$

For $(k+1)$ data points, there are k intervals and consequently $3k$ unknown constants (a's, b's and c's) to evaluate. Therefore, $3k$ equations or conditions are required to evaluate the unknowns. The required equations are derived within the following steps.

1. The function should be continuous at each knot except for the terminal ones, namely at the interior knots, so adjacent polynomials must have the same range values at a specific knot:

$$f_{(s)}(x) = a_{s-1} + b_{s-1}(x^s) + c_{s-1}(x^s)^2 \tag{2.23}$$

$$f_{(s)}(x) = a_s + b_s(x^s) + c_s(x^s)^2 \tag{2.24}$$

for $s = 2$ to k. Since the number of interior knots is $k-1$, then the continuity Equations in (2.23) and (2.24) provide, in total, $2k-2$ conditions.

2. The evaluation of first and last functions at the terminal points yields the following two additional equations:

$$f_{(s)}(x^1) = a_1 + b_1(x^1) + c_1(x^1)^2 \tag{2.25}$$

$$f_{(s)}(x^{k+1}) = a_{k+1} + b_{k+1}(x^{k+1}) + c_{k+1}(x^{k+1})^2 \tag{2.26}$$

With these two additional equations, the cumulative number of equations becomes $2k - 2 + 2 = 2k$.

3. In order to reach a smooth function, which has a continuous derivative, the derivatives of two adjacent polynomials should be equal at the interior knots:

$$f'_{(s)}(x^s) = b_{s-1} + 2c_{s-1}x^s \tag{2.27}$$
$$f'_{(s)}(x^s) = b_s + 2c_s x^s \tag{2.28}$$

for $s = 2$ to k. So, the smoothness of the first order derivatives of the quadratic spline interpolator provides $k - 1$ additional conditions. With these new additional equations, the cumulative number of equations becomes $2k + k - 1 = 3k - 1$.

4. In order to solve 3k unknown (a's, b's and c's) coefficients, one more equation is needed. The solvability can be achieved by assuming the second derivative as zero at the first data point (Equivalently by assuming first two points as connected by a straight line.):

$$2c_1 = 0 \tag{2.29}$$

One can solve these $3k$ linear equations for $3k$ unknown spline interpolator parameters by any numerical method developed for linear algebraic equations. For the case of three intervals, namely four data points, the system of equations can be given as in the following matrix form.

$$\begin{bmatrix} (x^3)^2 & 0 & 0 & (x^3) & 0 & 0 & 1 & 0 \\ 0 & (x^4)^2 & 0 & 0 & (x^4) & 0 & 0 & 1 \\ 0 & 0 & (x^2) & 0 & 0 & 1 & 0 & 0 \\ 0 & (x^3)^2 & 0 & 0 & (x^3) & 0 & 0 & 1 \\ -2(x^2) & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ (x^2)^2 & 0 & 0 & (x^2) & 0 & 0 & 1 & 0 \\ 0 & 0 & (x^1) & 0 & 0 & 1 & 0 & 0 \\ 2(x^3) & -2(x^3) & 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} (y^3) \\ (y^4) \\ (y^2) \\ (y^3) \\ 0 \\ (y^2) \\ (y^1) \\ 0 \end{bmatrix} \tag{2.30}$$

2.1.4.3 Cubic spline interpolation

Cubic splines are derived by concatenating a set of locally defined third order polynomials. The polynomial for a specific interval can be represented as:

$$f_{(s)}(x) = a_s + b_s(x) + c_s(x)^2 + d_s(x)^3, \qquad s \in \{1,2,\cdots,k+1\} \tag{2.31}$$

For $(k+1)$ data points, there are k intervals and, consequently, $4k$ unknown coefficients (a's, b's c's and d's) to be evaluated. Just as for quadratic splines, $4k$ equations are required to evaluate the unknown coefficients. These $4k$ equations are given as follows.

1. The function values of two adjacent local polynomials must be equal at the interior knots. (This condition yields $2k-2$ equations.)
2. The first and last local polynomials must pass through the end knots. (This condition yields 2 equations.)
3. The first order derivatives of two adjacent local polynomials must be equal at the interior knots. (This condition yields $k-1$ equations.)
4. The second derivatives at the interior knots must be equal. (This condition yields $k-1$ equations.)
5. The second derivatives are zero at the first and last knots. (This condition yields 2 equations and means that the first and last functions are straight lines.)

The $4k$ coefficients can be solved from the above given $4k$equations. An alternative way is presented below requiring the solution of only equations for $k-1$ (reduced) coefficients.

The first step in this derivation (Cheney & Kincaid, 1985) is based on the observation of the fact that the second derivative within each interval is a straight line since each pair of knots is connected by a cubic polynomial. The polynomial in (2.31) can be differentiated twice to verify this observation as:

$$f_s''(x) = 2c_s + 6d_s x \tag{2.32}$$

Now, the linear function in (2.32) can be represented by a first order Lagrange interpolating polynomial:

$$f_{(s)}''(x) = \frac{x - x^s}{x^{s-1} - x^s} f_{(s)}''(x^{s-1}) + \frac{x - x^{s-1}}{x^s - x^{s-1}} f_{(s)}''(x^s) \tag{2.33}$$

Where, $f_{(s)}''(x)$ is the value of the second derivative at any point x within the s th interval.

Then, an expression for the original function $f(x)$ can be obtained by integrating twice the linear second order derivative in (2.33).

$$f_{(s)}(x) = \frac{(x^s - x)^3}{6(x^s - x^{s-1})} f_{(s)}''(x^{s-1}) + \frac{(x - x^{s-1})^3}{6(x^s - x^{s-1})} f_{(s)}''(x^s) + c_1 x + c_2 \tag{2.34}$$

However, the expression in (2.34) contains two unknown integration constants (c_1 and c_2). These constants can be evaluated by using the fact that $f(x)$ must equal $f_{(s)}''(x^{s-1})$ at x^{s-1} and $f(x)$ must equal $f_{(s)}''(x^s)$ at x^s.

On the other hand, the function should give the range value y^s at the data point x^s:

$$f(x^0) = y^0, f(x^1) = y^1, \cdots, f(x^s) = y^s$$

So, one can get:

$$f_{(s)}(x^s) = y^s = \frac{(h_s)^3}{6h_s} f''_{(s)}(x^s) + c_1 x^s + c_2 \tag{2.35}$$

$$f_{(s)}(x^{s-1}) = y^{s-1} = \frac{(h_s)^3}{6h_s} f''_{(s)}(x^{s-1}) + c_1 x^{s-1} + c_2 \tag{2.36}$$

Where, $h_s = x^s - x^{s-1}$.

To find c_1, one can subtract (2.35) from (2.36):

$$y^{s-1} - y^s = \frac{(h_s)^3}{6h_s} \left[f''_{(s)}(x^{s-1}) - f''_{(s)}(x^s)\right] - c_1 h_s$$

$$c_1 = \frac{-(y^{s-1} - y^s)}{h_s} + \frac{h_s}{6} \left[f''_{(s)}(x^{s-1}) - f''_{(s)}(x^s)\right] \tag{2.37}$$

To find c_2, one can substitute c_1 into (2.35)

$$c_2 = y^s - \frac{(h_s)^2}{6} f''_{(s)}(x^s) - \left(\frac{-(y^{s-1} - y^s)}{h_s} + \frac{h_s}{6} \left[f''_{(s)}(x^{s-1}) - f''_{(s)}(x^s)\right]\right) x^s$$

c_2 can be rewritten as in the following form:

$$c_2 = \frac{y^{s-1} x^s}{h_s} - \frac{y^s x^{s-1}}{h_s} - \frac{h_s}{6} \left[f''_{(s)}(x^{s-1}) x^s - f''_{(s)}(x^s) x^{s-1}\right] \tag{2.38}$$

To find the cubic spline function, one can substitute the integral constants (c_1 and c_2) to (2.34):

$$f_{(s)}(x) = \frac{f''_{(s)}(x^{s-1})}{6(x^s - x^{s-1})}(x^s - x)^3 + \frac{f''_{(s)}(x^s)}{6(x^s - x^{s-1})}(x - x^{s-1})^3 - \frac{(y^{s-1} - y^s)x}{h_s}$$
$$+ \frac{h_s}{6}\left[f''_{(s)}(x^{s-1}) - f''_{(s)}(x^s)\right]x + \frac{y^{s-1}x^s}{h_s} - \frac{y^s x^{s-1}}{h_s}$$
$$- \frac{h_s}{6}\left[f''_{(s)}(x^{s-1})x^s - f''_{(s)}(x^s)x^{s-1}\right]$$

$$f_{(s)}(x) = \frac{f''_{(s)}(x^{s-1})}{6(x^s - x^{s-1})}(x^s - x)^3 + \frac{f''_{(s)}(x^s)}{6(x^s - x^{s-1})}(x - x^{s-1})^3$$
$$+ \left[\frac{f(x^{s-1})}{x^s - x^{s-1}} - \frac{f''_{(s)}(x^{s-1})(x^s - x^{s-1})}{6}\right](x^s - x) \tag{2.39}$$
$$+ \left[\frac{f(x^s)}{x^s - x^{s-1}} - \frac{f''_{(s)}(x^s)(x^s - x^{s-1})}{6}\right](x - x^{s-1})$$

The representation in (2.39) provides a cubic spline interpolation formula. However, the Equation (2.39) is a much more complicated expression for the cubic spline for the s th interval compare to (2.31). (2.39) contains only two unknown coefficients which are equal to the second derivatives $f''_{(s-1)}(x^s)$ at the beginning and $f''_{(s)}(x^s)$ at the end of the interval. Hence, if one can determine the proper second derivative at each knot, (2.39) provides a third order polynomial that can be used to interpolate within the interval.

The second derivatives can be evaluated by using the condition of that the first derivatives at the knots must be continuous:

$$f'_{(s-1)}(x^s) = f'_{(s)}(x^s) \tag{2.40}$$

To find the first derivative, one can differentiate the function in (2.39). Thus, the derivative function $f'_{(s)}(x)$ in the s th interval is given as:

$$f'_{(s)}(x) = -\frac{(x^s - x)^2}{2(x^s - x^{s-1})} f''_{(s)}(x^{s-1}) + \frac{(x - x^{s-1})^2}{2(x^s - x^{s-1})} f''_{(s)}(x^s) - \frac{(f(x^{s-1}) - f(x^s))}{x^s - x^{s-1}} + \frac{x^s - x^{s-1}}{6}\left[f''_{(s)}(x^{s-1}) - f''_{(s)}(x^s)\right] \quad (2.41)$$

On the other hand, $f'_{(s-1)}(x)$ in the interval $s-1$ is given as:

$$f'_{(s-1)}(x) = -\frac{(x^{s-1} - x)^2}{2(x^{s-1} - x^{s-2})} f''_{(s-1)}(x^{s-2}) + \frac{(x - x^{s-2})^2}{2(x^{s-1} - x^{s-2})} f''_{(s)}(x^{s-1}) - \frac{(f(x^{s-2}) - f(x^{s-1}))}{x^{s-1} - x^{s-2}} + \frac{x^{s-1} - x^{s-2}}{6}\left[f''_{(s)}(x^{s-2}) - f''_{(s)}(x^{s-1})\right] \quad (2.42)$$

The value of $f'_{(s-1)}(x)$ at the point x^{s-1} is

$$f'_{(s-1)}(x^{s-1}) = -\frac{(x^{s-1} - x^{s-1})^2}{2(x^{s-1} - x^{s-2})} f''_{(s-1)}(x^{s-2}) + \frac{(x^{s-1} - x^{s-2})^2}{2(x^{s-1} - x^{s-2})} f''_{(s)}(x^{s-1}) - \frac{(f(x^{s-2}) - f(x^{s-1}))}{x^{s-1} - x^{s-2}} + \frac{x^{s-1} - x^{s-2}}{6}\left[f''_{(s)}(x^{s-2}) - f''_{(s)}(x^{s-1})\right]$$

$$f'_{(s-1)}(x^{s-1}) = \frac{(x^{s-1} - x^{s-2})}{3} f''_{(s)}(x^{s-1}) - \frac{(f(x^{s-2}) - f(x^{s-1}))}{x^{s-1} - x^{s-2}} + \frac{(x^{s-1} - x^{s-2})}{6} f''_{(s)}(x^{s-2}) \quad (2.43)$$

The value of $f'_{(s)}(x)$ at the point x^{s-1} is:

$$f'_{(s)}(x^{s-1}) = -\frac{(x^s - x^{s-1})^2}{2(x^s - x^{s-1})} f''_{(s)}(x^{s-1}) + \frac{(x^{s-1} - x^{s-1})^2}{2(x^s - x^{s-1})} f''_{(s)}(x^{s-1}) - \frac{(f(x^{s-1}) - f(x^s))}{x^s - x^{s-1}} + \frac{x^s - x^{s-1}}{6}\left[f''_{(s)}(x^{s-1}) - f''_{(s)}(x^s)\right]$$

$$f'_{(s)}(x^{s-1}) = -\frac{(x^s - x^{s-1})}{3} f''_{(s)}(x^{s-1}) - \frac{(f(x^{s-1}) - f(x^s))}{x^s - x^{s-1}} - \frac{(x^s - x^{s-1})}{6} f''_{(s)}(x^s) \tag{2.44}$$

According to (2.40), both $f'_{(s-1)}(x^{s-1})$ and $f'_{(s)}(x^{s-1})$ take the same value at the point x^{s-1}. That is $f'_{(s-1)}(x^{s-1}) = f'_{(s)}(x^{s-1})$, so one can get:

$$\frac{(x^{s-1} - x^{s-2})}{3} f''_{(s)}(x^{s-1}) - \frac{(f(x^{s-2}) - f(x^{s-1}))}{x^{s-1} - x^{s-2}} + \frac{(x^{s-1} - x^{s-2})}{6} f''_{(s)}(x^{s-2}) = -\frac{(x^s - x^{s-1})}{3} f''_{(s)}(x^{s-1}) - \frac{(f(x^{s-1}) - f(x^s))}{x^s - x^{s-1}} - \frac{(x^s - x^{s-1})}{6} f''_{(s)}(x^s)$$

$$(x^{s-1} - x^{s-2}) f''_{(s)}(x^{s-1}) + 2 f''_{(s)}(x^{s-1})(x^s - x^{s-2}) + (x^s - x^{s-1}) f''_{(s)}(x^s) = \frac{6\big(f(x^{s-2}) - f(x^{s-1})\big)}{x^{s-1} - x^{s-2}} + \frac{6\big(f(x^s) - f(x^{s-1})\big)}{x^s - x^{s-1}} \tag{2.45}$$

If (2.45) is written for all interior knots, then $k-1$ equations involving $k+1$ unknown second derivatives are obtained. The problem reduces to $k-1$ equations with $k+1$ unknowns since the second derivatives at the end knots are zero for cubic splines. Thus, the above equations constitute a system of algebraic equation system which can be written in matrix-vector form AX=B where X represents the vector of coefficients $f''_{(s)}(x^s)$, B depends on y^s and A depends on x^s. Then, one can substitute the coefficients into (2.39), thus the cubic spline is found for the interval (x^{s-1}, x^s).

2.1.4.4 B-spline interpolation

Splines given in the last two previous sections are constructed by using piecewise polynomials satisfying certain degree of smoothness. Another type of splines, called B-splines (Schoenberg, 1967), is presented in this section. In contrast to splines described in the previous sections obtained by concatenating locally defined functions, B-splines is a linear weighted sum of linearly independent bases (spline) functions which are defined by some parameters (order of spline functions and the number of knots) and they span the piecewise polynomial smooth function space. Moreover, B-spline representation is a parametric representation not an explicit representation giving dependent variable $x, y \in R$ in terms of the independent variable $t \in R$. Instead, it provides a representation for the dependent x, y variable in terms of a parameter $t \in [0, 1]$.

B-splines with its distinguishing features have a number of advantages over the piecewise polynomial representations (de Boor, 1978; Schumaker, 1981). Spline bases locally support the function to be interpolated that is the function used in the interpolation can be locally tuned in order to fit to the function given by the sample data by adjusting defined basis functions. The sum of the weights of the basis splines is chosen as unity for each data point which is indeed scaled by the function value y^s, so yielding the desired function value at the considered data point. The most important feature of B-splines is in the calculation of their parameters by using a recurrence relation in a numerical way (Cox, 1972; de Boor, 1972).

The B-spline is a parametric representation $F(\cdot): [\, t_k,\ t_{L-k}] \subset R \to R^2$ defined by a linear combination of B-splines basis functions of degree k and is represented by:

$$\begin{bmatrix} x \\ y \end{bmatrix} = F(t) = \sum_{s=0}^{L} P_s b_s^k(t) \tag{2.46}$$

Where, $P_s = \begin{pmatrix} x^s \\ y^s \end{pmatrix}$ are called *control points or de Boor points* or coefficients (like weights in neural networks), which are composed of a set of finite data pairs $\{(x^s, y^s)\}_{s=0}^{L}$ with $x^s, y^s \in R$ and $b_s^k(t)$ is the normalized B-spline basis function of order k. The obtained B-spline interpolation function $F(t)$ degree is $k-1$. As seen from Equation (2.46), B-spline interpolation function $F(t)$ is linear with respect to the coefficients P_s but nonlinear in t due to nonlinearity of the basis function $b_s^k(t)$.

A knot sequence t_s may be defined as follows:

$$t_s = \begin{cases} 0, & if\ s < k \\ s - k + 1, & if\ k \leq s \leq L \\ L - k + 2, & if\ s > L \end{cases} \tag{2.47}$$

These knots satisfies the relation $t_s \leq t_{s+1}$ where $s \in \{0,1,2,\cdots,L\}$ and $t \in [t_k, t_{L-k}]$.

Illustrated example 1:

Let be given $L = 5$. For $k = 1$ and $k = 2$, the knot values are calculated by (2.47) as in the Table 2.1.

Table 2.1 The knot values for a given $k = 1$ and $k = 2$, and $L = 5$

$L = 5$	t_s	$k = 1$	$k = 2$
$s = 0$	t_0	0	0
$s = 1$	t_1	1	0
$s = 2$	t_2	2	1
$s = 3$	t_3	3	2
$s = 4$	t_4	4	3
$s = 5$	t_5	5	4
	t_6	6	5
	t_7	-	5

As shown in Table 2.1, the number of the knot sequence t_s depends on the order of the spline.

For the s'th normalized B-spline basis function of degree m, the basis function $b_s^k(t)$ is defined by Cox - de Boor recursion formulas:

$$b_s^1(t) = \begin{cases} 1, & if\ t_s \le t < t_{s+1} \\ 0, & otherwise \end{cases} \quad \text{with } s \in \{1,2,\cdots,L\} \tag{2.48}$$

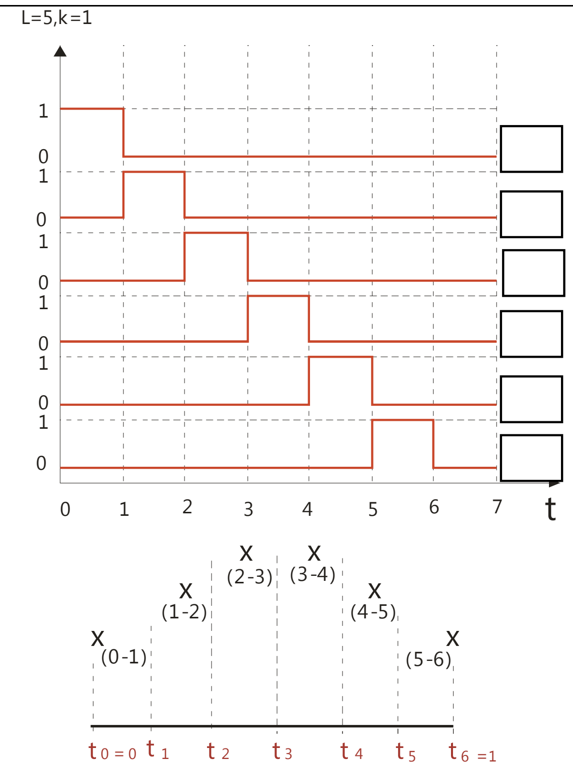
$$b_s^m(t) = \begin{cases} \dfrac{t-t_s}{t_{s+m}-t_s} b_s^{m-1}(t) + \dfrac{t_{s+m+1}-t}{t_{s+m+1}-t_{s+1}} b_{s+1}^{m-1}(t), & if\ t_s \le t < t_{s+k} \\ 0, & otherwise \end{cases} \tag{2.49}$$

$s \in \{0,1,2,\cdots,L\}$, $m \in \{1,2,\cdots,k\}$ with ${}^0/_0 = 0$.

Note that $s+m+1$ cannot exceed $L+k+1$ which the limits above recursion formula. Based on the knot sequence t_s, the B-spline is said to be either uniform (knots are equidistant) or nonuniform (knots are not equidistant) B-spline.

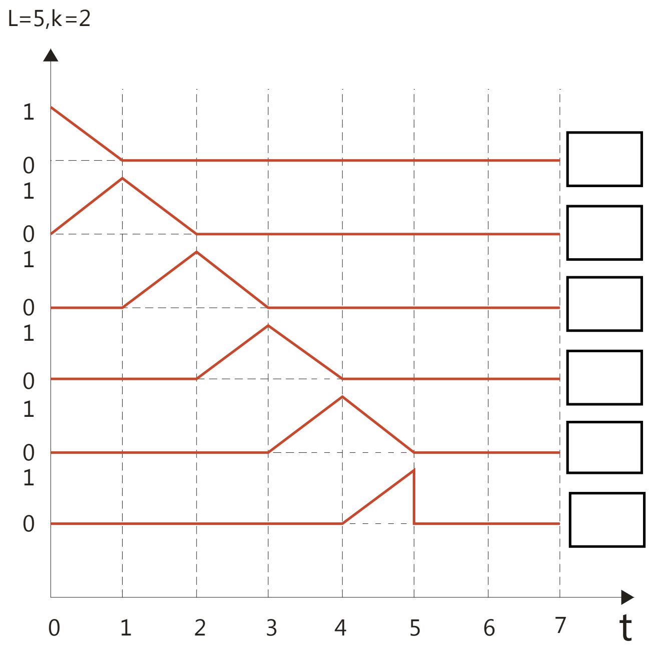
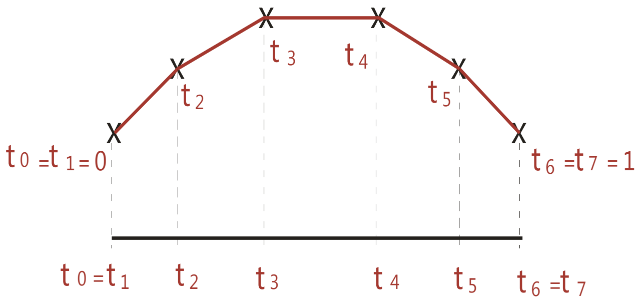
Illustrated example 2.1 Consider the case $k=1$. The domain of B-spline basis functions $b_s^k(t)$'s obtained by using the knot sequence in Table (2.1) is shown as in the following Table 2.2.

Table 2.2 The domain of B-spline basis functions $b_s^k(t)$'s for a given $k=1$, and $L=5$

$b_0^1(t) = \begin{cases} 1, & 0 \le t \le 1 \\ 0, & otherwise \end{cases}$ $b_1^1(t) = \begin{cases} 1, & 1 \le t \le 2 \\ 0, & otherwise \end{cases}$ $b_2^1(t) = \begin{cases} 1, & 2 \le t \le 3 \\ 0, & otherwise \end{cases}$ $b_3^1(t) = \begin{cases} 1, & 3 \le t \le 4 \\ 0, & otherwise \end{cases}$ $b_4^1(t) = \begin{cases} 1, & 4 \le t \le 5 \\ 0, & otherwise \end{cases}$ $b_5^1(t) = \begin{cases} 1, & 5 \le t \le 6 \\ 0, & otherwise \end{cases}$	 L=5,k=1 1 0 0 1 2 3 4 5 6 7 t x (0-1) x (1-2) x (2-3) x (3-4) x (4-5) x (5-6) t0=0 t1 t2 t3 t4 t5 t6=1

Illustrated example 2.2 Consider t he case $k = 2$. The dom ain of B-spline b asis functions $b_s^k(t)$'s is shown as in the following Table 2.3. Finding parametric function $F(t)$ is the continuous piecewise affine function in terms of B-splines.

Table 2.3 The domain of B-spline basis functions $b_s^k(t)$'s for a given $k = 2$, and $L = 5$

$b_0^2(t) = \frac{t-t_0}{t_1-t_0} b_0^1(t) + \frac{t_2-t}{t_2-t_1} b_1^1(t)$ $= \frac{t-0}{0} 0 + \frac{1-t}{1} b_2^0(t)$ $= \begin{cases} 1-t, & 0 \le t \le 1 \\ 0, & otherwise \end{cases}$ $b_1^2(t) = \frac{t-t_1}{t_2-t_1} b_1^1(t) + \frac{t_3-t}{t_3-t_2} b_2^1(t)$ $= \frac{t-0}{1} b_1^1(t) + \frac{2-t}{1} b_2^1(t)$ $= \begin{cases} t, & 0 \le t \le 1 \\ (2-t), & 1 \le t \le 2 \end{cases}$ $b_2^2(t) = \frac{t-t_2}{t_3-t_2} b_2^1(t) + \frac{t_4-t}{t_4-t_3} b_3^1(t)$ $= \frac{t-1}{1} b_2^1(t) + \frac{3-t}{1} b_3^1(t)$ $= \begin{cases} (t-1), & 1 \le t \le 2 \\ (3-t), & 2 \le t \le 3 \end{cases}$ $b_3^2(t) = \frac{t-t_3}{t_4-t_3} b_3^1(t) + \frac{t_5-t}{t_5-t_4} b_4^1(t)$ $= \frac{t-2}{1} b_3^1(t) + \frac{4-t}{1} b_4^1(t)$ $= \begin{cases} (t-2), & 2 \le t \le 3 \\ (4-t), & 3 \le t \le 4 \end{cases}$ $b_4^2(t) = \frac{t-t_4}{t_5-t_4} b_4^1(t) + \frac{t_6-t}{t_6-t_5} b_5^1(t)$ $= \frac{t-3}{1} b_4^1(t) + \frac{5-t}{1} b_5^1(t)$ $= \begin{cases} (t-3), & 3 \le t \le 4 \\ (5-t) & 4 \le t \le 5 \end{cases}$ $b_5^2(t) = \frac{t-t_5}{t_6-t_5} b_5^1(t) + \frac{t_7-t}{t_7-t_6} b_6^1(t)$ $= \frac{t-4}{1} b_5^1(t) + \frac{5-t}{0} 0$ $= \begin{cases} (t-4) b_5^1(t), & 4 \le t \le 5 \\ 0, & otherwise \end{cases}$	$b_0^1(t) = \begin{cases} 1, & t = 0 \\ 0, & otherwise \end{cases}$ $b_1^1(t) = \begin{cases} 1, & 0 \le t \le 1 \\ 0, & otherwise \end{cases}$ $b_2^1(t) = \begin{cases} 1, & 1 \le t \le 2 \\ 0, & otherwise \end{cases}$ $b_3^1(t) = \begin{cases} 1, & 2 \le t \le 3 \\ 0, & otherwise \end{cases}$ $b_4^1(t) = \begin{cases} 1, & 3 \le t \le 4 \\ 0, & otherwise \end{cases}$ $b_5^1(t) = \begin{cases} 1, & 4 \le t \le 5 \\ 0, & otherwise \end{cases}$   t0 =t 1=0 t 2 t 3 t 4 t 5 t 6 =t 7 =1 t 0 =t 1 t 2 t 3 t 4 t 5 t 6 =t 7

2.1.5 Canonical representations

A canonical representation of a function can be defined as a representation which is the minimal and the most compact form for the function requiring minimum number of parameters to define the function. This section presents two canonical representations: One for piecewise affine continuous functions and the other for continuous functions which are piecewise affine when holding fixed all of the variables except for a chosen one.

2.1.5.1 Canonical representation for piecewise affine functions

One Dimensional Case:

A Piece-Wise Affine (PWA) function $f(\cdot): R \rightarrow R$ with finite jump discontinuities is shown in Figure 2.1 where a PWA function with L breakpoints has $L+1$ intervals $I_0 \triangleq (-\infty, x_1],\ I_1 \triangleq (x_1, x_2], \ldots, I_{L-1} \triangleq \left(x_{L-1,} x_L\right]$, and $I_L \triangleq (x_{L,} \infty)$ in each of which the function is affine and m_j denotes the slope of segment j.

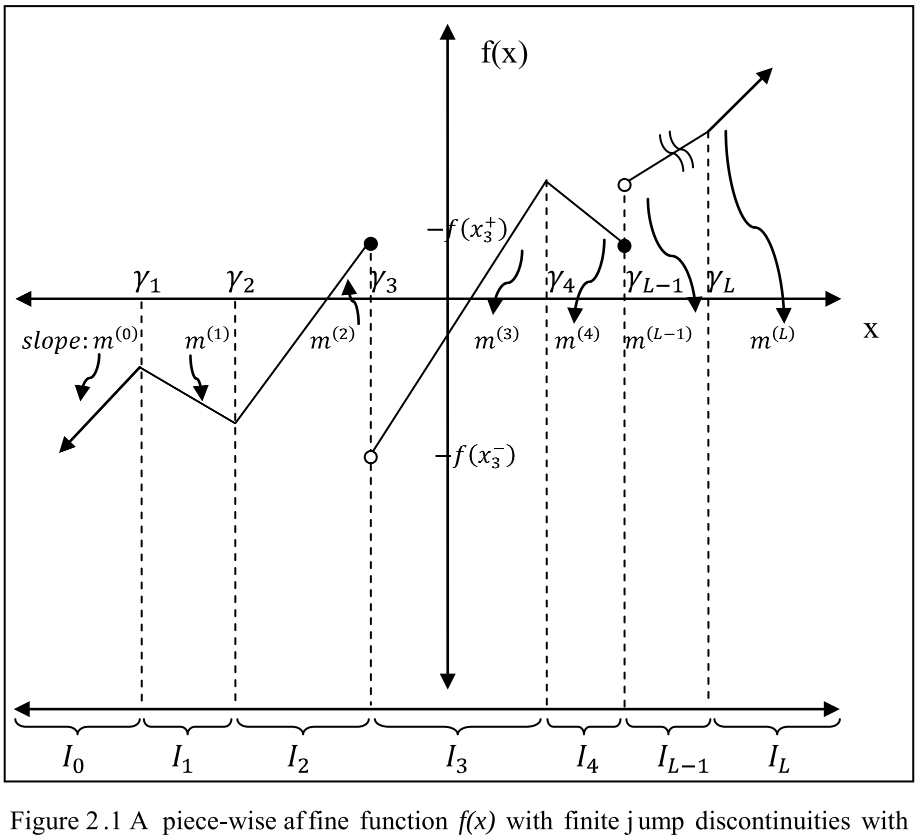


Figure 2.1 A piece-wise affine function *f(x)* with finite jump discontinuities with samples are breakpoints

As stated in the following theorem proved by (Chua & Kang, 1977), any PWA function described above has a compact representation in terms of absolute value and sign functions. This representation is called as canonical since it is shown in (Chua & Kang, 1977) that it needs minimum number of parameters necessary to describe the function in a complete way.

Theorem 2.2 [Chua & Kang 1977]: Any single variable single valued PWA function $f(\cdot): R \rightarrow R$ with at most L finite jump discontinuities at the L breakpoints $\gamma_1 < \gamma_2 < \cdots < \gamma_L$ can be represented uniquely by the expression:

$$f(x) = a_0 + a_1 x + \sum_{j=1}^{L} \{ b_j |x - \gamma_j| + c_j sign(x - \gamma_j) \} \tag{2.50}$$

Where, $|\cdot|$ denotes the absolute value function, $sign(\cdot)$ denotes the signum function and γ_j are equal to x^s breakpoints and the parameters $a_0,\ a_1,\ b_j,\ c_j \in R^1$ can be calculated as follows:

$$a_1 = (m^{(0)} + m^{(L)})/2$$

$$b_j = (m^{(j)} - m^{(j-1)})/2\,, \quad \text{with } j \in \{1, 2, \cdots, L\}$$

$$c_j = \begin{cases} 0, & if\ f(\cdot) is\ continuous\ at\ the\ breakpoint\ x = \gamma_j \\ \frac{1}{2}\left[f(x_j^+) - f(x_j^-)\right], & otherwise \end{cases} \tag{2.51}$$

$$a_0 = f(0) - \sum_{j=1}^{L} \left[b_j |\gamma_j| - c_j sgn(\gamma_j) \right]$$

□

The term $c_j sgn(x - \gamma_j)$ in (2.50) disappears when the function is continuous.

Theorem 2.2 provides a way of computing the coefficients of a canonical representation of a PWA function in terms of the breakpoints and the slopes of the segments where, a slope associated to an interval can be calculated as the ratio of the range deviation over the deviation in the breakpoints belonging to the interval. The canonical representation given above is a global representation not restricted to a sub-region of the domain space but valid for the whole domain. The absolute value and sign functions together with the addition and scalar multiplication are the sole algebraic operations used in defining the PWA canonical functions. These features make the PWA canonical functions efficient in many aspects. The analyses of the systems defined by PWA canonical models can be realized by the algorithms easy to be programmable and they require minimal amount of storage.

The observation of the above compact global absolute value based representation of one dimensional PWA functions led to the development of canonical representation for multi-variable PWA functions by Chua & Kang [1978].

n - dimensional case:

Chua and Kang extended one dimensional canonical representation into higher dimensions by introducing the following canonical representation for n-dimensional m-valued PWA continuous functions that are affine over convex polyhedral regions constructed by linear partitions. Herein, a locally defined affine function can be given by a Jacobian matrix $\boldsymbol{J}_{\boldsymbol{R}_I} \in R^{mxn}$ and offset vector $\boldsymbol{w}_{R_I} \in R^m$ as $f(\boldsymbol{x}) = \boldsymbol{J}_{R_I}\mathbf{x}$ $+\boldsymbol{w}_{R_I}$ for $x \in R_I \subset R^n$ where $R_I := \{\mathbf{x} \in \boldsymbol{R}^{\boldsymbol{n}} | \boldsymbol{\alpha}_j^T\mathbf{x} - \gamma_j \geq 0 \; \forall \, i \; \in I \; and \; \boldsymbol{\alpha}_j^T\mathbf{x} - \gamma_j \leq 0 \; \forall \, i \; \in I\}$ for a specific index set $I \subset \{1,2,\ldots,L\}$.

Theorem 2.3 (Necessary and sufficient condition) [Chua & Kang, 1988; Güzeliş & Göknar, 1991]: A continuous PWA function $f(\cdot): R^n \to R^m$ defined over a linear partition determined by a set of hyper-planes $\boldsymbol{\alpha}_j^T\mathbf{x} - \gamma_j = 0$ with $j \in \{1,2,\ldots,L\}$ has a 1-level canonical representation.

$$f(\mathbf{x}) = a + A\mathbf{x} + \sum_{j=1}^{L} b_j \left| \, \boldsymbol{\alpha}_j^T\mathbf{x} - \gamma_j \right| \tag{2.52}$$

with a , $b_j \in R^m$, $\boldsymbol{\alpha}_j \in R^n$, $A \in R^{mxn}$ and $\gamma_j \in R^1$ if and only if it satisfies the consistent variation property. The consistent variation property means that there exists a unique $\boldsymbol{q}_j \in R^m$ for each hyper-plane such that the variations in the jacobian matrices for each pair of n-dimensional regions R_{ji}^+ and R_{ji}^- separated by the hyper-plane $\boldsymbol{\alpha}_j^T\mathbf{x} - \gamma_j = 0$ are the same:

$$\boldsymbol{J}_{R_{j1}^+} - \boldsymbol{J}_{R_{j1}^-} = \boldsymbol{J}_{R_{j2}^+} - \boldsymbol{J}_{R_{j2}^-} = \cdots \boldsymbol{J}_{R_{jn_j}^+} - \boldsymbol{J}_{R_{jn_j}^-} = \boldsymbol{q}_j\boldsymbol{\alpha}_j^T,\ j \in \{1,2,\dots,L\} \qquad (2.53)$$

Where, $\boldsymbol{J}_{R_{j1}^+}$ and $\boldsymbol{J}_{R_{j1}^-}$ denote the jacobian matrices of the regions R_{ji}^+ and R_{ji}^-. $\boldsymbol{\alpha}_j^T\mathbf{x} - \gamma_j \geq 0$ for $\mathbf{x} \in R_{ji}^+$ and $\boldsymbol{\alpha}_j^T\mathbf{x} - \gamma_j \leq 0$ for $\mathbf{x} \in R_{ji}^-$. The intersection between R_{ji}^+ and R_{ji}^- must be a subset of an $(n-1)$-dimensional hyperplane and can not be covered by any hyperplane of lower dimension.

□

As the one in (2.50), the canonical representation in (2.52) is global in the sense that it is not valid for a specific domain region but for the whole domain covering all the convex polyhedral regions separated by the hyperplanes $\boldsymbol{\alpha}_j^T\mathbf{x} - \gamma_j$ with $j \in \{1,2,\dots,L\}$.

Although the canonical representation (2.50) can represent the whole set of one dimensional PWA functions $f(\cdot): R^1 \to R^m$ including discontinuous ones, the representation (2.52) only covers a subset of n-dimensional continuous PWA functions $f(\cdot): R^n \to R^m$. The consistent variation property is indeed satisfied for any kind of continuous PWA functions whose domain is a non-degenerate linear partition. But, this is not always true for degenerate partitions.

Definition 2.1 (Non-degenerate Partition) [Chua & Kang, 1988]: A linear partition determined by the hyper-planes

$$\boldsymbol{\alpha}_j^T\mathbf{x} - \gamma_j = 0,\ \ j \in \{1,2,\dots,L\}$$

is said to be nondegenerate iff for every set of linearly dependent vectors $\{\boldsymbol{\alpha}_{i2}, \boldsymbol{\alpha}_{i2}, \cdots \boldsymbol{\alpha}_{ik}\}$ with $k \in \{1,2,\dots,L\}$ the rank of $[\boldsymbol{\alpha}_{i2}, \boldsymbol{\alpha}_{i2}, \cdots \boldsymbol{\alpha}_{ik}]$ is strictly less than the rank of the following $(n+1)$ x m augmented matrix by γ_{ik}'s.

$$\begin{bmatrix} \boldsymbol{\alpha}_{i1} & \boldsymbol{\alpha}_{i2} & \cdots & \boldsymbol{\alpha}_{ik} \\ \gamma_{i1} & \gamma_{i2} & \cdots & \gamma_{ik} \end{bmatrix}.$$

A linear partition containing three lines intersecting at a common point is an example of degenerate partition in a 2-dimensional space. This fact can be seen by the observation that 2-dimensional normal vectors associated to the three lines is necessarily linearly dependent and the offsets of the three lines are consistent, so they do not increase the rank of the augmented matrix in Definition 2.1. A linear partition containing three planes having a common intersection of dimension 1 is another example for degenerate partitions. A non-degenerate linear partition is a linear partition where the dimension of the intersection of any i hyperplanes each of $n-1$ dimensional must be strictly less than $n-i+1$. The importance of nondegererenate linear partitions relies on the fact that the consistent variation property is always satisfied for PWA continuous functions defined over a nondegenerate linear partition, so ensuring the existence of the canonical representation (2.52).

Theorem 2.4 (Sufficient condition) [Chua & Kang, 1988]: A continuous PWA function $f(\cdot): R^n \to R^m$ defined over a linear partition determined by a set of hyper-planes $\boldsymbol{\alpha}_j^T \mathbf{x} - \gamma_j = 0$ with $j \in \{1,2,\dots,L\}$ has a 1-level canonical representation of the form (2.52) if the linearly partitioned domain space is nondegenerate.

□

The 1-level canonical representation (2.52) has been extended by several studies in the literature [Kahlert & Chua 1990; Güzeliş & Göknar, 1991; Unbehauen, 1994; Julian, Desages & Agamennoni, 1999] into higher-level canonical representations. These higher level representations employ bases functions defined by different levels of nested absolute value functions for handling inconsistent variations of each pair of the Jacobian matrices and offset vectors that define local affine functions in the neighbouring regions seperated by the same hyperplane.

A 2 -level c anonical r epresentation given i n (2.54) proposed by Güzeliş and Göknar (1991) extends the representation (2.55) into the piecewise affine partitioned domain spaces which indeed constitute a special class of degenerate linear partitions.

$$f(\mathbf{x}) = a + A\mathbf{x} + \sum_{j=1}^{L} b_j \left| \boldsymbol{\alpha}_j^T \mathbf{x} - \gamma_j \right| + \sum_{i=1}^{P} c_i \left| \boldsymbol{\beta}_i^T \mathbf{x} + \delta_i + \sum_{j=1}^{L} d_{ij} \left| \boldsymbol{\alpha}_j^T \mathbf{x} - \gamma_j \right| \right| \tag{2.54}$$

The 1-level representation (2.57) which uses the conventional hyperplanes

$$H_j = \{\mathbf{x} \in R^n | \ \boldsymbol{\alpha}_j^T \mathbf{x} - \gamma_j = 0\}, j \in \{1,2,\dots,L\}, \boldsymbol{\alpha}_j \in R^n, \gamma_j \in R, \tag{2.55}$$

and also piecewise affine hyper-planes

$$S_i = \{\mathbf{x} \in R^n | \Psi_i(\mathbf{x}) = 0\} \ i \in \{1,2,\dots,P\} \tag{2.56}$$

Where

$$\Psi_i(\mathbf{x}) \coloneqq \boldsymbol{\beta}_i^T \mathbf{x} + \delta_i + \textstyle\sum_{j=1}^{L} d_{ij} \left| \boldsymbol{\alpha}_j^T \mathbf{x} - \gamma_j \right|. \tag{2.57}$$

For t he canonical r epresentation (2.54), t he c onsistent va riation pr operty [Chua and Kang, 1988] or, in other words, the consistency of continuity vectors [Güzeliş and G öknar, 1991] i s g iven b y t he e quations (2.58) a nd (2.59). F or a ny pa ir of regions R^i and R^j separated b y a co nventional hyperplane H_k, the c onsistency of continuity vectors is the uniqueness of the continuity vectors $\boldsymbol{q}_k^{i,j}$'s for all i, j and k:

$$[J^i - J^j, \quad \mathrm{w}^i - \mathrm{w}^j] = \boldsymbol{q}_k^{i,j} [\boldsymbol{\alpha}_k^T, -\gamma_k] \tag{2.58}$$

For t he p air of regions of R^i and R^j separated b y a P WA h yperplane S_k, t he consistency of continuity vectors becomes the uniqueness of the following continuity vectors $\boldsymbol{q}_k^{i,j}$'s for all i, j and k:

$$[J^i - J^j,\quad \mathrm{w^i} - \mathrm{w^j}] = \boldsymbol{q}_k^{i,j}\left([\boldsymbol{\beta}_k^T, \delta_k] + \sum_{p\in P_1} d_{kp}[\boldsymbol{\alpha}_p^T, -\gamma_p] - \sum_{p\in P_2} d_{kp}[\boldsymbol{\alpha}_p^T, -\gamma_p]\right) \quad (2.59)$$

The existence of the continuity vectors $\boldsymbol{q}_k^{i,j}$ in the above equations are indeed the necessary and sufficient conditions for the continuity of the PWA function defined over the PWA partitioned domain space.

Theorem 2.5 (Necessary and sufficient condition) [Güzeliş & Göknar, 1991]: A PWA continuous function $f(\cdot): R^n \to R^m$ defined over a PWA partition determined by the hyper-planes and PWA hyper-planes given in (2.55)-(2.58) can be represented by the canonical form (2.54) if and only if its continuity vectors are consistent in the sense of expressions given in (2.58)-(2.59).

□

It is shown in [Güzeliş and Göknar, 1991] that the non-degeneracy of the PWA partition can be defined as in the linear partition case, so a necessary condition for the existence of the canonical representation (2.54) can be obtained as the non-degeneracy of the PWA partition. Although the 1-level representation (2.54) is quite general, it cannot cover the whole set of continuous PWA functions. In the literature [Chua & Kang, 1977; Kang & Chua, 1978, Chua & Kang, 1988; Kahlert & Chua, 1990; Güzeliş & Göknar, 1991; Kahlert & Chua, 1992; Kevenaar, Leenarts & Bokhoven, 1994; Lin , Xu & Unbehauen, 1994, Lin & Unbehauen, 1995; Leenarts, 1999; Julian, Desages & Agamennoni, 1999], there are many attempts to represent the whole class of continuous PWA functions using the absolute value function as the unique nonlinear building block. Among these attempts, the work presented in [Lin , Xu & Unbehauen, 1994] may be the most remarkable one as proving that any kind of continuous PWA function defined over a linear partition in R^n can be expressed by an , at most n −level canonical, representation employing n -nested absolute value functions.

2.1.5.2 Canonical representation for section-wise piecewise affine functions

Canonical representations can also be used for multi variable functions which are not PWA according to all variables but PWA for a single variable when all other variables are held fixed. Chua and Kang introduced such a representation in [Chua and Kang, 1977] and called it as section-wise PWA canonical representation:

$$f(x_1, x_2, x_3, \cdots, x_n) = \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} \cdots \sum_{k_n=1}^{N} \alpha(k_1, k_2, k_3, \cdots, k_n) \cdot \prod_{j=1}^{n} \varphi_{kj}(x_j) \qquad (2.60)$$

Where, $\alpha(k_1, k_2, k_3, \cdots, k_n)$ are the constant coefficients and N is a set of given data points (i.e. for the case $n = 3$, there are N^3data points such as $(x_{1i}, x_{2j}, x_{3k}), i, j, k \in \{1,2, \dots, N\}$.) and the basis functions are:

$$\begin{aligned} &\varphi_1(x_j) = 1 \\ &\varphi_2(x_j) = x_j \\ &\varphi_3(x_j) = |x_j - x_{j1}| \\ &\vdots \\ &\varphi_N(x_j) = |x_j - x_{jN-2}|. \end{aligned} \qquad (2.61)$$

Various interpolation methods are given in Section 2.1. They have wide applicability for the cases where the data is known to be precise. However, in practice, the data is imprecise in most of the cases. The noise and outliers are two possible sources of impreciseness. Data reduction, filtering and employing a low order interpolator not passing through all of the data are among the solutions for handling imprecise data. The approximate representations which will be studied in the following section serve solutions for imprecise data and also for large data cases.

2.2 Approximate representations (regression)

The interpolation, which is studied in Section 2.1, is not only the process of defining a function passing directly through a given set of data pairs but also the process of predicting acceptable range values for the input data not available in the design phase of the interpolating function. However, the interpolation may not be suitable especially when there exist noise and outliers. On the other hand, large set of data which are used in designing an interpolator requires considerable memory allocation and time consumption. Another disadvantage of the interpolation is its very poor generalization ability even for moderate size data sets.

In order to solve the mentioned problems dealing with interpolation, one can try to suppress the undesired data in a way. In this direction, one can eliminate the data at the beginning and then apply the interpolation to the reduced set of data. In a more general setting, one can employ an approximate function which does not aim to fit all of the given data but to fit them with the minimum approximation error. Such an approach, called as function approximation, yields more simple function representations which have numerical efficiencies and they also provide good generalization abilities. This section presents a diverse set of function approximation methods known in the literature in a comparative setting.

Function approximation can be defined as a problem of finding a function $f(\cdot): X \to Y$ which fits to a given set $\{(\mathbf{x}^s, \mathbf{y}^s)\}_{s=1}^{L}$ of domain-range sample pairs where $\mathbf{x}^s \in X$ denotes the samples of the independent variable and $\mathbf{y}^s \in Y$ denotes the samples of the dependent variable. The domain-range sample pairs $(\mathbf{x}^s, \mathbf{y}^s)$'s are, indeed, samples of an unknown function corresponding to, for instance, a signal or a relation among some of the variables of a system. The samples are usually obtained by measurements in experiments and observations. This thesis assumes real domain and range sets, i.e. $X = R^n$ and $Y = R^m$.

The first step in the function approximation is to choose a model $f_w(\cdot): R^n \to R^m$, more precisely building blocks so called basis functions and the type of combination of building blocks. Model selection is realized based on a priori information about the structure of the function to be approximated. The past experience of the user, the chosen implementation environment, the numerical efficiency and the structural capabilities such as flexibility, universality and generalization ability are among the factors taken into account in the model selection process. The second step is to determine the parameters $w \in R^p$ of the chosen function model. The determination of the model function parameters is usually done based on minimization of an approximation error measure: $\sum_{s=1}^{L} d(y^s, f_w(x^s))$ where $d(\cdot,\cdot) : R^m \text{x}\, R^m \rightarrow R^+$ denotes a function satisfying metric conditions (Rudin, 1976).

The function approximation problem is described above for finite number of data case in a deterministic setting. A similar concept, which is from the statistical domain, is the regression. The regression seeking for a relation between the range samples $\mathbf{y}^s$'s and domain samples $\mathbf{x}^s$'s is expressed as the estimation of an unknown conditional probability density function $f(\mathbf{x}^s)$ from the set $\{(\mathbf{x}^s, y^s)\}_{s=1}^{L}$ of samples based on the following model.

$$y^s = f(\mathbf{x}^s) + e^s \tag{2.62}$$

Where, e^s is a random error term, whose mean is zero ($\mu = 0$) and its variance is a constant (σ^2), and e^s is usually assumed to have normal (i.e. Gaussian) distribution (Montgomery & Peck, 1992). When $f(\mathbf{x})$ is parameterized by choosing a parametric model as $f_w(\mathbf{x})$, the estimation of unknown conditional probability density function $f(\mathbf{x}^s)$ amounts to the estimation of an unknown parameter vector $\boldsymbol{w} \in R^p$

Two important approximate representations are presented in this section. The first one is based on orthogonal basis functions and the other is based on non-orthogonal basis functions. In both cases, the bases functions are constructed from the data pairs.

2.2.1 Approximate function representations

In this subsection, a general framework for least square function approximation valid for orthogonal and also nonorthogonal basis function sets is firstly given. Then, approximate representations employing orthogonal basis, namely, polynomial, Fourier and wavelet basis functions are presented as special cases. Euclidean norm is chosen for the error metric in the approximations, so the approximation to be presented is, indeed, the least square approximation which corresponds to a parametric regression. For the sake of simplicity, the functions to be approximated and then the approximating model functions are assumed to be multi-variable single-valued, i.e. $f(\cdot): R^n \rightarrow R^1$.

2.2.1.1 Least Square Approximation: A General Framework

Assume that the model used for approximation is a linear weighted sum of a finite set $\{\emptyset_k(\mathbf{x})\}_{k=1}^{m}$ of bases.

$$y = f(\mathbf{x}) = \sum_{k=1}^{m} w_k \, \emptyset_k(\mathbf{x}) \tag{2.63}$$

Where, each basis function is multi-variable and single-valued: $\emptyset_k(\cdot): R^n \rightarrow R^1$. In order to have an affine representation in the feature space defined by $\emptyset_k(\cdot)$'s, one may choose $\emptyset_1(\mathbf{x}) = 1$, so w_1 becomes a bias term.

Now, the function approximation can be posed as a problem of determining the coefficients $w_k \in R^1$ with $k \in \{1,2,\dots,m\}$ minimizing the following total squared error for a given sample set $\{(\mathbf{x}^s, y^s)\}_{s=1}^{L}$ where $\mathbf{x}^s \in R^n, y^s \in R$ for $s \in \{1,2,\dots,L\}$.

$$\sum_{s=1}^{L} \left[y^s - \sum_{k=1}^{m} w_k \, \emptyset_k(\mathbf{x}^s) \right]^2 \tag{2.64}$$

The total squared error in (2.64) has minimum points only since it is a positive semi-definite quadratic function of the unknown w_k coefficients. When the quadratic error function is positive definite, there is a unique set $\{w_k\}_{k=1}^{m}$ of coefficients defining the unique m inimum point. For t he uni que m inimum c ase, in or der t o de termine w_k coefficients, one needs to consider only the first order necessary conditions which are obtained by taking the derivatives of the total squared error in (2.64) with respect to the w_k coefficients a nd t hen s etting t hem t o z ero. A s a r esult, t he first or der necessary c onditions, which a re also s ufficient f or t he pos itive s emi-definite quadratic t otal s quared e rror, yield t he f ollowing s et of e quations c alled as nor mal equations.

$$\frac{d}{dw_k}\left\{\sum_{s=1}^{L}\left[y^s - \sum_{k=1}^{m} w_k\,\emptyset_k(\mathbf{x}^s)\right]^2\right\} = 0,\;\; k \in \{1,2,\cdots,m\}$$

which yields:

$$-2\left\{\sum_{s=1}^{L}\left[y^s - \sum_{k=1}^{m} w_k\,\emptyset_k(\mathbf{x}^s)\right]\right\}\emptyset_1(\mathbf{x}^s) = 0$$

$$-2\left\{\sum_{s=1}^{L}\left[y^s - \sum_{k=1}^{m} w_k\,\emptyset_k(\mathbf{x}^s)\right]\right\}\emptyset_2(\mathbf{x}^s) = 0$$

$$\vdots$$

$$-2\left\{\sum_{s=1}^{L}\left[y^s - \sum_{k=1}^{m} w_k\,\emptyset_k(\mathbf{x}^s)\right]\right\}\emptyset_m(\mathbf{x}^s) = 0$$

The normal equations obtained above can be recast into the following form defined by the Gram matrix.

$$\begin{bmatrix} \sum_{s=1}^{L} \emptyset_1(\mathbf{x}^s).\emptyset_1(\mathbf{x}^s) & \sum_{s=1}^{L} \emptyset_2(\mathbf{x}^s).\emptyset_1(\mathbf{x}^s) & \cdots & \sum_{s=1}^{L} \emptyset_m(\mathbf{x}^s).\emptyset_1(\mathbf{x}^s) \\ \sum_{s=1}^{L} \emptyset_1(\mathbf{x}^s).\emptyset_2(\mathbf{x}^s) & \sum_{s=1}^{L} \emptyset_2(\mathbf{x}^s).\emptyset_2(\mathbf{x}^s) & \cdots & \sum_{s=1}^{L} \emptyset_m(\mathbf{x}^s).\emptyset_2(\mathbf{x}^s) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{s=1}^{L} \emptyset_1(\mathbf{x}^s).\emptyset_m(\mathbf{x}^s) & \sum_{s=1}^{L} \emptyset_2(\mathbf{x}^s).\emptyset_m(\mathbf{x}^s) & \cdots & \sum_{s=1}^{L} \emptyset_m(\mathbf{x}^s).\emptyset_m(\mathbf{x}^s) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} \sum_{s=1}^{L} y^s \emptyset_1(\mathbf{x}^s) \\ \sum_{s=1}^{L} y^s \emptyset_2(\mathbf{x}^s) \\ \vdots \\ \sum_{s=1}^{L} y^s \emptyset_m(\mathbf{x}^s) \end{bmatrix} \quad (2.65)$$

Generalized inverse:

A vector $\boldsymbol{w}$ which solves the system (2.65) may not exist, or if one exists, it may not be uni que. In s uch c ases, t he s o-called generalized i nverse s olutions c ould be employed to find a solution to the normal equations in the least square sense which actually c orrespond t o t he m inima of t he t otal s quared error i n (2.64). Where, the generalized inverse $\boldsymbol{\Phi}^{+} \in R^{l \times m}$ of a matrix $\boldsymbol{\Phi} \in R^{m \times l}$ is defined as follows.

$$\boldsymbol{\Phi}^{+} = \lim_{\varepsilon \to 0} [\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \varepsilon I]^{-1} \boldsymbol{\Phi}^T$$

The system (2.65) can be given in the following form:

$$\boldsymbol{\Phi w} = \mathbf{y} \quad (2.66)$$

To find $\boldsymbol{w} \in R^n$ minimizing (2.66), one can use the following identities.

$$\begin{aligned} \| \boldsymbol{\Phi w} - \mathbf{y} \|_2^2 &= (\boldsymbol{\Phi w} - \mathbf{y})^T (\boldsymbol{\Phi w} - \mathbf{y}) \\ &= (\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} - \mathbf{y}^{\mathrm{T}})(\boldsymbol{\Phi w} - \mathbf{y}) \\ &= \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi w} - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{y} - \mathbf{y}^{\mathrm{T}} \boldsymbol{\Phi w} + \mathbf{y}^{\mathrm{T}} \mathbf{y} \end{aligned} \quad (2.67)$$

The last expression can be rewritten as in (2.65) using the following properties of the generalized inverse.

Property 1: $\mathbf{\Phi}^{+}\mathbf{\Phi}\mathbf{\Phi}^{+} = \mathbf{\Phi}^{+}$

Property 2: $\mathbf{\Phi}\mathbf{\Phi}^{+}\mathbf{\Phi} = \mathbf{\Phi}$

Property 3:$(\mathbf{\Phi}\mathbf{\Phi}^{+})^{T} = \mathbf{\Phi}\mathbf{\Phi}^{+}$

Property 4: $(\mathbf{\Phi}^{+}\mathbf{\Phi})^{T} = \mathbf{\Phi}^{+}\mathbf{\Phi}$

$$\| \mathbf{\Phi}\boldsymbol{w} - \mathbf{y} \|_2^2 = (\boldsymbol{w} - \mathbf{\Phi}^{+}\mathbf{y})\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}(\boldsymbol{w} - \mathbf{\Phi}^{+}\mathbf{y}) + \mathbf{y}^{\mathrm{T}}(\mathbf{I} - \mathbf{\Phi}\mathbf{\Phi}^{+})\mathbf{y} \qquad (2.68)$$

It can be s een t hat t he ge neralized i nverse s olution $\boldsymbol{w} = \mathbf{\Phi}^{+}\mathbf{y}$ minimizes the Euclidean norm of the error vector**,** so (2.68) as observing that the first term becomes zero and the last term is not dependent on $\boldsymbol{w}$. It should be noted that there are many methods for calculating the generalized inverse of a m atrix. One of the numerically efficient ones is based on singular value decomposition (Golub & Van Loan, 1996).

2.2.1.2 Least Square Polynomial Approximation

For a given s ample s et $\{(x^s, y^s)\}_{s=1}^{L}$ where $x^s, y^s \in R$ for $s \in \{1,2, \dots, L\}$, consider polynomials $\emptyset_i(x) = x^i \; with \; i \in \{1,2, \dots, k\}$ as basis functions. The least square approximation based on these polynomials is the function:

$$f(x) = w_1 + w_2 x + \cdots + w_k x^k$$

Where, the coefficients $w_0, w_1, \cdots, w_m$ are the solutions of the following equation system.

$$\begin{bmatrix} \sum_{s=1}^{L}(1) & \sum_{s=1}^{L}(x^s) & \cdots & \sum_{s=1}^{L}(x^s)^k \\ \sum_{s=1}^{L}(x^s) & \sum_{s=1}^{L}(x^s)^2 & \cdots & \sum_{s=1}^{L}(x^s)^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{s=1}^{L}(x^s)^k & \sum_{s=1}^{L}(x^s)^{k+1} & \cdots & \sum_{s=1}^{L}(x^s)^{2k} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix} = \begin{bmatrix} \sum_{s=1}^{L} y^s \\ \sum_{s=1}^{L} y^s\,(x^s) \\ \vdots \\ \sum_{s=1}^{L} y^s\,(x^s)^k \end{bmatrix} \qquad (2.69)$$

Solving the above set of equations requires finding the inverse of the input samples matrix. To find the coefficients easier, one can use orthonormal polynomials yielding diagonal input sample matrices. To construct such an orthonormal set, one can use Gram-Schmidt orthogonalization.

Gram-Schmidt Orthogonalization Procedure : Given a basis $\{\emptyset_k(\mathbf{x})\}_{k=1}^{m}$, one can construct an orthononormal basis $\{\psi_k(\mathbf{x})\}_{k=1}^{m}$ for the space spanned by $\{\emptyset_k(\mathbf{x})\}_{k=1}^{m}$ via the following process called as Gram-Schmidt orthogonalization (Table 2.4).

Table 2.4 Gram-Schmidt orthogonalization procedure

Assume $\Omega_1 = \emptyset_1(x)$
then $\psi_1 = \frac{\Omega_1}{\Vert\Omega_1\Vert}$
$\Omega_2 = \emptyset_2(x) - \langle\emptyset_2, \psi_1\rangle\psi_1$
$\psi_2 = \dfrac{\Omega_2}{\Vert \Omega_2 \Vert}$
$\Omega_3 = \emptyset_3(x) - \langle\emptyset_3, \psi_1\rangle\psi_1 - \langle\emptyset_3, \psi_2\rangle\psi_2$
$\psi_3 = \dfrac{\Omega_3}{\Vert \Omega_3 \Vert}$
$\vdots$
$\Omega_k = \emptyset_k(x) - \sum_{k=1}^{k-1}\langle\emptyset_k, \psi_k\rangle\psi_k$
$\psi_k = \dfrac{\Omega_k}{\Vert \Omega_k \Vert}$

Illustrated Example 2.2

Consider t he s et of pol ynomial ba sis function set a s $\{1, x\}$. The G ram S chimdt orthogonalization process for this basis set $\{1, x\}$ is given as:

$$\Omega_1 = \emptyset_1(x) = 1$$

$$\psi_1 = \frac{\Omega_1}{\Vert \Omega_1 \Vert} = \frac{1}{\sqrt{\sum_{s=1}^{L} 1.1}} = \frac{1}{\sqrt{L}}$$

$$\Omega_2 = \emptyset_2(x) - \psi_1\langle\emptyset_2, w_1\rangle$$

$$= x - \frac{1}{\sqrt{L}}\left(\sum_{s=1}^{L}\frac{1}{\sqrt{L}}x^s\right) = x - \frac{1}{L}\left(\sum_{s=1}^{L}x^s\right)$$

$$\psi_2 = \frac{x - \frac{1}{L}(\sum_{s=1}^{L}x^s)}{\sqrt{\sum_{s=1}^{L}\left[x^s - \frac{1}{L}(\sum_{s=1}^{L}x^s)\right]^2}}$$

2.2.1.3 Fourier approximation

J. Fourier (1768-1830) is a French mathematician and physicist who improved a way to express a function in terms of an infinite number of sine and cosine terms. Fourier series expansion, which is valid for periodical, piecewise continuous and square integrable functions, represents the function in terms of the discrete sequence of the sinusoidal functions whose frequencies are indeed, integer multiples of the frequency of the periodic function. It reveals the frequency content of the function as providing the amplitude and phase information of the constituting frequency components. For a function $f(\cdot): R \to R$ with period T which satisfies the well known Dirichlet conditions described in terms of the piecewise continuity and square integrability of the function, a Fourier series expansion can be represented by:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(kw_0 x) + b_k \sin(kw_0 x)] \tag{2.70}$$

Where, $w_0 = \frac{2\pi}{T}$ is called the *fundamental frequency* and its constant multiples $2w_0, 3w_0, \cdots$ etc., are called *harmonics*. Hence, (2.70) represents $f(\cdot): R \to R$ as a linear combination of the set of orthogonal basis functions $\{1, \cos(kw_0 x), \sin(kw_0 x)\}_{k=1}^{\infty}$ The coefficients a_0, a_k and b_k in (2.70) can be computed by inner product defined in the continuous domain by the following integrals:

$$a_0 = \frac{2}{T} \int_0^T f(x)\, dx \tag{2.71}$$

$$a_k = \frac{2}{T} \int_0^T f(x) \cos(kw_0 x)\, dx \tag{2.72}$$

$$b_k = \frac{2}{T} \int_0^T f(x) \sin(kw_0 x)\, dx \tag{2.73}$$

For a function $f(\cdot): R \rightarrow R$ given by the set $\{(x^s, y^s)\}_{s=1}^{L}$, the following truncated Fourier series can be used for obtaining an approximation

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{m} [a_k \cos(w_k x) + b_k \sin(w_k x)] \tag{2.74}$$

Where, w_k can be chosen as $w_k = k w_0$ for periodic functions. It is known that quasi-periodic functions have such an exact representation and almost periodic functions can be approximated well by such a finite series (Bohr, 1947).

Finding a_0, a_k and $b_k{}'s$ coefficients is a special case of the procedure given in Subsection (2.2.1.1) for the most general setting. It should also be noted that determination of representative coefficients can be calculated in a easier way if the trigonometric functions are already orthogonalized in the discrete space as done in the previous subsection for the polynomial basis case.

2.2.1.4 Wavelet approximation

Wavelet series and transforms were developed to overcome the shortcomings of the Fourier series and transform (Morlet, Arens, Fourgeau, & Giard, 1982; Grossmann, & Morlet, 1984). Fourier series employes basis functions with an infinite duration (full support but not localized) in the time domain, although it gives a sharp precision in the frequency domain. In contrast, the wavelet approximation decomposes a function onto wavelets which are localized both in the time and the frequency domain.

A function in the wavelet series is represented by using a set of orthonormal basis function. $\psi_{i,j}(x) \in L_2\ (R)$ are indeed shifted (translated) and scaled (dilated) versions of a basis function $\psi(\cdot): R \rightarrow R$ which are so called wavelet or mother wavelet:

$$\psi_{i,j}(x) = 2^{i/2} \psi(2^i x - j) \qquad with\ i, j \in Z \tag{2.75}$$

Where, the terms i and j are the scaling and the shifting parameter, respectively. For a function $f(\cdot): R \to R$ given by the set $\{(x^s, y^s)\}_{s=1}^{L}$, the following truncated wavelet series can be used for obtaining an approximation.

$$f(x) = \sum_{i=-k}^{m} \sum_{j=-l}^{p} c_{ij} \psi_{i,j}(x) \tag{2.76}$$

Where, c_{ij} are the wavelet coefficients.

Finding $c_{ij}'s$ coefficients is a special case of the procedure given in Subsection (2.2.1.1) for the most general setting.

Approximation with orthogonal or orthogonalized bases functions like polynomial, Fourier and wavelets suffer from the necessity of excess number of coefficients for general data sets.

In this context, non-orthogonal basis functions are also widely used in the literature. The next section presents three non-orthogonal basis set examples, namely artificial neural networks as nonlinear regressions, support vector repressors and piecewise affine repressors.

There is no basis function set which has good approximation ability and implementation efficiency for all kind of data sets. Polynomial based approximations have good local approximation ability. Fourier series based approximations have a powerful global representation property for stationary functions possessing periodical changes. Wavelet series based representations have the capability of representing non-stationary periodical changes together with the localization property not only in the spectral domain but also in the original domain space. A similar comparison can be done from the implementation point of view such as in terms of numerical and hardware and/or software realization issues.

2.2.2 Approximating functions by non-orthogonal basis functions

Approximating functions by non-orthogonal basis functions can be implemented similar to the orthogonal basis functions case as explained in the Section 2.2.1.1. Artificial neural networks, support vector regression and also piecewise affine functions constitute such kind of approximations.

2.2.2.1 Artificial neural networks

Artificial Neural networks (ANNs) have been used for diverse applications including pattern recognition, classification, identification, control, interpolation and function approximation (regression) problems over the last three decades. There are many efficient ANN architectures and many associated efficient learning algorithms for designing them by a finite set of training data with providing a powerful generalization ability of responding well for the test data not learned before. Two of the most important ones of these architectures: Multi Layer Perceptrons (MLP) and Radial Basis Function Networks (RBFN).

ANNs can learn in supervised or in unsupervised ways depending on the availability of data class labels, or on desired outputs in a more general setting. The experimental knowledge is coded (stored) in the connection weights associated to the set of interconnected *neurons* which are the functional units of the ANN. The knowledge stored in the network can be modified by changing the values of the weights according to a learning rule. Learning, which is the process of determining the connection weights, is defined as an optimization problem where the cost function is the difference between desired and actual outputs for supervised learning cases and the quantization error between the learned prototype pattern and the sample data for unsupervised learning cases.

MLP is a multilayer, algebraic network of neurons called as perceptrons which are multi-input, single-output functional units taking firstly a weighted sum of their inputs and add bias then pass it through the activation function to form its output (See Figure 2.2). The architectural structure of an MLP network consists of one hidden layer and an output layer where neurons are fully connected. Signal flow in a feedforward way from left to right and on a layer by layer basis is depicted in Figure 2.2 (Haykin,1999).

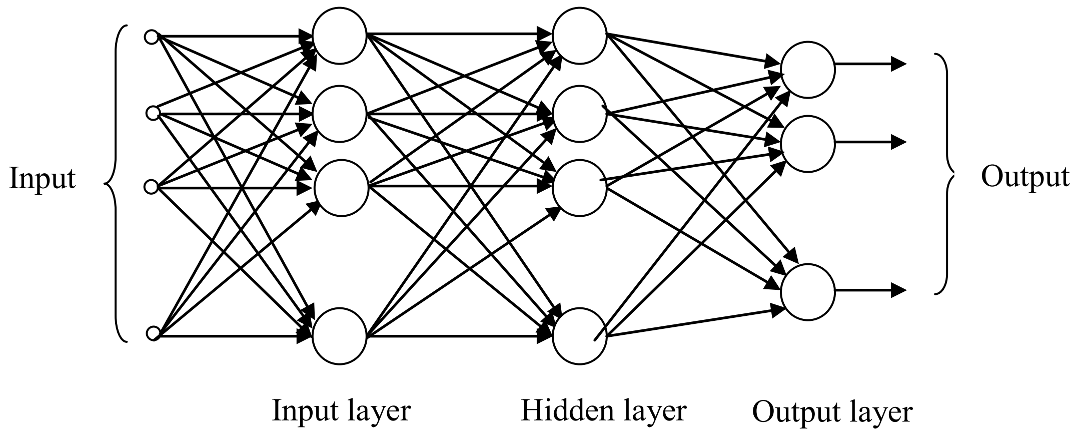


Figure 2.2 Architectural structure of a multi layer perceptron with one hidden layer

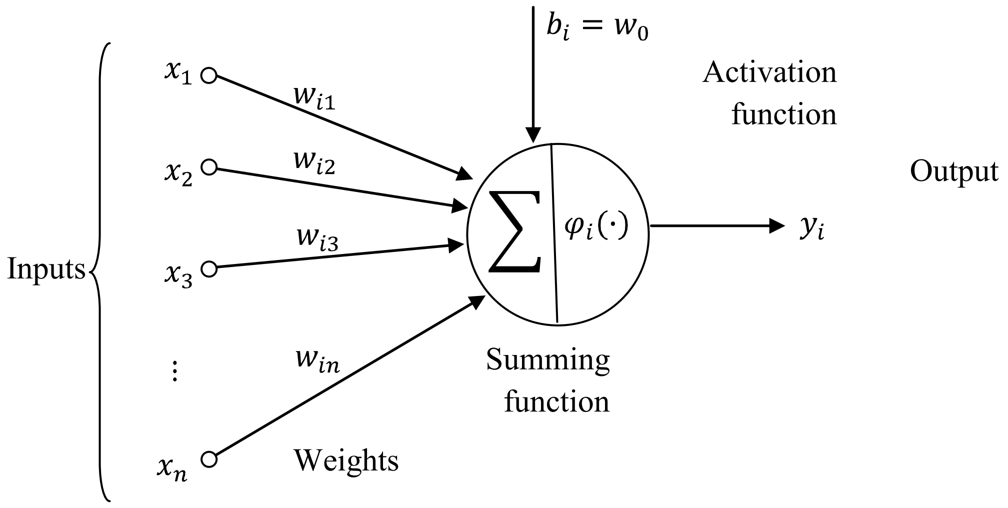


Figure 2.3 Structure of a neuron

As shown in Figure 2.3 output of the *i* th neuron with *n* inputs can be given by:

$$y_i = \varphi_i\left(\sum_{j=1}^{n} w_{ij}\, x_j + w_0\right) \tag{2.77}$$

Where, $\varphi_i(\cdot): R \rightarrow R$ is an activation function, w_{ij}'s are the weights, w_0 is the bias and x_j is the j'th input of the neuron. There are many different types of activation functions (i.e., threshold, piecewise linear, sigmoidal etc.) which are preferrable depending on the type of problem under consideration. Sigmoidal activation function is the most widely used for function approximation (regression) problems. [Cybenko, 1989; Jones, 1990; Ferrari & Stengel, 2005]. A unipolar sigmoidal function is defined by:

$$\varphi(x) = \frac{1}{1 + e^{-ax}} \tag{2.78}$$

Where, a is the slope parameter of the sigmoidal function. By varying the parameter a, one can obtain sigmoidal functions with different slopes as illustrated in Figure 2.4.

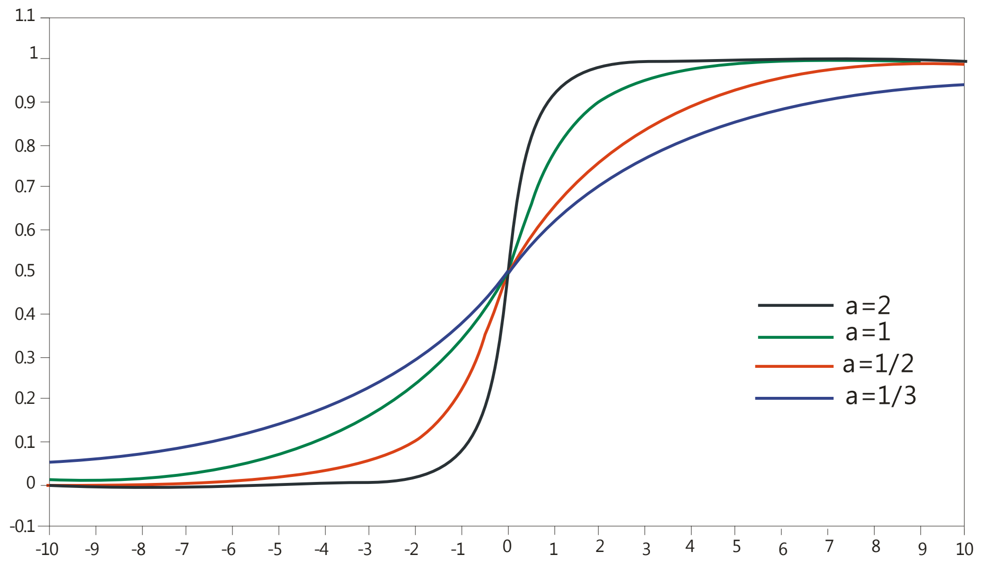


Figure 2.4 Sigmoidal functions with different slopes

Function approximation problems can be successively solved via MLP and RBF networks. Because of their parallel architecture and nonlinear structure, they handle nonlinear, noisy and imprecise data quite well. Cybenko, (1989) and Hornik, (1989) have proved that MLPs are universal function approximators which are capable of approximating to any continuous function in a compact set within arbitrary degree of accuracy, provided that a sufficient number of hidden layer neurons are used.

Theorem 2.6 (Universal approximation for one layer perceptron) [Cybenko, 1989; Hornik, Stinchcombe, & White, 1989; Funahashi, 1989]: Let $\varphi(\cdot)$ be any continuous sigmoidal function. Let I_n denote the n dimensional unit hypercube $[0,1]^n$. The space of continuous function on I_n is denoted by $C(I_n)$. Then,

$$F(\mathbf{x}) = \sum_{j=1}^{m} w_j\, \varphi\left(\mathbf{v}_j^T \mathbf{x} + b_j\right) \tag{2.79}$$

can approximate to any continuous function $f(\cdot): I_n \to R$ within an arbitrary accuracy by choosing sufficiently large number of hidden neurons m. In other words, given $f(\cdot) \in C(I_n)$ and $\varepsilon > 0$, there is $F(\mathbf{x})$ of the above form so that:

$$|F(\mathbf{x}_1, \cdots, \mathbf{x}_n) - f(\mathbf{x}_1, \cdots, \mathbf{x}_n)| < \varepsilon, \text{ for all } \mathbf{x} \in I_n \qquad \square$$

MLP is usually designed by determining the connection weights w_{ij}'s using the error Back Propagation (BP) algorithm which is indeed a gradient descent technique used for finding an acceptable local minimum of the squared error between the desired and actual outputs. The error at the output of neuron i is defined by:

$$e_i = [d_i - y_i] \tag{2.80}$$

Where, e_i represents the error for a specific data sample. The total error function is obtained by summing up of square of the errors e_i's obtained at the output layer neurons. So, one may write:

$$E = \frac{1}{2} \sum_{i \in \sigma} [y_i - d_i]^2 \tag{2.81}$$

Where, σ represents all neurons in the output layer. The error function can be minimized by adjusting each weight by the BP algorithm that calculates the partial derivatives of the output error in Equation (2.81) with respect to the connection weights by employing chain rule as shown in (2.82).

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial e_i}\frac{\partial e_i}{\partial y_i}\frac{\partial y_i}{\partial \varphi_i}\frac{\partial \varphi_i}{\partial w_{ij}} \tag{2.82}$$

The partial derivatives calculated are then used for updating the connection weights in the opposite of the gradient direction towards one of the local minima with a sufficiently small step size η, called also as learning rate. Update of the weights for the k th iteration is given by the following difference equation:

$$w(k+1) = w(k) - \eta\frac{\partial E[k]}{\partial w_{ij}} \tag{2.83}$$

That update of the weights is implemented in two different ways. In the pattern mode BP, the connection weights are changed for each sample. On the other hand, batch mode BP allows an update for the whole set of training samples once at each time instant which requires summing up the individual gradients obtained for each specific sample to take a step. The convergence of gradient-descent algorithms can be shown by using the so called descent Lemma (Bertsekas,1995).

Lemma 2.1 (Descent) [Bertsekas, 1995]: Given a continuously differentiable scalar function $f(\cdot): R^n \to R$. If its gradient $\nabla f(\mathbf{x})$ is Lipschitz continuous with Lipschitz constant K, i.e.,

$$\exists\, K < \infty, \qquad \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \le K \cdot \|\mathbf{x}-\mathbf{y}\|_2 \quad \forall\, \mathbf{x},\mathbf{y} \in R^n$$

then

$$f(\mathbf{x}+\mathbf{y}) \le f(\mathbf{x}) + \mathbf{y}^{\mathrm{T}}\nabla f(\mathbf{x}) + \frac{K}{2}\|\mathbf{y}\|_2^2$$

□

Theorem 2.7 (Convergence) [Bertsekas, 1995]: Assume th at $f(\cdot) \in C^1$ is a scalar function d efined on a c ompact s et. If t he l earning factor η is c hosen i n t he interval of $(0, {}^{2}/_{K})$ with Lipschitz c onstant K for $\nabla f(\mathbf{x})$, t hen t he di fference equation

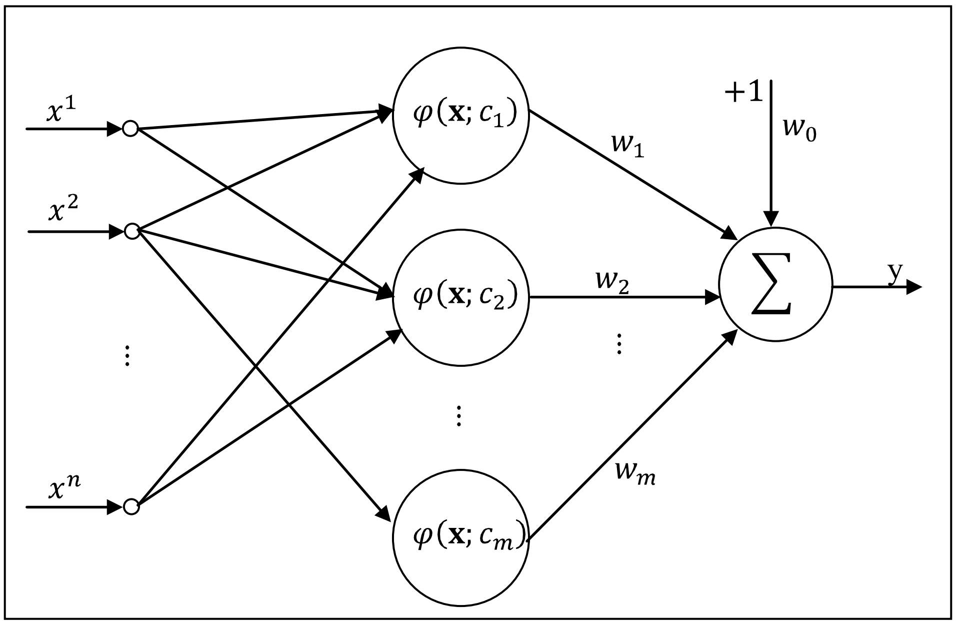
$$\mathbf{x}(k+1) = \mathbf{x}(k) - \eta \nabla f(\mathbf{x}(k))$$

produces a sequence of points $\mathbf{x}^{k\prime}$s converging to one of the minima of $f(\mathbf{x})$.

□

Due t o the nonc onvexity of t he t otal s quare e rror function, BP al gorithm can converge to any o f lo cal min ima. Determining t he ne twork s ize f or t he M LP, i n particular, the number of hidden layer neurons and also the number of neurons in the hidden layer is a difficult problem in general.

RBFN is used as an alternative ANN model in the literature. Its first layer neuron parameters, i.e., t he cen ters and w idths of t he G aussian a ctivation functions can b e learned i n an uns upervised w ay ve ry efficiently. T hen, t he s econd l inear l ayer neurons c an be t rained by an e fficient l inear w eight upda te r ule. T his pos sibility i n the de sign of R BFN m akes s uperior i t ov er ot her A NN m odels. Locally t unable property o f t he first l ayer G aussian n eurons of t he R BFN i s a nother r eason for t he wide spread use of RBFNs.


x^1
x^2
x^n
$\varphi(\mathbf{x}; c_1)$
$\varphi(\mathbf{x}; c_2)$
$\varphi(\mathbf{x}; c_m)$
w_1
w_2
w_m
$+1$
w_0
$\sum$
y
Figure 2.5 Architecture of the Radial Basis Function Network

RBFNs consist of two layers interconnected in a feedforward way as shown Figure 2.4. *The input layer* is made up of its input neurons where each neuron applies a nonlinear transformation, called radial basis activation, to the inputs and *the output layer* which is made up of weighted sum of the outputs of the input neurons.

The radial basis activation function is given by:

$$\varphi_j = \varphi_j(\|\mathbf{x} - \mathbf{c}_j\|) \tag{2.84}$$

where c_j is the center of the RBF, $\|\mathbf{x} - \mathbf{c}_j\|$ is the Euclidean distance between the $\mathbf{x}$ and $\mathbf{c}_j$ for neuron j, and $\varphi_j(\cdot): R^n \to R$ is chosen typically as Gaussian function:

$$\varphi_j(\mathbf{x}; \mathbf{c}_j) = exp\left(-\frac{1}{\sigma_j^2} \|\mathbf{x} - \mathbf{c}_j\|^2\right) \tag{2.85}$$

where, σ_j is the width of the RBF.

The outputs of the RBFN are simply the weighted sum of the outputs of the neurons in the input layer as depicted Figure 2.4.

$$y = \sum_{j=1}^{m} w_j \, \varphi_j(\mathbf{x}; c_j) + w_o \tag{2.86}$$

If (2.88) is substituted into (2.89), then one may formulate the input-output mapping realized by the Gaussian RBF as follows:

$$y = \sum_{j=1}^{m} w_j \, exp\left(-\frac{1}{\sigma_j^2} \|\mathbf{x} - c_j\|_2^2\right) + w_o \tag{2.87}$$

Theorem 2.8 (Universal approximation for RBFN) [Park & Sandberg, 1991]: Let $\varphi(\cdot): R^n \to R$ be an integrable bounded function such that $\varphi(\cdot)$ is continuous almost everywhere and $\int_{R^n} \varphi(x)\, dx \neq 0$. Let I_φ denote the family of RBF networks consisting of functions $F(\cdot): R^n \to R$ represented by

$$F(\mathbf{x}) = \sum_{j=1}^{m} w_j\, \varphi\left(\frac{\mathbf{x} - \boldsymbol{c}_j}{\sigma}\right)$$

Where, $\sigma > 0$, $w_j \in R$ and $\boldsymbol{c}_j \in R^n$ for $j \in \{1,2, \cdots m\}$.

For any continuous mapping $f(\cdot): R^n \to R$ there is an RBF network with a set of centers $\{\boldsymbol{c}_j\}_{j=1}^{m}$ and a common width $\sigma > 0$ such that the input-output mapping $F(\cdot): R^n \to R$ realized by the RBF network is close to $f(\cdot)$ in the L_p norm with $p \in [1, \infty]$. In other words, given $f(\cdot) \in (I_\varphi)$ and $\varepsilon > 0$, there is $F(\cdot)$ of the above form so that

$$|F(\mathbf{x}_1, \cdots, \mathbf{x}_n) - f(\mathbf{x}_1, \cdots, \mathbf{x}_n)| < \varepsilon, \qquad \text{for all } \mathbf{x} \in (I_n)$$

□

For a given sample set $\{(\mathbf{x}^s, y^s)\}_{s=1}^{L}$ with $\mathbf{x}^s \in R^n$ and $y^s \in R$, one can determine the linear connection weights w_j's with $j \in \{0,1,2, \dots, m\}$ by solving the following linear algebraic equation system under the assumption that the centers and widths are known.

$$\begin{bmatrix} 1 & e^{-\frac{1}{\sigma_1^2}\|\mathrm{x}^1 - c_1\|^2} & e^{-\frac{1}{\sigma_2^2}\|\mathrm{x}^1 - c_2\|^2} & \cdots & e^{-\frac{1}{\sigma_m^2}\|\mathrm{x}^1 - c_m\|^2} \\ 1 & e^{-\frac{1}{\sigma_1^2}\|\mathrm{x}^2 - c_1\|^2} & e^{-\frac{1}{\sigma_2^2}\|\mathrm{x}^2 - c_2\|^2} & \cdots & e^{-\frac{1}{\sigma_m^2}\|\mathrm{x}^2 - c_m\|^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{1}{\sigma_1^2}\|\mathrm{x}^L - c_1\|^2} & e^{-\frac{1}{\sigma_2^2}\|\mathrm{x}^L - c_2\|^2} & \cdots & e^{-\frac{1}{\sigma_m^2}\|\mathrm{x}^L - c_m\|^2} \end{bmatrix} \begin{bmatrix} w_o \\ w_1 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^L \end{bmatrix} \qquad (2.88)$$

In order to design RBFN, one may use a BP like gradient descent algorithm for determining all network parameters. The common learning strategy in the RBFN design is to employ hybrid learning that is to determine firstly the centers and widths of the first layer Gaussian neurons and then apply the least mean square rule or generalized inverse for learning linear weights of the output neuron. Herein, centers can be assigned fixed, in a random way, from the input samples or can be calculated by a clustering method applied on input samples (Moody, J. E. & Darken, C. J., 1989; Musavi, M. T., Ahmed, W., Chan, K. H., Faris, K. B. & Hummels, D. M. 1992) or on input-output samples (Uykan, Z., Güzeliş, C. & Çelebi, M.E., 2000).

2.2.2.2 Support Vector Machines

Support Vector Machines (SVMs) were initially developed to solve decision problems then applied to classification (Vapnik & Lerner, 1963; Vapnik & Chervonenkis 1974), regression (Vapnik, 1995, Gunn, 1998), and clustering (Ben-Hur, A., Horn, D., Siegelmann, H.T. & Vapnik, V., 2001). Aim of SVMs in classification is to find an optimal separating boundary which has good generalization ability for a given set of input-output data mapped into a high dimensional feature space. The optimal separating boundary is, indeed, represented by a small subset of the whole training data, called Support Vectors (SV), and found by convex optimization methods.

The superiority of SVMs over ANN models is due to the possibility of determining regression model in terms of only some samples called support vectors and a compact kernel representation and also due to their better generalization ability which is achieved by minimizing not only the training error but also a norm of the model parameters to obtain a less complex model. Vapnik (1995) defined the following epsilon (ε) insensitive loss function that ignore errors which are within a determined ε distance of desired outputs (Figure 2.6).

$$L_\varepsilon(y, f(\mathbf{x})) = \begin{cases} |\, y - f(\mathbf{x}) \,| - \varepsilon, & for \;\; |\, y - f(\mathbf{x}) \,| \geq \varepsilon \\ 0 & otherwise \end{cases} \tag{2.89}$$

There are many different loss functions other than ε insensitive such as Laplace, quadratic, Huber, etc. used in the literature.

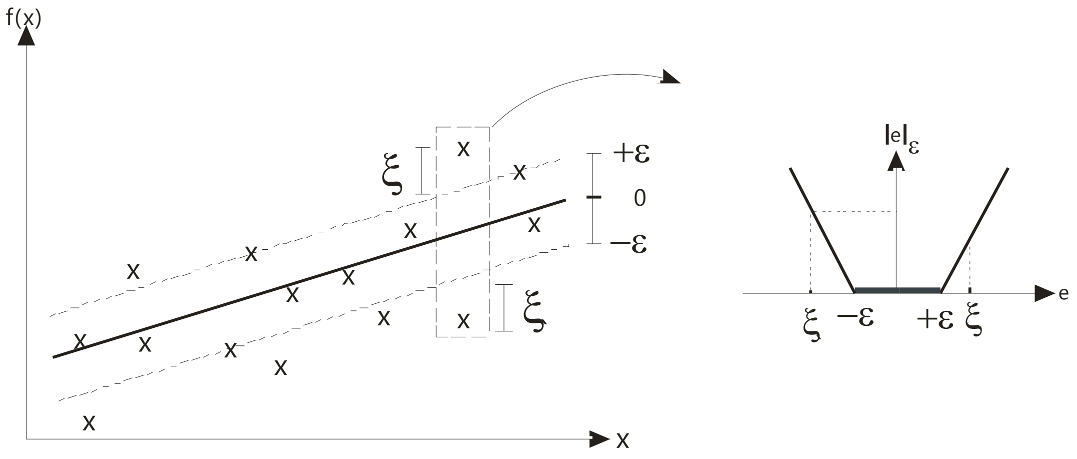


Figure 2.6 ε-insensitive loss function

Linear Support Vector Regression (LSVR)

Let a function $f(\cdot): R^{\mathrm{n}} \to R$ to be approximated be given by a finite set of input-output data pairs $\{(\mathbf{x}^s, y^s)\}_{s=1}^{L}$ with $\mathbf{x}^s \in R^{\mathrm{n}}$ and $y^s \in R$. Support vector machine for linear regression attempts to find a function which is given by

$$f(\mathbf{x}) = \boldsymbol{w}^T\mathbf{x} + b \tag{2.90}$$

such that it should be at most ε deviation from all of the measurements y^s for all the training data and, at the same time, the hyperplane defined in the feature space is as flat a s pos sible. This r egression pr oblem c an be f ormulated a s t he f ollowing optimization:

$$\min R_{emp} = \frac{1}{L}\sum_{s=1}^{L} L_{\varepsilon}(y^s, f(\mathbf{x^s})) \tag{2.91}$$
$$\text{subject to inequality} \quad \| \boldsymbol{w} \|_2^2 \leq c_0$$

Where, c_0 is a constant. The ε-insensitive loss function $L_{\varepsilon}(y^s, f(\mathbf{x^s}))$ will be denoted as $|e|_{\varepsilon}$ in this thesis. $|e^s|_{\varepsilon}$ is a PWA continuous function, so can also be represented by the following canonical form in terms of the absolute value functions:

$$|e|_\varepsilon = -\varepsilon + \frac{1}{2} | e - \varepsilon | + \frac{1}{2} | e + \varepsilon |$$

Flatness of the h yperplane de fined b y (2.90) can be ensured by m inimizing t he squared Euclidean nor m $\| \boldsymbol{w} \|_2^2$. S o, one may reformulate th is c onstrained optimization pr oblem a s a convex c ost function of $\boldsymbol{w}$ and c onstraints b eing l inear in $\boldsymbol{w}$. Thus, so called primal optimization problem is obtained as:

$$\min_{\boldsymbol{w} \epsilon R^m} \frac{1}{2} \| \boldsymbol{w} \|_2^2 + C \sum_{s=1}^{L} |e^s|_\varepsilon \tag{2.92}$$

In t he a bove primal o ptimization formulation, the e rrors du e t o t he i nput-output sample d ata ar e m inimized i n the av erage s ense. The following p rimal o ptimization problem w hich i s t he standard f ormulation of l inear s upport ve ctor regression formulation (SVR) (Vapnik, 1995) i s obt ained by i ntroducing penalization s lack variables $\{\xi_s\}_{s=1}^L$ and $\{\xi_s'\}_{s=1}^L$.

$$\min_{\boldsymbol{w} \epsilon R^m} \frac{1}{2} \| \boldsymbol{w} \|_2^2 + C \sum_{s=1}^{L} (\xi_s + \xi_s')$$

$$\text{subject to} \begin{cases} y^s - \boldsymbol{w}^T \mathbf{x}^s - b \leq \varepsilon + \xi_s \\ -y^s + \boldsymbol{w}^T \mathbf{x}^s + b \leq \varepsilon + \xi_s' \\ \xi_s \xi_s' \geq 0 \text{ with } s \in \{1, \dots, L\} \end{cases} \tag{2.93}$$

where the constant parameter C is a u ser specified p arameter an d controls the trade-off b etween t he f latness, s o the generalization a bility (small C) an d low e mpirical error (large C). T he p arameter ε controls the w idth of t he ε-insensitive t ube. Increasing ε reduces t he num ber of s upport ve ctors, so y ielding sparseness which produces a smoother f unction. O n t he ot her ha nd, i f ε is i ncreased t oo much, t he fitting error becomes unacceptable large. Thus, the user specified parameters ε and C controls the model complexity in two different ways.

This p rimal o ptimization p roblem c an b e s olved by transforming it in to the following dual form by the method of Lagrange multipliers (Bertsekas, 1995).

$$\begin{aligned}\min_{\substack{\boldsymbol{w}\in R^m, b\in R\\ \alpha_s,\alpha_s',\gamma_s,\gamma_s',\xi_s,\xi_s'\geq 0\\ s\in\{1,\dots,L\}}} \quad & J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s') \\ & = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{s=1}^{L}(\xi_s + \xi_s') \\ & -\sum_{s=1}^{L}\alpha_s(\varepsilon + \xi_s - y^s + \boldsymbol{w}^T\mathbf{x}^s + b) \\ & -\sum_{s=1}^{L}\alpha_s'(\varepsilon + \xi_s' + y^s - \boldsymbol{w}^T\mathbf{x}^s - b) \\ & -\sum_{s=1}^{L}(\gamma_s\xi_s + \gamma_s'\xi_s')\end{aligned} \tag{2.94}$$

where, the nonne gative variables α_s, α_s' and γ_s, γ_s' are called as dual variables or Lagrange mu ltipliers. The s olution of t he pr oblem (2.93) is d etermined b y finding saddle poi nts of t he Lagrangian f unction $J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s')$ via partial differentiating it w ith r espect to th e p rimal v ariables $(\boldsymbol{w}, b, \xi_s, \xi_s')$ and s etting t he resulting gradients to zero.

$$\frac{\partial J}{\partial b} = \sum_{s=1}^{L}(\alpha_s' - \alpha_s) = 0 \tag{2.95}$$

$$\nabla_{\boldsymbol{w}} J = \boldsymbol{w} - \sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\mathbf{x}^s = 0 \tag{2.96}$$

$$\frac{\partial J}{\partial \xi_s} = C - \alpha_s - \gamma_s = 0 \tag{2.97}$$

$$\frac{\partial J}{\partial \xi_s'} = C - \alpha_s' - \gamma_s' = 0 \tag{2.98}$$

Now, substituting (2.96) into (2.94), one obtains:

$$
\begin{aligned}
&J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s') \\
&\quad = \frac{1}{2}\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\mathbf{x}^s\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\mathbf{x}^r\right] \\
&\quad - \left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\mathbf{x}^s\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\mathbf{x}^r\right] \\
&\quad - \varepsilon\sum_{s=1}^{L}(\alpha_s + \alpha_s') + \sum_{s=1}^{L} y^s \cdot (\alpha_s - \alpha_s') + b\sum_{s=1}^{L}(\alpha_s' - \alpha_s) \\
&\quad + C\sum_{s=1}^{L}(\xi_s + \xi_s') - \sum_{s=1}^{L}\alpha_s\,\xi_s - \sum_{s=1}^{L}\alpha_s'\,\xi_s' \\
&\quad - \sum_{s=1}^{L}(\gamma_s\xi_s + \gamma_s'\xi_s')
\end{aligned}
\tag{2.99}
$$

Then, s ubstituting (2.95), (2.97) and (2.98) into the above formula, (2.100) is obtained.

$$
\begin{aligned}
&J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s') \\
&\quad = -\frac{1}{2}\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\mathbf{x}^s\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\mathbf{x}^r\right] \\
&\quad - \varepsilon\sum_{s=1}^{L}(\alpha_s + \alpha_s') + \sum_{s=1}^{L} y^s \cdot (\alpha_s - \alpha_s') + C\sum_{s=1}^{L}(\xi_s + \xi_s') \\
&\quad - \sum_{s=1}^{L}\xi_s(\alpha_s + \gamma_s) - \sum_{s=1}^{L}\xi_s'(\alpha_s' + \gamma_s')
\end{aligned}
\tag{2.100}
$$

Finally, after p erforming s everal s implifications th e following dual o ptimization formulation is obtained for the linear SVR.

$$\max_{\alpha_s,\alpha_s'} J(\boldsymbol{w},b,\alpha_s,\alpha_s',\gamma_s,\gamma_s',\xi_s,\xi_s') =$$

$$-\frac{1}{2}\sum_{s=1}^{L}\sum_{r=1}^{L}(\alpha_s-\alpha_s')(\alpha_r-\alpha_r')(\mathbf{x}^s)^{\mathrm{T}}\mathbf{x}^r$$

$$-\varepsilon\sum_{s=1}^{L}(\alpha_s+\alpha_s')+\sum_{s=1}^{L}y^s(\alpha_s-\alpha_s') \tag{2.101}$$

$$\text{subject to}\begin{cases}\sum_{s=1}^{L}(\alpha_s'-\alpha_s)=0\\ \alpha_s',\alpha_s\in[0,C]\end{cases}$$

Where, ε and C are free parameters to be specified by the user which control the generalization ability of the approximation and ${\mathbf{x}^s}^{\mathrm{T}}\mathbf{x}^r$ is the inner product of the input sample vector with the input variable vector. The dual optimization problem in (2.101) can be rewritten in the following matrix form:

$$\min_{\alpha_s,\alpha_s'}\frac{1}{2}\begin{bmatrix}\alpha\\ \alpha'\end{bmatrix}^T\begin{bmatrix}\mathbf{XX}^{\mathrm{T}} & -\mathbf{XX}^{\mathrm{T}}\\ -\mathbf{XX}^{\mathrm{T}} & \mathbf{XX}^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\alpha\\ \alpha'\end{bmatrix}+\begin{bmatrix}\boldsymbol{\varepsilon u}-\mathbf{y}\\ \boldsymbol{\varepsilon u}+\mathbf{y}\end{bmatrix}^T\begin{bmatrix}\alpha\\ \alpha'\end{bmatrix}$$

$$\text{subject to}\left\{\begin{bmatrix}\boldsymbol{u}\\ -\boldsymbol{u}\end{bmatrix}^T\begin{bmatrix}\alpha\\ \alpha'\end{bmatrix}=0\ and\ \alpha_s',\alpha_s\in[0,C]\right. \tag{2.102}$$

Where, $\boldsymbol{u}=[1\ 1\ \cdots 1]^T\in R^L, \mathbf{y}=[y^1\ y^2\cdots y^L]^{\mathrm{T}}\in R^L$ and $\mathbf{X}^{\mathbf{T}}\triangleq[\mathbf{x}^1\ \ \mathbf{x}^2\ \cdots\ \mathbf{x}^L]\in R^{n\times L}$.

The Karush-Kuhn-Tucker (KKT) conditions (Karush, 1939; Kuhn and Tucker, 1951) are $\alpha_s\alpha_s'=0$ with $s\epsilon\{1,2,\cdots.L\}$. The support vectors are points where exactly one of the α_s,α_s' Lagrange multipliers is greater than zero.

Once Lagrange multipliers are found by solving (2.102) one can determine the optimal $\boldsymbol{w}$ as follows. Equation (2.96) can be written as:

$$\boldsymbol{w}=\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,\mathbf{x}^s \tag{2.103}$$

and the support vector regression function (2.90) becomes

$$f(\mathbf{x}) = \boldsymbol{w}^T\mathbf{x} + b = \sum_{s=1}^{L} (\alpha_s - \alpha_s')\, (\mathbf{x}^s)^{\mathrm{T}}\mathbf{x} + b \tag{2.104}$$

Computing *b*

The bias term b is computed by considering the KKT conditions as follows. The product of dual variables with constraints should be equal to zero at the optimal:

$$\begin{aligned} \alpha_s(\varepsilon + \xi_s - y^s + \boldsymbol{w}^T\mathbf{x}^s + b) &= 0 \\ \alpha_s'(\varepsilon + \xi_s' + y^s - \boldsymbol{w}^T\mathbf{x}^s - b) &= 0 \end{aligned} \tag{2.105}$$

and

$$\begin{aligned} (C - \alpha_s)\xi_s &= 0 \\ (C - \alpha_s')\xi_s' &= 0 \end{aligned} \tag{2.106}$$

(2.105) and (2.106) imply that only samples $(\mathbf{x}^s, y^s)$ with corresponding $\alpha_s = C$ and $\alpha_s' = C$ lie outside the ε-insensitive tube. Another conclusion is described as $\alpha_s\alpha_s' = 0$, i.e, there can never be a set of dual variables α_s, α_s' which are both simultaneously nonzero because this would require nonzero slacks in both directions (Smola & Schölkoph, 1998). If $\alpha_s\alpha_s' \neq 0$, there is a contradiction due to the definition of function that can never be multi-valued range data for a single sample data in the domain. So, one can conclude that

$$\varepsilon - y^s + \boldsymbol{w}^T\mathbf{x}^s + b \geq 0 \quad \text{and} \quad \xi_s = 0 \qquad \text{if } \alpha_s < C \tag{2.107}$$

$$\varepsilon - y^s + \boldsymbol{w}^T\mathbf{x}^s + b \leq 0 \qquad \text{if } \alpha_s > 0 \tag{2.108}$$

A similar analysis on α_s' yields

$$\begin{aligned} &\max\{-\varepsilon + y^s - \boldsymbol{w}^T\mathbf{x}^s \mid \alpha_s < C \text{ or } \alpha_s' > 0\} \leq b \\ &\qquad \leq \min\{-\varepsilon + y^s - \boldsymbol{w}^T\mathbf{x}^s \mid \alpha_s > 0 \text{ or } \alpha_s' < C\} \end{aligned} \tag{2.109}$$

For some dual variables $\alpha_s, \alpha'_s \in (0, C)$, the inequalities become equalities. That is,

$$\begin{aligned} b &= y^s - (\mathbf{x}^s)^{\mathrm{T}}(\mathbf{x}^r) - \varepsilon \quad for\ \alpha_s \in (0, C) \\ b &= y^s - (\mathbf{x}^s)^{\mathrm{T}}(\mathbf{x}^r) + \varepsilon \quad for\ \alpha'_s \in (0, C) \end{aligned} \tag{2.110}$$

The bias term b can be calculated by using (2.110) just for a specific support vector. One of the possible alternatives for calculating the bias term b is to use the following formula (Chuang, Su, Jeng, & Hsiao, 2002).

$$b = \frac{1}{2}\left\{\min_s\left(y^s - \sum_{s=1}^{L}(\alpha_s - \alpha'_s)\,(\mathbf{x}^s)^{\mathrm{T}}\mathbf{x}\right) + \min_s\left(y^s - \sum_{s=1}^{L}(\alpha_s - \alpha'_s)\,(\mathbf{x}^s)^{\mathrm{T}}\mathbf{x}\right)\right\}$$

Nonlinear support vector regression

Nonlinear support vector regression attempts to find a function which is given by:

$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j\, \emptyset_j(\mathbf{x}) + b = \boldsymbol{w}^T \emptyset(\mathbf{x}) + b \tag{2.111}$$

The constraint optimization problem can be formulated as:

$$\begin{aligned} &\min_{\boldsymbol{w} \in R^m} \frac{1}{2} \parallel \boldsymbol{w} \parallel_2^2 + C\sum_{s=1}^{L}(\xi_s + \xi'_s) \\ &\text{subject to} \begin{cases} y^s - \boldsymbol{w}^T\emptyset(\mathbf{x}^s) - b \le \varepsilon + \xi_s \\ -y^s + \boldsymbol{w}^T\emptyset(\mathbf{x}^s) + b \le \varepsilon + \xi'_s \\ \xi_s, \xi'_s \ge 0, \qquad s \in \{1, \dots, L\} \end{cases} \end{aligned} \tag{2.112}$$

This pr imal opt imization pr oblem c an be s olved by transforming i t i nto a dua l form via the method of Lagrange multipliers.

$$\min_{\substack{\boldsymbol{w}\in R^m, b\in R \\ \alpha_s,\alpha_s',\gamma_s,\gamma_s',\xi_s,\xi_s'\geq 0 \\ s\in\{1,\ldots,L\}}} J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s')$$

$$= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{s=1}^{L}(\xi_s + \xi_s') - \sum_{s=1}^{L}\alpha_s(\varepsilon + \xi_s - y^s + \boldsymbol{w}^T\emptyset(\mathbf{x}^s) + b) - \sum_{s=1}^{L}\alpha_s'(\varepsilon + \xi_s' + y^s - \boldsymbol{w}^T\emptyset(\mathbf{x}^s) - b) - \sum_{s=1}^{L}(\gamma_s\xi_s + \gamma_s'\xi_s') \tag{2.113}$$

Where, the nonnegative variables α_s, α_s' and γ_s, γ_s' are dual varaibles or Lagrange multipliers. The solution of the problem (2.113) is determined by finding the saddle points of Lagrangian function $J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s')$ via partial differentiating it with respect to the primal variables $(\boldsymbol{w}, b, \xi_s, \xi_s')$ and setting the results to zero.

$$\frac{\partial J}{\partial b} = \sum_{s=1}^{L}(\alpha_s' - \alpha_s) = 0 \tag{2.114}$$

$$\nabla_w J = \boldsymbol{w} - \sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s) = 0 \tag{2.115}$$

$$\frac{\partial J}{\partial \xi_s} = C - \alpha_s - \gamma_s = 0 \tag{2.116}$$

$$\frac{\partial J}{\partial \xi_s'} = C - \alpha_s' - \gamma_s' = 0 \tag{2.117}$$

Substituting (2.114), (2.115), (2.116) and (2.117) into Equation (2.113), one obtains the dual optimization problem after implementing several simplifications:

$$\max_{\alpha_s,\alpha_s'} J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s') = \begin{matrix} -\frac{1}{2}\sum_{s=1}^{L}\sum_{r=1}^{L}(\alpha_s - \alpha_s')(\alpha_r - \alpha_r')\emptyset^{\mathrm{T}}(\mathbf{x}^s)\cdot\emptyset(\mathbf{x}^r) \\ -\varepsilon\sum_{s=1}^{L}(\alpha_s + \alpha_s') + \sum_{s=1}^{L} y^s(\alpha_s - \alpha_s') \end{matrix} \tag{2.118}$$

$$\text{subject to}\begin{cases}\sum_{s=1}^{L}(\alpha_s' - \alpha_s) = 0 \\ \alpha_s, \alpha_s' \in [0, C]\end{cases}$$

Where, ε and C are free parameters to be specified by the user which control the generalization ability of the approximation.

Kernel Trick

The algorithms described in the previous section construct the linear regression for a given set of samples $\{(\mathbf{x}^s, y^s)\}_{s=1}^{L}$ in the input space. To construct a linear regression in a feature space (which corresponds to nonlinear regression in the input space), one has to use a nonlinear function $\emptyset(\cdot) \in \mathcal{H}$ mapping from an input space $\mathcal{X}$ into a feature space $\mathcal{F}$ (Aizerman, Braverman, & Rozonotr, 1964, Nilson, 1965). For a real space, $\emptyset(\cdot)$ can be given as

$$\emptyset(\cdot): R^n \to R^m \tag{2.119}$$

where, $m \gg n$.

Then, the mapped data set becomes:

$$\{(\emptyset(\mathbf{x}^s), y^s)\}_{s=1}^{L}, \qquad \emptyset(\mathbf{x}^s) \in R^m, \qquad \mathbf{x}^s \in R^n, \qquad y^s \in R \tag{2.120}$$

The mapping of each input data sample individually to the feature space leads to several problems. One of these problems is the inefficiency of the calculation of the mapped v ectors f or each s ample b y u sing t he v ector-valued nonl inear function $\emptyset(\cdot): R^n \to R^m$. T herefore, instead o f m apping e ach i nput da ta s ample $\mathbf{x}^s$ into the feature space yielding $\emptyset(\mathbf{x}^s)$ and then calculating the inner products $\emptyset^{\mathrm{T}}(\mathbf{x}^s) \cdot \emptyset(\mathbf{x}^r)$, the following so called kernel function $\mathrm{K}(\cdot,\cdot)$ is commonly used in the SVR literature.

$$\mathrm{K}(\cdot,\cdot): R^m \times R^m \to R \tag{2.121}$$

The regression problem can be defined and solved in terms of such a kernel function without know ing t he a ssociated m apping $\emptyset(\cdot)$. W hen t he m apping i s know n t he determination of the corresponding kernel is straight forward. On the other hand, the Mercer's T heorem given be low pr ovides t he conditions unde rwhich f or a ke rnel there exists such a mapping $\emptyset(\cdot)$ (Vapnik, 1995; Courant & Hilbert, 1953).

Theorem 2.9 (Mercer's Conditions) [Mercer, 1909]:

For a function $\mathrm{K}(\cdot,\cdot): R^n \times R^n \to R$ there exists a mapping $\emptyset(\cdot): R^n \to R^m$ such that $\mathrm{K}(\mathbf{x},\mathbf{y}) = \emptyset^{\mathrm{T}}(\mathbf{x}) \cdot \emptyset(\mathbf{y})$
where, m might be infinite, if and only if

$$\iint \mathrm{K}(\mathbf{x},\mathbf{y}) p(\mathbf{x}) q(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \tag{2.122}$$

for a ll s quared integrable f unctions $p(\cdot), q(\cdot) \in R^n \to R$ i.e. $\int p(\mathbf{x})^2 d\mathbf{x} < \infty$ and $\int q(\mathbf{x})^2 d\mathbf{x} < \infty$.

□

If one uses a kernel which does not satisfy Mercer's condition, then the Hessian may not be positive definite, so quadratic programming problem may have no solution.

Generally, there is more than one kernel to map the input space into the feature space. Typical examples of kernel functions are used in this thesis are given Table 2.5.

Table 2.5 Kernel functiona and representtions

Type of kernel	Representation of kernel
Polynomial	$\mathrm{K}(\mathbf{x^s},\mathbf{x}) = [\mathbf{x}^T\mathbf{x}^s + c]^p, \quad c \geq 0, \quad p > 1$
Gaussian	$\mathrm{K}(\mathbf{x^s},\mathbf{x}) = e^{-\left(\frac{\lVert\mathbf{x}-\mathbf{x}^s\rVert_2^2}{2\sigma^2}\right)}, \quad \sigma > 0$
Sigmoid	$\mathrm{K}(\mathbf{x^s},\mathbf{x}) = tanh(p.\mathbf{x}^T\mathbf{x}^s + q), \quad p > 0$ and $q < 0$
Fourier	$\mathrm{K}(\mathbf{x^s},\mathbf{x}) = \dfrac{1-p^2}{2(1-2pcos(\mathbf{x}^s-\mathbf{x}))} + p^2, \quad p > 0$
B_n-spline	$\mathrm{K}(\mathbf{x^s},\mathbf{x}) = B_n(\lVert\mathbf{x}-\mathbf{x}^s\rVert_2^2), \quad n = 2p+1 \; and \; n \geq 1$

For L samples, kernel function, which is, indeed, the function of two independent n-dimensional variables, define the kernel matrix by evaluating it at pair of sample points:

$$\mathbf{K}(\mathbf{x}^s,\mathbf{x}^r) = [\emptyset^{\mathrm{T}}(\mathbf{x}^s)\cdot\emptyset(\mathbf{x}^r)]_{\mathrm{s,r}}$$

$$= \begin{bmatrix} \mathrm{K}(\mathbf{x}^1,\mathbf{x}^1) & \mathrm{K}(\mathbf{x}^1,\mathbf{x}^2) & \cdots & \mathrm{K}(\mathbf{x}^1,\mathbf{x}^L) \\ \mathrm{K}(\mathbf{x}^2,\mathbf{x}^1) & \mathrm{K}(\mathbf{x}^2,\mathbf{x}^2) & \cdots & \mathrm{K}(\mathbf{x}^2,\mathbf{x}^L) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{K}(\mathbf{x}^L,\mathbf{x}^1) & \mathrm{K}(\mathbf{x}^L,\mathbf{x}^2) & \cdots & \mathrm{K}(\mathbf{x}^L,\mathbf{x}^L) \end{bmatrix} \qquad (2.123)$$

The dual optimization problem in (2.122) can be rewritten in the following matrix form:

$$\min_{\alpha,\alpha'} \frac{1}{2}\begin{bmatrix}\alpha \\ \alpha'\end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Phi}(\mathrm{x})\boldsymbol{\Phi}(\mathrm{x})^{\mathrm{T}} & -\boldsymbol{\Phi}(\mathrm{x})\boldsymbol{\Phi}(\mathrm{x})^{\mathrm{T}} \\ -\boldsymbol{\Phi}(\mathrm{x})\boldsymbol{\Phi}(\mathrm{x})^{\mathrm{T}} & \boldsymbol{\Phi}(\mathrm{x})\boldsymbol{\Phi}(\mathrm{x})^{\mathrm{T}} \end{bmatrix} \begin{bmatrix}\alpha \\ \alpha'\end{bmatrix} + \begin{bmatrix}\boldsymbol{\varepsilon u} - \mathbf{y} \\ \boldsymbol{\varepsilon u} + \mathbf{y}\end{bmatrix}^T \begin{bmatrix}\alpha \\ \alpha'\end{bmatrix}$$

$$(2.124)$$

$$\text{subject to}\left\{\begin{bmatrix}\boldsymbol{u} \\ -\boldsymbol{u}\end{bmatrix}^T \begin{bmatrix}\alpha \\ \alpha'\end{bmatrix} = 0 \; and \; \alpha_s', \alpha_s \in [0, C]\right.$$

Where, $\boldsymbol{u} = [1\;1\;\cdots 1]^T \in R^L, \mathbf{y} = [y^1\; y^2 \cdots y^L]^{\mathrm{T}} \in R^L$ and

$$\boldsymbol{\Phi}(\mathbf{x})^{\mathrm{T}} = [\emptyset(\mathbf{x}^1) \;\; \emptyset(\mathbf{x}^2) \;\; \cdots \;\; \emptyset(\mathbf{x}^L)] \in R^{m\times L}$$

Equation (2.122) can be written as follows:

$$\boldsymbol{w} = \sum_{s=1}^{L} (\alpha_s - \alpha_s') \, \emptyset(\mathbf{x}^s) \tag{2.125}$$

and the support vector regression function (2.115) is obtained as:

$$\begin{aligned} f(\mathbf{x}) = \boldsymbol{w}^T \emptyset(\mathbf{x}) + b &= \sum_{s=1}^{L} (\alpha_s - \alpha_s') \, \emptyset^{\mathrm{T}}(\mathbf{x}^s) \emptyset(\mathbf{x}) \\ &= \sum_{s=1}^{L} (\alpha_s - \alpha_s') \, \mathrm{K}(\mathbf{x}^s, \mathbf{x}) + b \end{aligned} \tag{2.126}$$

It should be noted that the bias term b in (2.126) can also be provided by choosing the first basis function as unity, i.e. $\emptyset_1(\mathbf{x}) = 1$. However, for a kernel representation where $\emptyset_k(\mathbf{x})$ basis functions are not available, b should be introduced explicitly as done above since there is no guarantee of having a unity basis function embedded in the kernel.

in order to have an affine representation in the feature space defined by $\emptyset_k(\cdot)$'s, one may choose $\emptyset_1(\mathbf{x}) = 1$, so w_1 becomes a bias term.

2.2.2.3 Piecewise affine regression

Piecewise affine interpolation for one dimensional case and canonical piecewise linear interpolation for multi dimensional functions are explained in Section 2.1.4 and, respectively, in Section 2.1.5. When there exists noise and outliers and also large number of data requiring memory allocation and time consumption, the piecewise affine interpolation is not suitable. Therefore, piecewise affine regression (approximation) which is a more appropriate strategy for such cases can be used. In the case of piecewise affine approximation, a given finite set of data is divided up into smaller segments, and then simple linear regression is applied to each segment.

The poi nt w here one segments j oins t he ne xt i s c alled as br eakpoint w here differentiability of the function cannot be ensured, but continuity of the function can be ensured. The location and the number of breakpoints are very crucial in piecewise affine approximation. V arious m ethods ha ve be en pr oposed in s tatistics a nd mathematics, and also in engineering disciplines (Parente, 1999; S eber, 2003; Hudson, 1966; Hudson, 1966; G allant & F uller, 1973; F errari-Trecate, M uselli, Liberati, & M orari, 2001; F errari-Trecate, Muselli, 2001; Konstantinides & Natarajan, 1994; Pittman J. & Murthy, 2000), but the determination of the location and the number of breakpoints is still an open research problem.

Let a set of finite data pairs $\{(x^s, y^s)\}_{s=0}^{L}$ be given and let this domain be divided into $l+1$ segments b y us ing $l < L$ breakpoints s uch a s $\gamma_1 < \gamma_2 < \cdots < \gamma_l$. Corresponding to the $l + 1$ segments which have been labeled consecutively from *"0"* (leftmost segment) through *"l"* (rightmost segment) and *'m_j''* denote the slope of s egment j, p artition th e *x-axis* into $l + 1$ intervals $I_0 \triangleq (-\infty, \gamma_1]$, $I_1 \triangleq (\gamma_1, \gamma_2]$, ….,$I_{l-1} \triangleq (\gamma_{l-1,}\gamma_l]$, and $I_l \triangleq (\gamma_l, \infty)$. A typical continuous PWA function *f(x)* is shown in Figure 2.1.

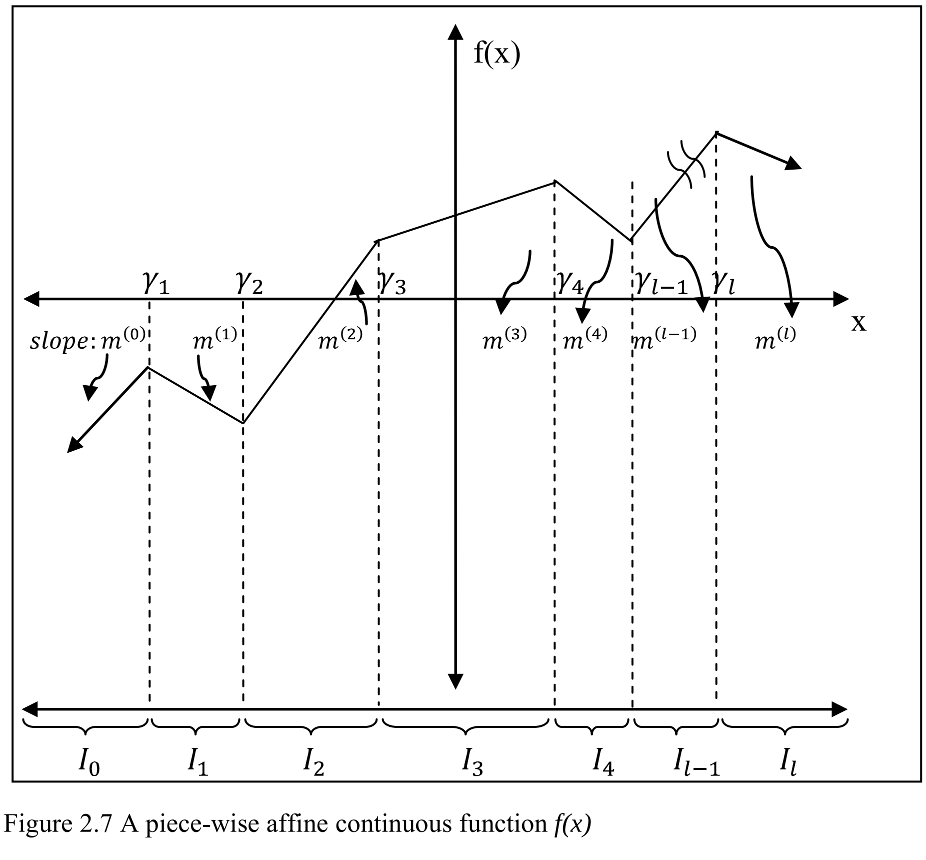


Figure 2.7 A piece-wise affine continuous function *f(x)*

As explained by Theorem 2.1 in Section 2.1.5.1, any single valued continuous PWA with *l* breakpoints $\gamma_1 < \gamma_2 < \cdots < \gamma_l$ can be represented uniquely by the expression:

$$f(x) = a_0 + a_1 x + \sum_{j=1}^{l} b_j \left| x - \gamma_l \right| \tag{2.127}$$

Least square PWA approximation can be formulated as determining $a_0,\ a_1,\ b_j \in R^1$ parameters minimizing the following mean squared error:

$$\min_{\substack{a_0,\, a_1, b_j \in R \\ j \in \{1,2,\cdots l\}}} \sum_{s=1}^{L} \left| a_0 + a_1 x^s + \sum_{j=1}^{l} b_j \left| x^s - \gamma_j \right| - y^s \right|^2 \tag{2.128}$$

The solution to the problem (2.128) is determined by solving the following equation system.

$$\begin{bmatrix} \sum_{s=1}^{L} 1 & \sum_{s=1}^{L} x^s & \sum_{s=1}^{L} |x^s - \gamma_1| & \cdots & \sum_{s=1}^{L} |x^s - \gamma_l| \\ \sum_{s=1}^{L} x^s & \sum_{s=1}^{L} x^s \cdot x^s & \sum_{s=1}^{L} |x^s - \gamma_1| \cdot x^s & \cdots & \sum_{s=1}^{L} |x^s - \gamma_l| \cdot x^s \\ \sum_{=1}^{L} |x^s - \gamma_1| & \sum_{s=1}^{L} x^s \cdot |x^s - \gamma_1| & \sum_{s=1}^{L} |x^s - \gamma_1| \cdot |x^s - \gamma_1| & \cdots & \sum_{s=1}^{L} |x^s - \gamma_l| \cdot |x^s - \gamma_1| \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{s=1}^{L} |x^s - \gamma_l| & \sum_{s=1}^{L} x^s \cdot |x^s - \gamma_l| & \sum_{s=1}^{L} |x^s - \gamma_1| \cdot |x^s - \gamma_l| & \cdots & \sum_{s=1}^{L} |x^s - \gamma_l| \cdot |x^s - \gamma_l| \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ b_l \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{s=1}^{L} y^s \\ \sum_{s=1}^{L} y^s x^s \\ \sum_{s=1}^{L} y^s |x^s - \gamma_1| \\ \vdots \\ \sum_{s=1}^{L} y^s |x^s - \gamma_l| \end{bmatrix} \tag{2.129}$$

2.2.2.4 Piecewise polynomial regression

Piecewise polynomial regression is similar to the piecewise affine regression, but there is an important difference; for each interval between the breakpoints, higher order polynomial such as quadratic and cubic polynomial is applied instead of first order polynomial.

Let a set of finite data pairs $\{(x^s, y^s)\}_{s=0}^{L}$ be given and let this domain be divided into $l+1$ segments by using $l < L$ breakpoints such as $\gamma_1 < \gamma_2 < \cdots < \gamma_l$. Corresponding to the *l + 1* segments which have been labeled consecutively from *"0"* (leftmost segment) through *"l"* (rightmost segment) and *'m_j''* denote the slope of segment j, partition the *x-axis* into *l + 1* intervals $I_0 \triangleq (-\infty, \gamma_1]$, $I_1 \triangleq (\gamma_1, \gamma_2]$, ….,$I_{l-1} \triangleq (\gamma_{l-1,}\gamma_l]$, and $I_l \triangleq (\gamma_l, \infty)$.

A q'<u>th</u> order piecewise-polynomial regression is defined as the concatenation of the following polynomials which are defined for a specific interval.

$$g_j(x, \alpha_j) = \sum_{i}^{q} \alpha_{ij} x^i \,, \quad i \in \{1,2,\cdots,q\} \tag{2.130}$$

Where, α_{ij}'s are coefficients of the polynomial of segment j.

CHAPTER THREE
ROBUST AND LOW COMPLEX SUPPORT VECTOR REGRESSION MODELS

Support Vector Machines (SVMs) were initially developed to solve pattern recognition problems (Vapnik and Lerner, 1963; Vapnik and Chervonenkis 1974; Samanta, Al-Balushi, & Al-Araimi, 2003; Hao, 2008; Mohammadi and Gharehpetian, 2009), then they have been extended to the domain of regression problems so called Support Vector Regression (SVR) (Vapnik et al., 1997; Tao, D. Tang, Li, & Wu, 2005; Goel and Pal, 2009; Osowski and Garanty, 2007; Colliez, Dufrenois, & Hamad, 2006; Vong, Wong, & Li 2006; Bergeron, Cheriet, Ronsky, Zernicke, & Labelle 2005; Huang, Lai, Luo, & Yan, 2005; Wang, Wang, & Lai, 2005; Bao, Liu, Guo, Wang, 2005; Sun and Sun, 2003; Wu, 2009; Lute, Upadhyay, & Singh, 2009; Wu, Yan, & Yang, 2008a,b). In order to extend SVMs into SVRs, Vapnik defined epsilon (ε) insensitive loss function that ignored errors which are within a determined epsilon distance of desired output. The proper choice of ε is critical for generalization. There are many different loss functions other than ε-insensitive such as Huber, quadratic etc. This chapter of the thesis presents novel robust and low complex regression models by introducing new loss functions for rejecting outliers and noises, and ℓ_p with $p \leq 1$ "norms" for model parameters in order to reduce model complexity.

3.1 Support vector nonlinear regression

Let a function $f(\mathbf{x}) = R^{\mathrm{n}} \rightarrow R$ to be approximated be given by a finite set of input-output data pairs $\{(\mathbf{x}^s, y^s)\}_{s=1}^{L}$ with $\mathbf{x}^s \in R^{\mathrm{n}}$ and $y^s \in R$. Let $\emptyset(\cdot): R^n \rightarrow R^m$ be a nonlinear basis function which maps the training data into a high dimensional feature space where the linear regression is performed by support vector algorithm. In this case, the approximated function is.

$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j \emptyset_j(\mathbf{x}) + b = \boldsymbol{w}^T \emptyset(\mathbf{x}) + b \tag{3.1}$$

In the sequel, a quite general SVR formulation introduced in (Smola,1999) will be presented. The formulation is based on a general symmetric and convex loss function given in (3.2) where $e(\cdot): R \to R$ is an arbitrary convex and differentiable function.

$$E(x) = \begin{cases} 0 & for\ |x| < \varepsilon \\ e(|x| - \varepsilon) & otherwise \end{cases} \tag{3.2}$$

As in any SVR formulation, the cost function consists of two terms one of which is the squared Euclidean norm of the $\boldsymbol{w}$ parameter vector and the other is the empirical error:

$$\min_{\boldsymbol{w} \in R^m, b \in R} \frac{1}{2} \parallel \boldsymbol{w} \parallel_2^2 + C \sum_{s=1}^{L} E(y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b) \tag{3.3}$$

Using slack variables $\xi_s, \xi_s' \geq 0$, one can reformulate the above minimization problem as follows.

$$\begin{aligned} & \min_{\boldsymbol{w} \in R^m, b \in R} \frac{1}{2} \parallel \boldsymbol{w} \parallel_2^2 + C \sum_{s=1}^{L} [e(\xi_s) + e(\xi_s')] \\ & \text{subject to} \begin{cases} y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b \leq \varepsilon + \xi_s \\ -y^s + \boldsymbol{w}^T \emptyset(\mathbf{x}^s) + b \leq \varepsilon + \xi_s' \\ \xi_s, \xi_s' \geq 0, \qquad s \in \{1, \dots, L\} \end{cases} \end{aligned} \tag{3.4}$$

This primal optimization problem can be solved by transforming it into a dual form by the method of Lagrange multipliers.

$$\min_{\substack{\boldsymbol{w}\in R^m, b\in R \\ \alpha_s,\alpha_s',\gamma_s,\gamma_s',\xi_s,\xi_s'\geq 0 \\ s\in\{1,\dots,L\}}} J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s')$$

$$= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{s=1}^{L}[e(\xi_s) + e(\xi_s')]]$$

$$- \sum_{s=1}^{L}\alpha_s(\varepsilon + \xi_s - y^s + \boldsymbol{w}^T\emptyset(\mathbf{x}^s) + b) \tag{3.5}$$

$$- \sum_{s=1}^{L}\alpha_s'(\varepsilon + \xi_s' + y^s - \boldsymbol{w}^T\emptyset(\mathbf{x}^s) - b)$$

$$- \sum_{s=1}^{L}(\gamma_s\xi_s + \gamma_s'\xi_s')$$

Where, the nonnegative variables α_s, α_s' and γ_s, γ_s' are Lagrange multipliers. The solution of the problem (3.4) is determined by finding saddle points of Lagrangian function $J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s')$ via partial differentiating it with respect to the primal variables $(\boldsymbol{w}, b, \xi_s, \xi_s')$ and setting the result equal to zero.

Then, one obtains the following equations:

$$\frac{\partial J}{\partial b} = \sum_{s=1}^{L}(\alpha_s' - \alpha_s) = 0 \tag{3.6}$$

$$\nabla_{\boldsymbol{w}} J = \boldsymbol{w} - \sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s) = 0 \tag{3.7}$$

$$\frac{\partial J}{\partial \xi_s} = C\frac{\partial e(\xi_s)}{\partial \xi_s} - \alpha_s - \gamma_s = 0 \tag{3.8}$$

$$\frac{\partial J}{\partial \xi_s'} = C\frac{\partial e(\xi_s')}{\partial \xi_s'} - \alpha_s' - \gamma_s' = 0 \tag{3.9}$$

Substituting (3.7) into (3.5), (3.10) is obtained.

$$J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s')$$

$$= \frac{1}{2}\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s)\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\emptyset(\mathbf{x}^r)\right]$$
$$- \left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s)\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\emptyset(\mathbf{x}^r)\right]$$
$$- \varepsilon\sum_{s=1}^{L}(\alpha_s + \alpha_s') + \sum_{s=1}^{L} y^s(\alpha_s - \alpha_s') + b\sum_{s=1}^{L}(\alpha_s' - \alpha_s) \qquad (3.10)$$
$$+ C\sum_{s=1}^{L}[e(\xi_s) + e(\xi_s'))] - \sum_{s=1}^{L}\alpha_s\,\xi_s - \sum_{s=1}^{L}\alpha_s'\,\xi_s'$$
$$- \sum_{s=1}^{L}(\gamma_s\xi_s + \gamma_s'\xi_s')$$

Now, substituting (3.8) and (3.9) into the (3.10), the cost function takes the following form which is in terms of the Lagrange multipliers.

$$J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \gamma_s, \gamma_s', \xi_s, \xi_s')$$

$$= -\frac{1}{2}\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s)\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\emptyset(\mathbf{x}^r)\right] - \varepsilon\sum_{s=1}^{L}(\alpha_s + \alpha_s')$$
$$+ \sum_{s=1}^{L} y^s(\alpha_s - \alpha_s') + C\sum_{s=1}^{L}[e(\xi_s) + e(\xi_s'))] - C\sum_{s=1}^{L}\xi_s\left[\frac{\partial e(\xi_s)}{\partial \xi_s}\right]$$
$$- C\sum_{s=1}^{L}\xi_s'\left[\frac{\partial E(\xi_s')}{\partial \xi_s'}\right]$$

$$= -\frac{1}{2}\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s)\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\emptyset(\mathbf{x}^r)\right] - \varepsilon\sum_{s=1}^{L}(\alpha_s + \alpha_s')$$
$$+ \sum_{s=1}^{L} y^s(\alpha_s - \alpha_s') + C\sum_{s=1}^{L}[e(\xi_s) + e(\xi_s'))]$$
$$- C\left(\sum_{s=1}^{L}\xi_s\left[\frac{\partial e(\xi_s)}{\partial \xi_s}\right] + \xi_s'\left[\frac{\partial e(\xi_s')}{\partial \xi_s'}\right]\right)$$

$$J(\boldsymbol{w}, b, \alpha_s, \alpha'_s, \gamma_s, \gamma'_s, \xi_s, \xi'_s)$$
$$= -\frac{1}{2}\left[\sum_{s=1}^{L}(\alpha_s - \alpha'_s)\,\emptyset(\mathbf{x}^s)\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha'_r)\,\emptyset(\mathbf{x}^r)\right]$$
$$- \varepsilon\sum_{s=1}^{L}(\alpha_s + \alpha'_s) + \sum_{s=1}^{L} y^s(\alpha_s - \alpha'_s)$$
$$+ C\left[\left(Q(\xi_s)\right) + \left(Q(\xi'_s)\right)\right] \tag{3.11}$$

Where, $\sum_{s=1}^{L}(\alpha'_s - \alpha_s) = 0$ and $\xi = \inf\left\{\xi \,\middle|\, \frac{\partial e(\xi_s)}{\partial \xi_s} \geq \frac{\alpha}{C}\right\}$ and $\xi, \alpha \geq 0$, and $Q(\xi) \coloneqq e(\xi) - \xi\frac{\partial e(\xi_s)}{\partial \xi_s}$.

Special choices of $e(\xi)$ is given below.

ε-insensitive loss:

This case is the original SVR case as described in Section 2.2.2.2. Since $e(\xi) = \xi$, then $Q(\xi) = 0$, or equivalently, $Q(\alpha) = 0$ and $\alpha_s, \alpha'_s \in [0, C]$

Polynomial loss:

Quadratic loss function is another special case for this loss function class.

$$e(\xi) = \frac{1}{p}\xi^p \quad \text{with } p > 1 \tag{3.12}$$

which yields

$$Q(\xi) = \frac{1-p}{p}\xi^p \tag{3.13}$$

By substituting α for ξ, one gets

$$Q(\alpha) = \frac{1-p}{p}\left(\frac{\alpha}{C}\right)^{\frac{p}{p-1}} \text{ and } \alpha \in [0, \infty] \tag{3.14}$$

Piecewise polynomial and linear loss:

For some $\mu \in R$,

$$e(\xi) = \begin{cases} \mu^{1-p}\frac{1}{p}\xi^{p} & for\ \xi < \mu \\ \xi + \left(\frac{1}{p} - 1\right)\mu & for\ \xi \geq \mu \end{cases} \tag{3.15}$$

This loss function is a special Huber loss function and is known as robust against outliers in the literature. Similar calculations yield

$$Q(\xi) = \frac{1-p}{p}\xi^{p} \tag{3.16}$$

which implies

$$Q(\xi) = \frac{1-p}{p}\mu^{1-p}\left(\frac{\alpha}{C}\right)^{\frac{p}{p-1}} \text{and } \alpha_s, \alpha_s' \in [0, C] \tag{3.17}$$

3.2 Novel robust and low complex regression models

A general SVR formulation in terms of symmetric and convex loss function is presented in the previous section. These loss functions may suffer from poor generalization ability when presence of outliers.

This section presents novel robust and low complex regression models by introducing new loss functions for rejecting outliers and noises, and ℓ_p with $p \leq 1$ "norms" for model parameters in order to reduce model complexity. The introduced model class is described by the following optimization formulation.

$$\min_{\boldsymbol{w} \in R^m, b \in R} \|\boldsymbol{w}\|_{p,\varepsilon_2}^{p} + C\sum_{s=1}^{L} \|y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b\|_{q,\varepsilon_1}^{q} \tag{3.18}$$

Where, the first term in the cost forces the primal parameters (i.e. weights) to be sparse and the second forces the errors so the dual parameters (i.e. Lagrange multipliers) to be sparse. p and q might be a real number in the interval of $[0, \infty)$.

For p, q belonging to $[0,1)$, the functions $\|\cdot\|_p^p\colon R^m \to R^+$, $\|\cdot\|_q^q\colon R \to R^+$ do not define a norm. The positive homogeneity condition of norm is violated for $p, q = 0$ while the triangle inequality condition is not satisfied for $0 < p, q < 1$. On the other hand, the epsilon insensitive versions $\|\cdot\|_{p,\varepsilon_2}^p$, $\|\cdot\|_{q,\varepsilon_1}^q$ define semi-norms for $\varepsilon_1, \varepsilon_2 \neq 0$ since they can be zero for the points inside the epsilon tubes.

The case of $p = q = 1$:

$$\min_{\boldsymbol{w}\in R^m, b\in R} \| \boldsymbol{w} \|_{1,\varepsilon_2} + C \sum_{s=1}^{L} |y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b|_{\varepsilon_1} \tag{3.19}$$

Using slack variables $\xi_s, \xi'_s, \eta, \eta' \geq 0$, one can reformulate the above minimization problem as follows.

$$\min \sum_{i=1}^{m} (\eta_i + \eta'_i) + C \sum_{s=1}^{L} (\xi_s + \xi'_s)$$
$$\text{subject to} \begin{cases} y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b \leq \varepsilon_1 + \xi_s \\ -y^s + \boldsymbol{w}^T \emptyset(\mathbf{x}^s) + b \leq \varepsilon_1 + \xi'_s \\ w_i \leq \varepsilon_2 + \eta_i \\ -w_i \leq \varepsilon_2 + \eta'_i \\ \eta_i, \eta'_i, \xi_s, \xi'_s \geq 0 \\ s \in \{1, \dots, L\} \text{ and } i \in \{1, \dots, m\} \end{cases} \tag{3.20}$$

This primal optimization problem can be solved by transforming it into a dual form using the method of Lagrange multipliers.

$$\min_{\substack{\boldsymbol{w}\in R^m, b\in R \\ \alpha_s,\alpha_s',\gamma_s,\gamma_s',\beta_i,\beta_i',\delta_i,\delta_i',\xi_s,\xi_s',\eta_i\eta_i'\geq 0 \\ s\in\{1,\dots,L\} \text{ and } i\in\{1,\dots,m\}}} J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \beta_i, \beta_i', \gamma_s, \gamma_s', \delta_i, \delta_i', \xi_s, \xi_s', \eta_i, \eta_i')$$

$$\begin{aligned} &= \sum_{i=1}^{m}(\eta_i + \eta_i') + C\sum_{s=1}^{L}(\xi_s + \xi_s') \\ &- \sum_{s=1}^{L}\alpha_s(\varepsilon_1 + \xi_s - y^s + \boldsymbol{w}^T\emptyset(\mathbf{x}^s) + b) \\ &- \sum_{s=1}^{L}\alpha_s'(\varepsilon_1 + \xi_s' + y^s - \boldsymbol{w}^T\emptyset(\mathbf{x}^s) - b) \\ &- \sum_{i=1}^{m}\beta_i(\varepsilon_2 + \eta_i - w_i) - \sum_{i=1}^{m}\beta_i'(\varepsilon_2 + \eta_i' + w_i) \\ &- \sum_{s=1}^{L}(\gamma_s\xi_s + \gamma_s'\xi_s') - \sum_{i=1}^{m}(\delta_i\eta_i + \delta_i'\eta_i') \end{aligned} \quad (3.21)$$

Where, the nonnegative variables $\alpha_s, \alpha_s', \gamma_s, \gamma_s', \delta_i, \delta_i'$ and β_i, β_i' are Lagrange multipliers. The solution of the problem (3.20) is determined by partial differentiating it with respect to the primal variables $(\boldsymbol{w}, b, \xi_s, \xi_s', \eta_i, \eta_i')$ and setting the results equal to zero.

$$\frac{\partial J}{\partial b} = \sum_{s=1}^{L}(\alpha_s' - \alpha_s) = 0 \quad (3.22)$$

$$\nabla_{\boldsymbol{w}} J = \sum_{i=1}^{m}(\beta_i - \beta_i') - \sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s) = 0 \quad (3.23)$$

$$\frac{\partial J}{\partial \xi_s} = C - \alpha_s - \gamma_s = 0 \quad (3.24)$$

$$\frac{\partial J}{\partial \xi_s'} = C - \alpha_s' - \gamma_s' = 0 \quad (3.25)$$

$$\frac{\partial J}{\partial \eta_i} = 1 - \beta_i - \delta_i = 0 \quad (3.26)$$

$$\frac{\partial J}{\partial \eta_i'} = 1 - \beta_i' - \delta_i' = 0 \quad (3.27)$$

The above necessary conditions do not provide to solve $\boldsymbol{w} \in R^m$ in terms of the Lagrange multipliers in order to obtain a cost which is in terms of Lagrange multipliers only, thus a dual formulation cannot be obtained. So, the problem defined in (3.19) can be solved by solving the linear programming problem in (3.20) using simplex method or interior point method or any other numerical method developed for solving linear programming problems (Margaret, 1998).

In order to reach a kernel representation associated with a quadratic cost minimization in terms of Lagrange multipliers, this thesis proposes to augment the cost in (3.20) by adding a squared Euclidean norm $\frac{P}{2} \parallel \boldsymbol{w} \parallel_2^2$ with a small $P \in R^+$ and then derive the standard SVR formulation for the newly proposed formulation as done in the following.

$$\min_{\boldsymbol{w}\in R^m, b\in R} \frac{P}{2} \parallel \boldsymbol{w} \parallel_2^2 + \parallel \boldsymbol{w} \parallel_{1,\varepsilon} + C \sum_{s=1}^{L} |y^s - (\boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b)|_\varepsilon \tag{3.28}$$

This optimization problem can be reformulated by introducing slack variables $\xi_s, \xi_s', \eta, \eta' \geq 0$ as follows.

$$\min_{\boldsymbol{w}\in R^m, b\in R} \frac{P}{2} \parallel \boldsymbol{w} \parallel_2^2 + \sum_{i=1}^{m} (\eta_i + \eta_i') + C \sum_{s=1}^{L} (\xi_s + \xi_s')$$
$$\text{subject to} \begin{cases} y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b \leq \varepsilon_1 + \xi_s \\ -y^s + \boldsymbol{w}^T \emptyset(\mathbf{x}^s) + b \leq \varepsilon_1 + \xi_s' \\ w_i \leq \varepsilon_2 + \eta_i \\ -w_i \leq \varepsilon_2 + \eta_i' \\ \eta_i, \eta_i', \xi_s, \xi_s' \geq 0 \\ s \in \{1, \dots, L\} \text{ and } i \in \{1, \dots, m\} \end{cases} \tag{3.29}$$

Minimization problem in (3.29) can be transformed into an unconstrained problem by the method of Lagrange multipliers.

$$\min_{\substack{\boldsymbol{w}\in R^m, b\in R \\ \alpha_s,\alpha_s',\gamma_s,\gamma_s',\beta_i,\beta_i',\delta_i,\delta_i',\xi_s,\xi_s',\eta_i\eta_i'\geq 0 \\ s\in\{1,\ldots,L\} \text{ and } i\in\{1,\ldots,m\}}} J(\boldsymbol{w}, b, \alpha_s, \alpha_s', \beta_i, \beta_i', \gamma_s, \gamma_s', \delta_i, \delta_i', \xi_s, \xi_s', \eta_i, \eta_i')$$

$$\begin{aligned} &= \frac{P}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_{i=1}^{m}(\eta_i + \eta_i') + C\sum_{s=1}^{L}(\xi_s + \xi_s') \\ &- \sum_{s=1}^{L}\alpha_s(\varepsilon_1 + \xi_s - y^s + \boldsymbol{w}^T\emptyset(\mathbf{x}^s) + b) \\ &- \sum_{s=1}^{L}\alpha_s'(\varepsilon_1 + \xi_s' + y^s - \boldsymbol{w}^T\emptyset(\mathbf{x}^s) - b) \\ &- \sum_{i=1}^{m}\beta_i(\varepsilon_2 + \eta_i - w_i) - \sum_{i=1}^{m}\beta_i'(\varepsilon_2 + \eta_i' + w_i) \\ &- \sum_{s=1}^{L}(\gamma_s\xi_s + \gamma_s'\xi_s') - \sum_{i=1}^{m}(\delta_i\eta_i + \delta_i'\eta_i') \end{aligned} \tag{3.30}$$

Where, the nonnegative variables $\alpha_s, \alpha_s', \gamma_s, \gamma_s', \delta_i, \delta_i'$ and β_i, β_i' are Lagrange multipliers. The solution of the problem (3.30) is determined by partial differentiating it with respect to the primal variables $(\boldsymbol{w}, b, \xi_s, \xi_s', \eta_i, \eta_i')$ and setting the results equal to zero.

$$\frac{\partial J}{\partial b} = \sum_{s=1}^{L}(\alpha_s' - \alpha_s) = 0 \tag{3.31}$$

$$\nabla_{\boldsymbol{w}} J = P \cdot \boldsymbol{w} - \sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s) + \sum_{i=1}^{m}(\beta_i - \beta_i')\,e_i = 0 \tag{3.32}$$

$$\frac{\partial J}{\partial \xi_s} = C - \alpha_s - \gamma_s = 0 \tag{3.33}$$

$$\frac{\partial J}{\partial \xi_s'} = C - \alpha_s' - \gamma_s' = 0 \tag{3.34}$$

$$\frac{\partial J}{\partial \eta_i} = 1 - \beta_i - \delta_i = 0 \tag{3.35}$$

$$\frac{\partial J}{\partial \eta_i'} = 1 - \beta_i' - \delta_i' = 0 \tag{3.36}$$

Where, $e_i = [0 \cdots 0 \;\; 1 \;\; 0 \cdots 0]^T \in R^m$ is the unit vector whose i'th component is unity while the others are all zero. $\varepsilon_1, \varepsilon_2,$ and C are parameters defined by the user. Now, substituting (3.32) into (3.30), one obtains (3.37).

$$
\begin{aligned}
& J(\alpha_s, \alpha_s', \beta_i, \beta_i', \gamma_s, \gamma_s', \delta_i, \delta_i', \xi_s, \xi_s', \eta_i, \eta_i') \\
&\quad = \frac{P}{2}\left[\frac{1}{P}\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s) - \frac{1}{P}\sum_{i=1}^{m}(\beta_i - \beta_i')\,e_i\right]^T \left[\frac{1}{P}\sum_{r=1}^{L}(\alpha_r \right.\\
&\quad \left. - \alpha_r')\,\emptyset(\mathbf{x}^r) - \frac{1}{P}\sum_{j=1}^{m}(\beta_j - \beta_j')\,e_j\right] \\
&\quad - \sum_{s=1}^{L}\alpha_s\left[\frac{1}{P}\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\emptyset(\mathbf{x}^r) - \frac{1}{P}\sum_{i=1}^{m}(\beta_i - \beta_i')\,e_i\right]^T \emptyset(\mathbf{x}^s) \\
&\quad + \sum_{s=1}^{L}\alpha_s'\left[\frac{1}{P}\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\emptyset(\mathbf{x}^r) - \frac{1}{P}\sum_{j=1}^{m}(\beta_j - \beta_j')\,e_j\right]^T \emptyset(\mathbf{x}^s) \\
&\quad - \varepsilon_1\sum_{s=1}^{L}(\alpha_s + \alpha_s') - \sum_{s=1}^{L}\alpha_s\,\xi_s - \sum_{s=1}^{L}\alpha_s'\,\xi_s' + \sum_{s=1}^{L}y^s(\alpha_s - \alpha_s') \\
&\quad + b\sum_{s=1}^{L}(\alpha_s' - \alpha_s) + K\sum_{i=1}^{m}(\eta_i + \eta_i') + C\sum_{s=1}^{L}(\xi_s + \xi_s') \\
&\quad - \sum_{i=1}^{m}\beta_i\left(\varepsilon_2 + \eta_i - \frac{1}{P}\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s) - \sum_{i=1}^{m}(\beta_i - \beta_i')\,e_i\right]\right) \\
&\quad - \sum_{i=1}^{m}\beta_i'\left(\varepsilon_2 + \eta_i' + \frac{1}{P}\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s) + \sum_{i=1}^{m}(\beta_i - \beta_i')\,e_i\right]\right) \\
&\quad - \sum_{s=1}^{L}(\gamma_s\xi_s + \gamma_s'\xi_s') - \sum_{i=1}^{m}(\delta_i\eta_i + \delta_i'\eta_i')
\end{aligned}
\tag{3.37}
$$

Then, substituting (3.31), (3.33), (3.34), (3.35) and (3.36) into (3.37), one can obtain (3.38).

$$
\begin{aligned}
&J(\alpha_s, \alpha_s', \beta_i, \beta_i', \gamma_s, \gamma_s', \delta_i, \delta_i', \xi_s, \xi_s', \eta_i, \eta_i') \\
&\qquad = \frac{1}{2P}\left[\sum_{s=1}^{L}\sum_{r=1}^{L}(\alpha_s - \alpha_s')(\alpha_r - \alpha_r')\, \emptyset^{\boldsymbol{T}}(\mathbf{x}^s)\emptyset(\mathbf{x}^r)\right. \\
&\qquad - \sum_{i=1}^{m}\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,(\beta_i - \beta_i')\, \emptyset^{\boldsymbol{T}}(\mathbf{x}^s)e_i \\
&\qquad - \sum_{i=1}^{m}\sum_{r=1}^{L}(\alpha_s - \alpha_s')\,(\beta_i - \beta_i')\, {e_i}^T\emptyset(\mathbf{x}^s) \\
&\qquad \left. + \sum_{j=1}^{m}\sum_{i=1}^{m}(\beta_i - \beta_i')(\beta_j - \beta_j')\, {e_i}^T e_j\right] \\
&\qquad - \frac{1}{P}\sum_{s=1}^{L}\sum_{r=1}^{L}(\alpha_s - \alpha_s')(\alpha_r - \alpha_r')\, \emptyset^{\boldsymbol{T}}(\mathbf{x}^s)\emptyset(\mathbf{x}^r) \\
&\qquad + \frac{1}{P}\sum_{i=1}^{m}\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,(\beta_i - \beta_i')\, {e_i}^T\emptyset(\mathbf{x}^s) - \varepsilon_1\sum_{s=1}^{L}(\alpha_s + \alpha_s') \\
&\qquad + \sum_{s=1}^{L} y^s(\alpha_s - \alpha_s') + b\sum_{s=1}^{L}(\alpha_s' - \alpha_s) + \sum_{i=1}^{m}(\eta_i + \eta_i') \\
&\qquad + C\sum_{s=1}^{L}(\xi_s + \xi_s') - \sum_{s=1}^{L}\xi_s(\alpha_s + \gamma_s) - \sum_{s=1}^{L}\xi_s'(\alpha_s' + \gamma_s') \\
&\qquad - \sum_{i=1}^{m}(\beta_i + \delta_i)\eta_i - \sum_{i=1}^{m}(\beta_i' + \delta_i')\eta_i' - \sum_{i=1}^{m}(\beta_i + \beta_i')\varepsilon_2 \\
&\qquad - \frac{1}{P}\sum_{i=1}^{m}\sum_{s=1}^{L}(\alpha_s - \alpha_s')\, \emptyset_{\boldsymbol{i}}(\mathbf{x}^s)(\beta_i - \beta_i') \\
&\qquad + \frac{1}{P}\left[\sum_{i=1}^{m}\sum_{j=1}^{m}(\beta_i - \beta_i')\,(\beta_j - \beta_j')\right]
\end{aligned}
\tag{3.38}
$$

Finally, after performing several simplifications, the following dual optimization formulation is obtained for the introduced low complex SVR.

$$\max_{\substack{\sum_{s=1}^{L}(\alpha_s'-\alpha_s)=0 \\ \alpha_s',\alpha_s\in[0,C] and\ \beta_i,\beta_i'\in[0,K]}} \quad J(\alpha_s,\alpha_s',\beta_i,\beta_i')$$
$$= -\frac{1}{2P}\sum_{s=1}^{L}\sum_{r=1}^{L}(\alpha_s-\alpha_s')(\alpha_r-\alpha_r')\,\emptyset^{\boldsymbol{T}}(\mathbf{x}^s)\emptyset(\mathbf{x}^r)$$
$$-\varepsilon_1\sum_{s=1}^{L}(\alpha_s+\alpha_s')+\sum_{s=1}^{L}y^s(\alpha_s-\alpha_s')-\varepsilon_2\sum_{i=1}^{m}(\beta_i+\beta_i') \tag{3.39}$$
$$-\frac{1}{P}\sum_{i=1}^{m}\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,\emptyset_{\boldsymbol{i}}(\mathbf{x}^s)(\beta_i-\beta_i')$$
$$+\frac{1}{P}\sum_{i=1}^{m}\sum_{j=1}^{m}(\beta_i-\beta_i')\,(\beta_j-\beta_j')$$

which is equivalent to

$$\min_{\substack{\sum_{s=1}^{L}(\alpha_s'-\alpha_s)=0 \\ \alpha_s',\alpha_s\in[0,C] and\ \beta_i,\beta_i'\in[0,K]}} \quad J(\alpha_s,\alpha_s',\beta_i,\beta_i')$$
$$= \Bigg[\frac{1}{2P}\sum_{s=1}^{L}\sum_{r=1}^{L}(\alpha_s-\alpha_s')(\alpha_r-\alpha_r')\,\emptyset^{\boldsymbol{T}}(\mathbf{x}^s)\emptyset(\mathbf{x}^r)$$
$$+\varepsilon_1\sum_{s=1}^{L}(\alpha_s+\alpha_s')-\sum_{s=1}^{L}y^s(\alpha_s-\alpha_s')+\varepsilon_2\sum_{i=1}^{m}(\beta_i+\beta_i') \tag{3.40}$$
$$+\frac{1}{P}\sum_{i=1}^{m}\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,\emptyset_{\boldsymbol{i}}(\mathbf{x}^s)(\beta_i-\beta_i')$$
$$-\frac{1}{P}\sum_{i=1}^{m}\sum_{j=1}^{m}(\beta_i-\beta_i')\,(\beta_j-\beta_j')\Bigg]$$

For small P values, the terms in the second line of (3.40) are dominated by the other three terms, so they can be omitted. The following truncated quadratic minimization problem is obtained as an equivalent problem in (3.19). It means that SVR with ε-insensitive loss function and ℓ_1 norm for model parameters whose associated minimization problem described by (3.19) can be designed by solving the truncated dual SVR problem in (3.41).

$$\min_{\substack{\sum_{s=1}^{L}(\alpha_s'-\alpha_s)=0 \\ \alpha_s',\alpha_s\in[0,C] and \ \beta_i,\beta_i'\in[0,K]}} J_T(\alpha_s,\alpha_s',\beta_i,\beta_i')$$

$$= \frac{1}{2}\sum_{s=1}^{L}\sum_{r=1}^{L}(\alpha_s-\alpha_s')(\alpha_r-\alpha_r')\,\emptyset^{\boldsymbol{T}}(\mathbf{x}^s)\emptyset(\mathbf{x}^r)$$
$$+\sum_{i=1}^{m}\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,\emptyset_{\boldsymbol{i}}(\mathbf{x}^s)(\beta_i-\beta_i')$$
$$-\sum_{i=1}^{m}\sum_{j=1}^{m}(\beta_i-\beta_i')\,(\beta_j-\beta_j') \tag{3.41}$$

Solving (3.40) or (3.41), one can determine Lagrange multipliers, and then obtain the optimum $\boldsymbol{w}$ weight vector as:

$$\boldsymbol{w} = \frac{1}{P}\left[\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,\emptyset(\mathbf{x}^s) - \sum_{i=1}^{m}(\beta_i-\beta_i')\,e_i\right] \tag{3.42}$$

This yields the following regression function described in terms of the Lagrange multipliers.

$$f(\mathbf{x}) = \frac{1}{P}\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,\emptyset^{\mathrm{T}}(\mathbf{x}^s)\emptyset(\mathbf{x}) - \frac{1}{P}\sum_{i=1}^{m}(\beta_i-\beta_i')\,\emptyset_i(\mathbf{x}) \tag{3.43}$$

Although the first term in (3.43) can be represented in terms of the kernel function $\mathrm{K}(\mathbf{x}^s,\mathbf{x}) = \emptyset^{\mathrm{T}}(\mathbf{x}^s)\cdot\emptyset(\mathbf{x})$ without knowing $\emptyset(\cdot)\colon R^n \to R^m$ function itself, the second term requires knowing $\emptyset(\cdot)$. However, if the samples $\mathbf{x}^{i}$'s exist such that $\emptyset(\mathbf{x}^i) = e_i$, then $\emptyset_i(\mathbf{x})$ can be calculated as $\emptyset_i(\mathbf{x}) = \mathrm{K}(\mathbf{x}^i,\mathbf{x}) = \emptyset^{\mathrm{T}}(\mathbf{x}^i)\cdot\emptyset(\mathbf{x})$ so, (3.43) can have the following kernel representation.

$$f(\mathbf{x}) = \frac{1}{P}\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,\mathrm{K}(\mathbf{x}^s,\mathbf{x}) - \frac{1}{P}\sum_{i=1}^{m}(\beta_i-\beta_i')\,\mathrm{K}(\mathbf{x}^i,\mathbf{x}) \tag{3.44}$$

Considering (3.28) with ℓ_1 norm for $\boldsymbol{w}$ parameter, one deduce that minimizations of (3.40) or (3.41) produce sparse primal parameter space representations which are convenient when $\emptyset(\cdot): R^n \to R^m$ transformation is available and when $\emptyset(\mathbf{x}^i) = e_i$ with $i \in \{1,2,\cdots,m\}$ are known.

***Inverse-Gaussian function as primal parameter flatness*:**

The ε-insensitive ℓ_1 norm in (3.19) and (3.28) formulations may be replaced with ℓ_0 "norm" to obtain more sparseness in the primal parameter (weight) space. However, the minimization problems become now NP-hard, so computationally intractable. This thesis proposes to employ inverse-Gaussian function to approximate to ℓ_0 "norm". Then, the following optimization formulations are developed for low complex regression models.

$$\min_{\boldsymbol{w}\in\boldsymbol{R}^m, b\in R} \sum_{i=1}^{m}\left[1 - e^{-\frac{w_i^2}{\varepsilon_2^2}}\right] + C\sum_{s=1}^{L}\left|y^s - \boldsymbol{w}^T\emptyset(\mathbf{x}^s) - b\right|_{\varepsilon_1} \tag{3.45}$$

Or,

$$\min_{\boldsymbol{w}\in\boldsymbol{R}^m, b\in R} \frac{P}{2}\parallel \boldsymbol{w} \parallel_2^2 + \sum_{i=1}^{m}\left[1 - e^{-\frac{w_i^2}{\varepsilon_2^2}}\right] + C\sum_{s=1}^{L}\left|y^s - \boldsymbol{w}^T\emptyset(\mathbf{x}^s) - b\right|_{\varepsilon_1} \tag{3.46}$$

It should be noted that the inverse-Gaussian function is not a convex function, so are the cost functions in (3.45) and (3.46). However, the inverse-Gaussian has a unique minimum point and no other extremum, so the minimization methods may be applied to find minimum points of (3.45) and (3.46) in some efficient ways.

If the kernel $\mathrm{K}(\mathbf{x}^s, \mathbf{x})$ is available without knowing $\emptyset(\cdot)$ explicitly, then one would try to use optimization formulations which provide sparseness in the dual parameter space, i.e. in the Lagrange multipliers. In the sequel, two different regression models will be proposed in this direction. The first one exploits flatness term in the dual parameter space. The second employs a loss function rejects the contribution of the outliers.

Flatness in the dual parameter space:

Flatness of the function$f(\mathbf{x}) = \boldsymbol{w}^T\mathbf{x} + b$, can be ensured by minimizing the squared Euclidean norm $\| \boldsymbol{w} \|_2^2$ in primal space.

$$\min_{\boldsymbol{w}\in\boldsymbol{R}^m, b\in R} \frac{1}{2} \| \boldsymbol{w} \|_2^2 + C\sum_{s=1}^{L} |y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b|_\varepsilon$$
$$\text{subject to} \begin{cases} y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b \le \varepsilon + \xi_s \\ -y^s + \boldsymbol{w}^T \emptyset(\mathbf{x}^s) + b \le \varepsilon + \xi_s' \\ \xi_s, \xi_s' \ge 0, \qquad s \in \{1, \dots, L\} \end{cases} \tag{3.47}$$

This primal optimization problem in (3.47) can be solved by transforming it into the following dual form by the method of Lagrange multipliers as mentioned in the Subsection (2.2.2.2).

$$\begin{aligned} J(\alpha_s, \alpha_s') = +\frac{1}{2}\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s)\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\emptyset(\mathbf{x}^r)\right] \\ + \varepsilon\sum_{s=1}^{L}(\alpha_s + \alpha_s') - \sum_{s=1}^{L} y^s(\alpha_s - \alpha_s') \end{aligned} \tag{3.48}$$

Flatness parameter term $\| \boldsymbol{w} \|_2^2$ in the primal parameter space provides sparseness in the primal parameter space but it may not cause sparseness in the dual parameter space. For a kernel representation, the sparseness in the dual parameter space is the feature that is desired. For this purpose the ε-insensitive ℓ_1 semi-norm $|\alpha_s|_\varepsilon$ for the dual parameters are introduced in the thesis. So, (3.48) is augmented by using this proposed flatness term in the dual space as follows.

$$\begin{aligned} J(\alpha_s, \alpha_{s'}',) = \frac{1}{2}\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\emptyset(\mathbf{x}^s)\right]^T \left[\sum_{r=1}^{L}(\alpha_r - \alpha_r')\,\emptyset(\mathbf{x}^r)\right] \\ + \varepsilon\sum_{s=1}^{L}(\alpha_s + \alpha_s') - \sum_{s=1}^{L} y^s(\alpha_s - \alpha_s') \\ + \delta\sum_{s=1}^{L}[|\alpha_s|_\varepsilon + |\alpha_s'|_\varepsilon] \end{aligned} \tag{3.49}$$

***$lncosh_{\varepsilon}(\cdot)$* loss function for robustness:**

ε-insensitive (absolute value) loss function $|e|_{\varepsilon}$ has useful properties as follows. It ignores small errors so provides a filtering on the noise around the optimal regression function. On the other hand, it will be cleared by the qualitative analysis, summarized in the Table 4.1 at the end of the Chapter 4, that $|e|_{\varepsilon}$ loss function does not reject outliers even very far from the optimal regression function, however it limits their undesirable effects on the parameters of the regression function. This is a consequence of the fact that the Lagrange multipliers corresponding to the data outside of the epsilon tube are all the same, more precisely either C or –C, so their undesirable contributions to the kernel representation does not increase as the distance of the sample away from the optimal regression function increases. $|e|_{\varepsilon}$ has another useful property of being convex, allowing the application of convex optimization methods in the design of regression functions.

In short, finding a loss function which has the properties of rejecting and/or limiting the bad effects of noise and outliers like $|e|_{\varepsilon}$ and of computationally tractable. It will be shown below that $|e|_{\varepsilon}$ can well be approximated by a continuously differentiable function defined as the composition of $\cosh(\cdot)$ and $\ln(\cdot)$ functions. The introduced approximation is still a convex function enabling to exploit the optimization methods which are developed for convex and differentiable costs.

First observe that, for large values of β parameters, the following function defined in terms of cosh $(\cdot)$ and ln $(\cdot)$ functions approaches to $|e|$.

$$\frac{1}{\beta}\text{lncosh}[\beta e] =: \frac{1}{\beta}\ln\,(\cosh[\beta e]) \tag{3.50}$$

Where, $\cosh[\beta e] = \frac{e^{\beta e}+e^{-\beta e}}{2}$. When, β goes to infinity the $\frac{1}{\beta}\text{lncosh}[\beta e]$ tends to $|e|$ and the derivative of $\frac{1}{\beta}\text{lncosh}[\beta e]$ with respect to e tends to $sign(e)$. For small values of β parameters, $\frac{1}{\beta}\text{lncosh}[\beta e]$function becomes similar to Huber function. For small e values, $\frac{1}{\beta}\text{lncosh}[\beta e]$ with a small β behaves like a square function and, for large e values, $\frac{1}{\beta}\text{lncosh}[\beta e]$ with a small β approaches to $|e|$. Further observe that $\frac{1}{\beta}\text{lncosh}[\beta e])$ function is a convex function for all β values. So, the continuously differentiable and convex function $\frac{1}{\beta}\text{lncosh}[\beta e]$ with moderate β values can be used well as a loss function which limits bad effects of outliers and allows exploiting efficient optimization methods requiring differentiability and convexity. As in the ε-insensitive loss function $|e|_\varepsilon$, the following ε-insensitive version of it might be preferable for suppressing the bad effect of noise around the optimal regression function.

$$\frac{1}{\beta}\text{lncosh}_\varepsilon[\beta e] \tag{3.51}$$

Now, considering the fact that $\cosh_\varepsilon[\beta e] = \frac{e^{\beta e}+e^{-\beta e}}{2}$, the function $\frac{1}{\beta}\text{lncosh}[\beta e]$ can be seen to be approximated by the function in (3.52), as ignoring higher order terms in the Taylor series expansion of $\cosh[\beta e]$.

$$\frac{1}{\beta}\ln\left(1+\frac{1}{2}[\beta e]^2\right) \tag{3.52}$$

The function in (3.52) is not a convex function but it has a unique minimum point with no other extremum. It should be noted that, for $\beta = 1$, (3.52) corresponds to a loss function whose minimization provides a regression which is optimal for Cauchy distribution. This thesis proposes the following ε-insensitive version of (3.52) as a saturating loss function for rejecting outliers while preserving the above mentioned useful properties of the ε-insensitive loss function $|e|_\varepsilon$.

$$\frac{1}{\beta}\ln\left(1+\frac{1}{2}[\beta e]_\varepsilon^2\right) \tag{3.53}$$

3.3 A quantitative analysis of the developed robust and low complex regression models

This subsection presents a numerical study for a comparison of the performances of inverse Gaussian norm based flat regression models and the models with $\frac{1}{\beta}\mathrm{lncosh}_{\varepsilon}[\beta e]$ and $\frac{1}{\beta}\ln\left(1+\frac{1}{2}[\beta e]_{\varepsilon}^{2}\right)$ loss functions.

Dual optimization formulations of a part of the developed regression models are failed to be obtained. On the other hand, a part of the obtained dual optimization formulations is not so suitable to be minimized in an efficient way. For the mentioned reasons, a quantitative comparison is given below based on the results obtained by minimizing the primal optimization problems. Since all of the primal optimization problems are constrained optimization where the costs are nonlinear and the constraints are linear inequalities, then the so called ellipsoid algorithm is chosen for finding a solution for each formulation.

To compare the conventional SVR and the developed robust and low complex SVR models, the codes for ellipsoid method was written in Matlab 7.5. Two test functions are considered in the simulations. Outliers are added artificially. To measure the performance of the models (i.e. training and test errors, flatness and computational cost) Percentage of Root mean square Difference (PRD), Root Mean Square Error (RMSE), norm of $\boldsymbol{w}$ and CPU time are used. The results of various cases with different cost functions given in (3.45), (3.46), (3.51), (3.53), conventional SVR with $C = 10$ and with different $\varepsilon_1 = 0$, $\varepsilon_1 = 0.01$ and $\varepsilon_1 = 0.1$ are presented in Table 3.1 and in Table 3.2.

Table 3.1 presents result for the affine test function given below.

$$f(x) = 5x - 5 \;\; with\; x \in [-10,10]$$

101 training patterns are generated for the affine function. Three artificial outliers are created. Linear SVR is chosen for approximating to the given affine function.

Table 3.1 A comparison of ε-insensitive linear SVR and the developed robust & low complex SVR models for the affine function $f(x) = 5x - 5$ with $\varepsilon_2 = 1$, $C = 10$.

$f(x) = 5x - 5$		Percentage of Root mean square Difference (PRD)			Root Mean Square Error (RMSE)			Norm of w			CPU time (In seconds)		
	Epsilon	0	0.01	0.1	0	0.01	0.1	0	0.01	0.1	0	0.01	0.1
SVR Models													
$1 - e^{-\frac{w^2}{\varepsilon_2^2}}$	Train	36.200	37.521	36.762	11.136	11.134	10.675	5	5	5	0.344	0.344	0.328
	Test	39.785	37.628	37.876	11.084	11.082	11.709						
$Pw^2 + \left(1 - e^{-\frac{w^2}{\varepsilon_2^2}}\right)$	Train	35.311	35.423	36.825	10.693	11.134	10.675	5	5	5	0.359	0.328	0.344
	Test	40.130	40.398	37.449	11.731	11.082	11.709						
$\frac{1}{\beta}\ln\cosh_{\varepsilon}(\beta e)$	Train	36.669	37.016	36.419	11.133	10.688	10.663	4.999	4.999	4.999	0.344	0.344	0.344
	Test	38.544	37.318	37.989	11.081	11.725	11.706						
$\frac{1}{\beta}\ln\left(1 + \frac{1}{2}[\beta e]_{\varepsilon}^{2}\right)$	Train	35.754	34.200	36.527	10.693	10.692	11.132	5	4.999	4.989	0.391	0.422	0.391
	Test	39.357	42.396	38.236	11.731	11.729	11.072						
ε-insensitive linear SVR	Train	35.959	35.518	35.282	11.063	9.354	11.077	4.844	4.831	4.848	0.344	0.328	0.344
	Test	39.533	38.017	39.767	10.881	13.175	10.893						

Table 3.2 presents results for the following nonlinear benchmark function.

$$f(x) = (x^2)^{\frac{1}{3}}\, with\; x \in [-2,2]$$

Where, 41 training patterns are generated for this nonlinear function. Polynomial SVR is chosen for approximating to the given nonlinear function. Where, $\mathrm{K}(\mathbf{x}^s, \mathbf{x}) = [\mathbf{x}^T\mathbf{x}^s + 1]^2$ and $\emptyset(\mathbf{x}) = \left[x^2 \;\; \sqrt{2}\cdot x \;\; 1\right]^T$.

Table 3.2 A comparison of ε-insensitive polynomial SVR and the developed robust and low complex SVR models for the nonlinear function $f(x) = (x^2)^{\frac{1}{3}}$ with $\varepsilon_2 = 1,\ C = 10$.

$f(x) = (x^2)^{\frac{1}{3}}$		Percentage of Root mean square Difference (PRD)			Root Mean Square Error (RMSE)			Norm of w			CPU time (In seconds)		
	Epsilon	0	0.01	0.1	0	0.01	0.1	0	0.01	0.1	0	0.01	0.1
SVR Models													
$1 - e^{-\frac{\boldsymbol{w}^2}{\varepsilon_2^2}}$	Train	48.006	38.822	41.645	0.194	0.159	0.170	0.747	0.679	0.663	0.328	0.344	0.344
	Test	52.365	54.511	50.942	0.227	0.232	0.218						
$Pw^2 + \left(1 - e^{-\frac{\boldsymbol{w}^2}{\varepsilon_2^2}}\right)$	Train	48.253	40.193	37.224	0.181	0.169	0.158	0.738	0.635	0.656	0.328	0.328	0.344
	Test	49.984	52.875	58.724	0.234	0.216	0.228						
$\frac{1}{\beta}\mathrm{lncosh}_{\varepsilon}(\beta e)$	Train	43.556	44.248	44.794	0.183	0.180	0.197	0.629	0.649	0.596	0.344	0.344	0.344
	Test	56.694	46.619	50.447	0.202	0.199	0.189						
$\frac{1}{\beta}\ln\left(1 + \frac{1}{2}[\beta e]_{\varepsilon}^2\right)$	Train	41.910	43.104	46.607	0.190	0.180	0.199	0.607	0.634	0.598	0.406	0.391	0.391
	Test	56.571	48.102	52.611	0.193	0.200	0.189						
ε-insensitive polynomial SVR	Train	47.848	43.030	44.683	0.196	0.169	0.197	0.726	0.712	0.655	0.344	0.344	0.328
	Test	48.985	51.674	47.018	0.205	0.231	0.173						

The obtained regression functions for both of the test functions are depicted in Figure 3.1-3.8.

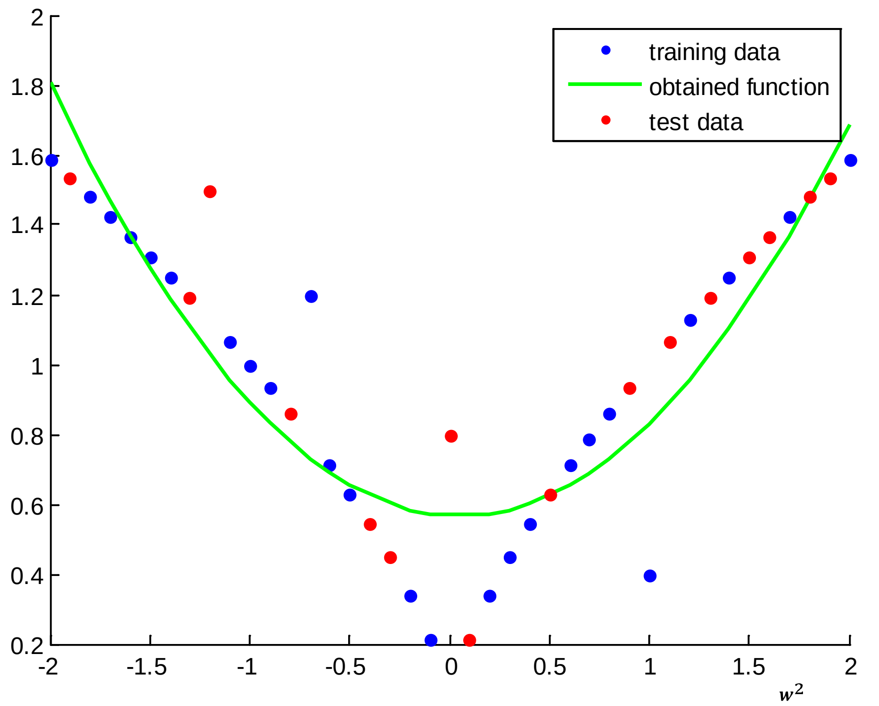


Figure 3.1 ε-insensitive polynomial SVR model obtained with $1-e^{-\frac{w^2}{\varepsilon_2^2}}$ norm for $f(x)=(x^2)^{\frac{1}{3}}$

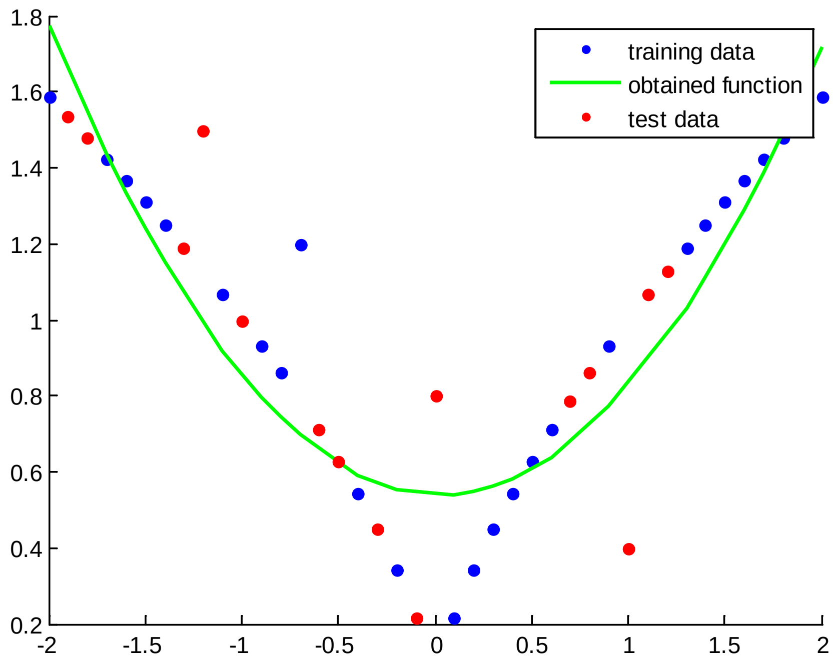


Figure 3.2 ε-insensitive polynomial SVR model obtained with $\frac{1}{\beta}\ln\cosh_{\varepsilon}$ for $f(x)=(x^2)^{\frac{1}{3}}$

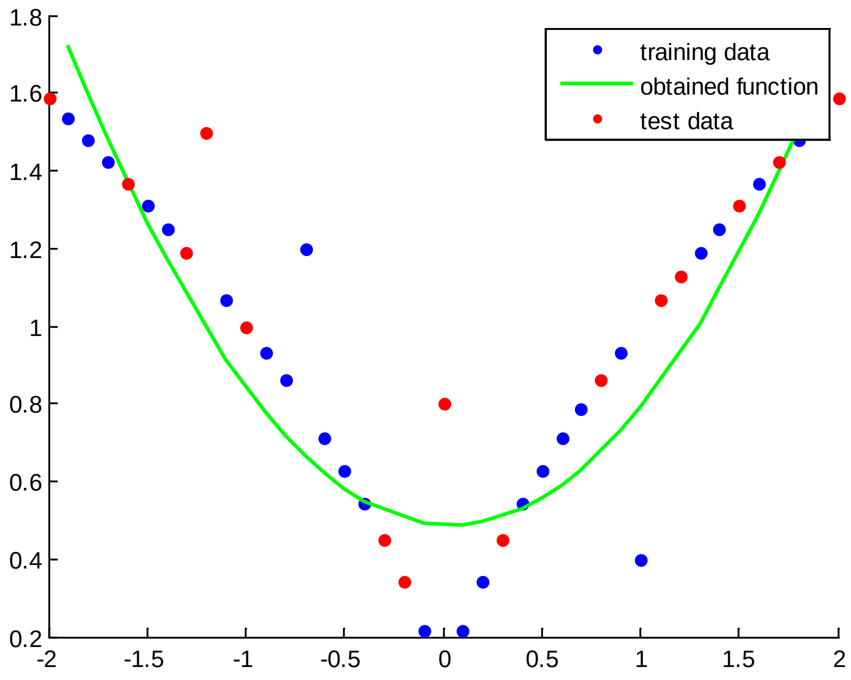


Figure 3.3 ε-insensitivep polynomial SVR model obtained with $\frac{1}{\beta}\ln\left(1+\frac{1}{2}[\beta e]_{\varepsilon}^{2}\right)$ for $f(x) = (x^2)^{\frac{1}{3}}$

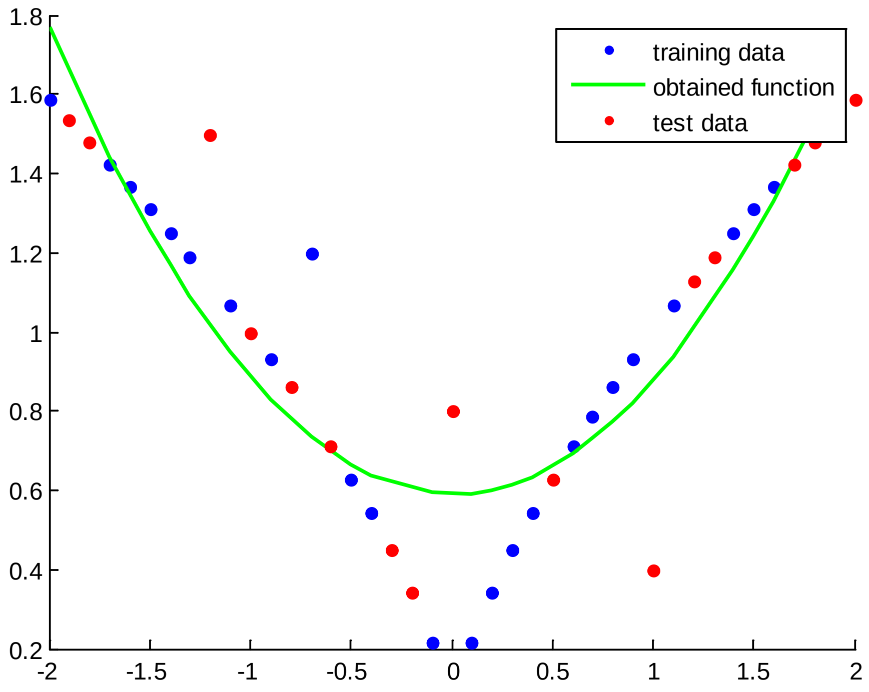


Figure 3.4 ε-insensitive polynomial SVR model for $f(x) = (x^2)^{\frac{1}{3}}$

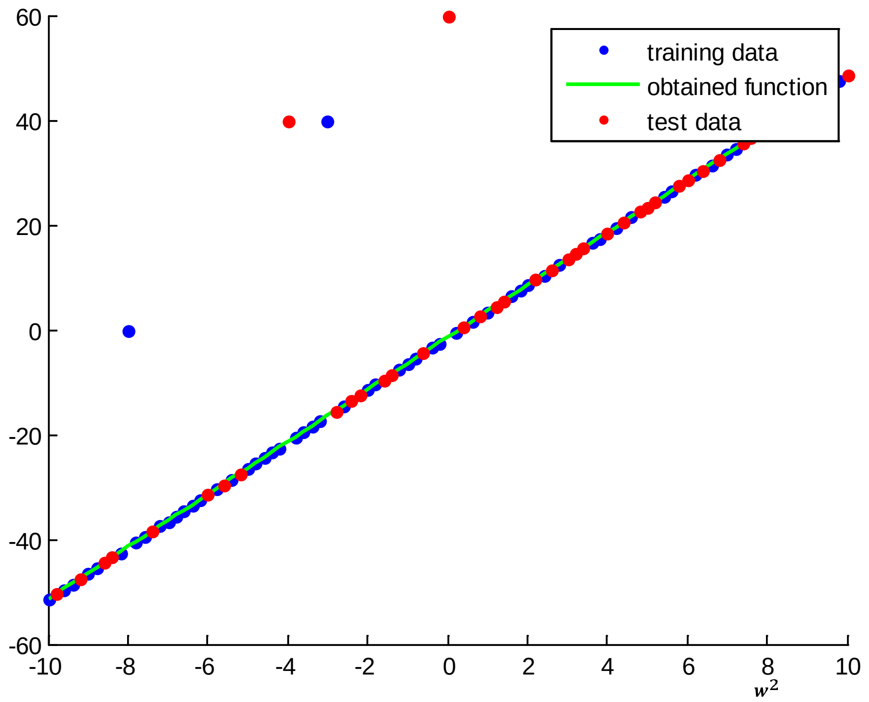


Figure 3.5 ε-insensitive linear SVR model obtained with $1 - e^{-\frac{w^2}{\varepsilon_2^2}}$ for $f(x) = 5x - 5$

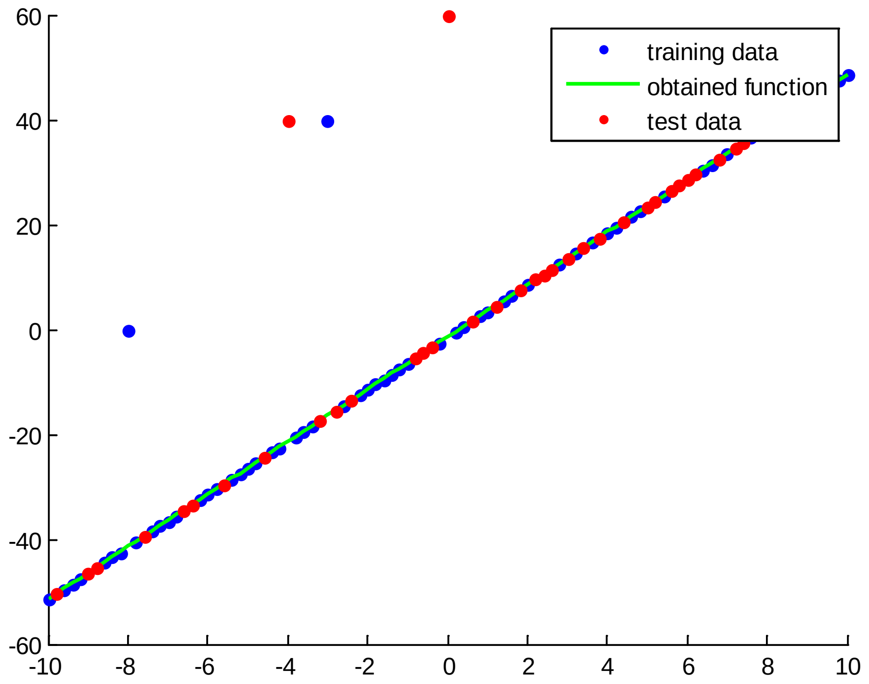


Figure 3.6 5 ε-insensitive linear SVR model obtained with $\frac{1}{\beta}\text{lncosh}_{\varepsilon}$ for $f(x) = 5x - 5$

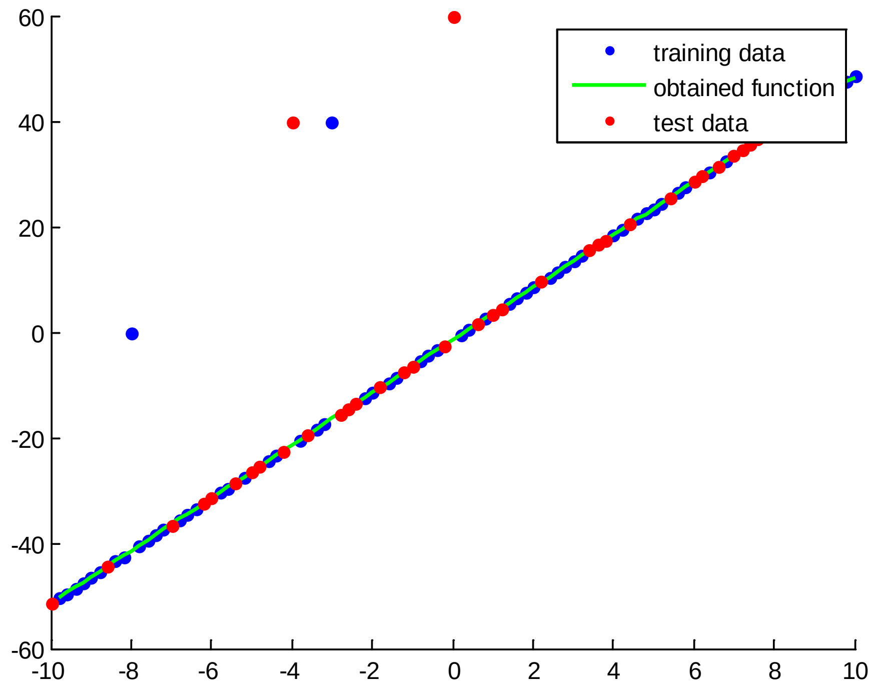


Figure 3.7 5 ε-insensitive linear SVR model obtained with $\frac{1}{\beta}\ln\left(1+\frac{1}{2}[\beta e]_{\varepsilon}^{2}\right)$ for the $f(x) = 5x - 5$

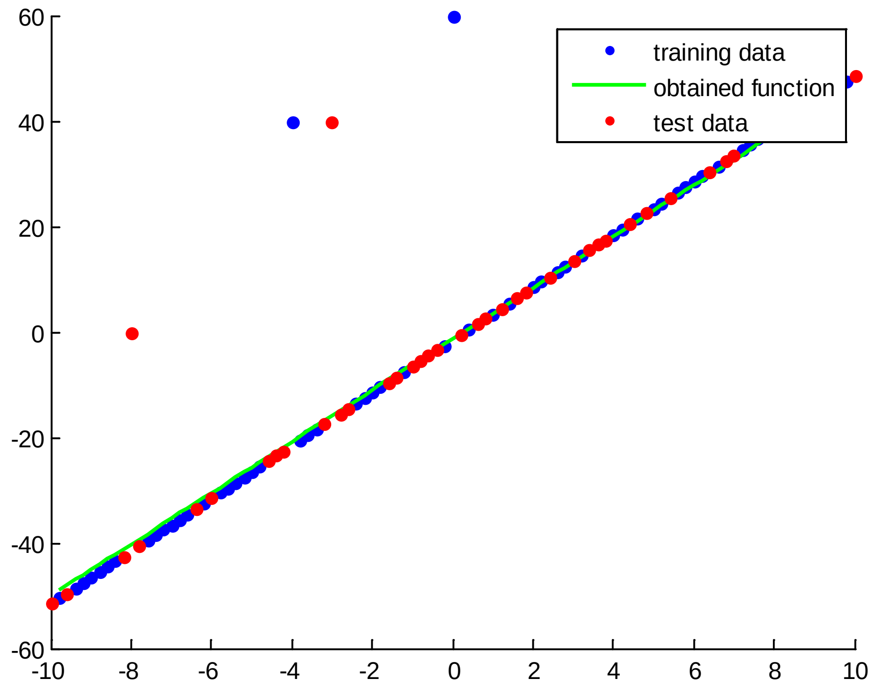


Figure 3.8 ε-insensitive linear SVR model for $f(x) = 5x - 5$

CHAPTER FOUR

LEAST SQUARE SUPPORT VECTOR REGRESSION WITH EPSILON INSENSITIVE QUADRATIC LOSS FUNCTION

This chapter extends Least Squares Support Vector Regression (LS-SVR) with squared loss function to the case of LS-SVR with epsilon insensitive squared loss function that ignores the small errors less than a predetermined number ε. The LS-SVR ,which is a modified version of standard Support Vector Regression (SVR), is introduced by Saunders, et all, (1998) and Suykens, et all, (2002). In conventional SVR, the ε-insensitive loss function is used as the cost function and it is represented by the inequality constraints. In the LS-SVR, the squared loss function is used as the cost function and the errors terms are represented as the equality constraints and the minimization problem is eventually converted to solving a linear algebraic equation system. Nonlinear identification and modeling, function approximation and optimal control are among the numerous applications of LS-SVR (Goethals, et all., 2005; Espinoza, et al., 2005; Espinoza, et al., 2004; Suykens,et. all, 2000; Jiang, et. all, 2009; Suykens,et. all, 2001; Suykens, 2001; Espinoza, et. all 2005, 2006; Wu, 2006; Pelckmans, et. all., 2005).

In the following, the derivation of LS-SVR and its proposed ε-insensitive version will be presented and their associated solutions will be compared in a qualitative way. The comparison will also be made with conventional least square solution firstly for the linear one-dimensional case for simplicity by no means of loosing generality.

4.1 Least Square Support Vector Regression

In this subsection, LS-SVR is described in terms of least squares and ridge regression.

In Section 2.2, least square approximation is described in a general framework. According to this, for a set of domain-range samples $\{(\boldsymbol{x}^s, y^s)\}_{s=1}^{L}$ with $\boldsymbol{x}^s \in R^n$ $y^s \in R$, regression estimation is defined as

$$y^s = \boldsymbol{w}^T \emptyset(\mathbf{x}^s) + b + e^s \tag{4.1}$$

Where, $\emptyset(\cdot): R^{\mathrm{n}} \rightarrow R^{\mathrm{m}}$ is a nonlinear function which maps the input data into a high dimensional feature space, $\boldsymbol{w} \in R^{\mathrm{m}}$ is a weight vector, $e^s \in R$ is the error variable and b is the bias term. Then, the cost function is:

$$L_{LS}(y, f(\boldsymbol{x})) = \sum_{s=1}^{L} [y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) + b]^2 \tag{4.2}$$

The cost function of ridge regression is the following modified version of the least squares approximation cost (Saunders, Gammerman, & Vovk 1998):

$$L_R(y, f(\boldsymbol{x})) = \frac{1}{2} \parallel \boldsymbol{w} \parallel_2^2 + \frac{C}{2} \sum_{s=1}^{L} [y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b]^2 \tag{4.3}$$

Where, C is a fixed positive constant. Note that the cost function in (4.3) consists of a least square error and a regularization term. $C = 0$, which is a special case of ridge regression, corresponds to least squares regression (Saunders, et all, 1998).

One may desire to obtain the dual space representation instead of the primal space for computational purposes (Suyken et al., 2002). For the LS-SVR, the following optimization problem in the primal weight space can be described in the dual space as explained in the sequel.

$$\min_{w \in R^m, b \in R} \frac{1}{2} \parallel \boldsymbol{w} \parallel_2^2 + \frac{C}{2} \sum_{s=1}^{L} (e^s)^2 \tag{4.4}$$

$$\text{subject to } y^s = \boldsymbol{w}^T \emptyset(\mathbf{x}^s) + b + e^s, \qquad s \in \{1, \ldots, L\}$$

One defines the corresponding Lagrangian form as:

$$J(\boldsymbol{w}, b, e; \alpha_s) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{2}\sum_{s=1}^{L}(e^s)^2 - \sum_{s=1}^{L}\alpha_s(\boldsymbol{w}^T\emptyset(\mathbf{x}^s) + b + e^s - y^s) \tag{4.5}$$

Where, $\alpha_s \in R$ are the Lagrange multipliers (also called as support vectors) in the SVR literature. First order necessary conditions for the optimality are:

$$\frac{dJ}{d\boldsymbol{w}} = \boldsymbol{w} - \sum_{s=1}^{L}\alpha_s\,\emptyset(\mathbf{x}^s) = 0 \rightarrow \boldsymbol{w} = \sum_{s=1}^{L}\alpha_s\,\emptyset(\mathbf{x}^s) \tag{4.6}$$

$$\frac{dJ}{db} = \sum_{s=1}^{L}\alpha_s = 0 \tag{4.7}$$

$$\frac{dJ}{de^s} = \alpha_s - Ce^s = 0 \tag{4.8}$$

$$\frac{dJ}{d\alpha_s} = \boldsymbol{w}^T\emptyset(\mathbf{x}^s) + b + e^s - y^s = 0 \tag{4.9}$$

These conditions are similar to the Vapnik's SVR optimality conditions, except for the condition $\alpha_s = Ce_s$. Eliminating the variables $\boldsymbol{w}$ and e^s, the following cost function which should be maximized is obtained:

$$J(\alpha_s) = -\frac{1}{2}\sum_{s=1}^{L}\sum_{r=1}^{L}\alpha_s\,\emptyset(\mathbf{x}^s)\,\emptyset(\mathbf{x}^r)\alpha_r - \frac{1}{2C}\sum_{s=1}^{L}{\alpha_s}^2 + \sum_{s=1}^{L}\alpha_s y^s \tag{4.10}$$

which can be rewritten as the following minimization problem.

$$\min_{\boldsymbol{\alpha}\in R^L} J(\boldsymbol{\alpha}) = \frac{1}{2}\alpha^T \mathrm{K}(\mathbf{x}^s, \mathbf{x}^\mathrm{r})\alpha + \frac{1}{2C}\alpha^T\alpha - y^T\alpha$$

$$\text{subject to } \sum_{s=1}^{L}\alpha_s = 0 \tag{4.11}$$

Instead of solving the above quadratic optimization problem, one may prefer to solve the following set of linear algebraic equation system which is obtained from first order optimality conditions.

$$\begin{bmatrix} 0 & \boldsymbol{u}^T \\ \boldsymbol{u} & \mathrm{K} + \frac{1}{C}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{4.12}$$

Where, $\boldsymbol{I} \in R^{L\times L}$ is an identity matrix, $\boldsymbol{u} = [1, \dots, 1]^T \in R^L$, $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_L]^T \in R^L$, $y = [y^1, \dots, y^L]^T \in R^L$, $\mathrm{K} = \mathrm{K}(\mathbf{x}^s, \mathbf{x}^r) = [\emptyset^{\mathrm{T}}(\mathbf{x}^s) \cdot \emptyset(\mathbf{x}^r)]_{\mathrm{s,r}} \in R^{L\times L}$

For the test sample **x,** one can predict the range of the learned approximate function as:

$$f(\mathbf{x}) = \mathbf{w}^{\mathbf{T}}\emptyset(\mathbf{x}) + \boldsymbol{b} = \sum_{s=1}^{L} \alpha_s\, \emptyset^{\mathrm{T}}(\mathbf{x}^s).\, \emptyset(\mathbf{x}) + b = \sum_{s=1}^{L} \alpha_s\, \mathrm{K}(\mathbf{x}^s, \mathbf{x}) + b \tag{4.13}$$

Where, α_s's and b are the solutions to the equation in (4.12).

4.2 ε-insensitive Least Square Support Vector Regression

This subsection presents ε-Insensitive Least Squares Support Vector Regression (LS-SVR) model which constitutes one of the contribution of the thesis. It is derived by inspiring Vapnik's ε-insensitive loss function that ignores errors which are within a determined distance of desired output (Figure 4.1). It is an improvement on LS-SVR model towards obtaining less number of support vectors.

For the ε-insensitive LS-SVR derivation, the optimization problem is chosen as:

$$\min_{\boldsymbol{w}\in R^m, b\in R} \frac{1}{2} \parallel \boldsymbol{w} \parallel_2^2 + \frac{C}{2} \sum_{s=1}^{L} (e^s)_\varepsilon^2$$
$$\text{subject to } y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b = e^s \text{ with } s \in \{1, \dots, L\} \tag{4.14}$$

Where, the ε-insensitive quadratic loss function is defined as:

$$(e)_\varepsilon^2 = \begin{cases} (e-\varepsilon)^2 & if\ e \geq \varepsilon \\ 0 & if - \varepsilon \leq e \leq \varepsilon \\ (e+\varepsilon)^2 & if\ e \leq -\varepsilon \end{cases} \tag{4.15}$$

The graphs of the ε-insensitive quadratic cost and its derivative are illustrated by Figure 4.1 (a) and (b) respectively.

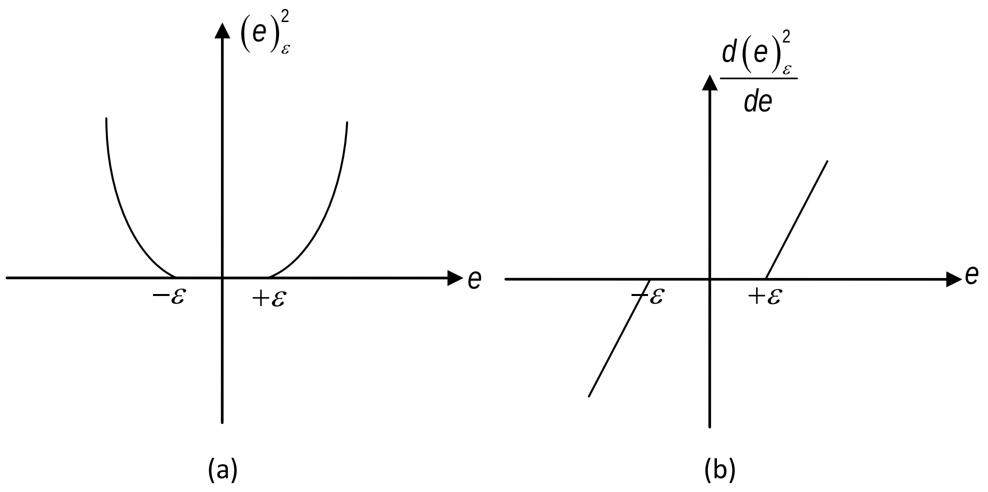


Figure 4.1 (a) ε-insensitive quadratic loss functions (b) its derivative

It should be noted that $(e)_\varepsilon^2$ is a continuous differentiable function whose derivative has the following canonical representation.

$$\frac{\partial(e)_\varepsilon^2}{\partial e} = 2e + |e-\varepsilon| - |e+\varepsilon|$$

One defines the Lagrangian form as:

$$J(\boldsymbol{w}, b, e; \alpha_s) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{2}\sum_{s=1}^{L}(e^s)_\varepsilon^2 - \sum_{s=1}^{L}\alpha_s(\boldsymbol{w}^T\emptyset(\mathbf{x}^s) + b + e^s - y^s) \tag{4.16}$$

First order necessary conditions for its optimality are:

$$\frac{dJ}{d\boldsymbol{w}} = \boldsymbol{w} - \sum_{s=1}^{L} \alpha_s\, \emptyset(\mathbf{x}^s) = 0 \rightarrow \boldsymbol{w} = \sum_{s=1}^{L} \alpha_s\, \emptyset(\mathbf{x}^s) \tag{4.17}$$

$$\frac{dJ}{db} = \sum_{s=1}^{L} \alpha_s = 0 \tag{4.18}$$

$$\frac{dJ}{d\alpha_s} = y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) - b - e^s = 0 \tag{4.19}$$

$$\frac{dJ}{de^s} = \frac{C}{2}[-2e^s + | e^s - \varepsilon | - | e^s + \varepsilon |] - \alpha_s = 0 \tag{4.20}$$

In order to reach a cost in terms of the Lagrange multipliers α_s's, e^s should be solved in terms of α_s from (4.20).

There are three different regions in each of which e^s and α_s are related to each other in an affine way:

$$\left.\begin{array}{l} if\ e^s \geq \varepsilon, \text{ then} \\ \frac{C}{2}2(e^s - \varepsilon) - \alpha_s = 0 \rightarrow e^s = \frac{1}{C}\alpha_s + \varepsilon \\ \\ if - \varepsilon \leq e^s \leq \varepsilon, \text{ then} \\ 0 - \alpha_s = 0 \rightarrow \alpha_s = 0 \\ \\ if\ e^s \leq -\varepsilon, \text{ then} \\ \frac{C}{2}2(e^s + \varepsilon) - \alpha_s = 0 \rightarrow e^s = \frac{1}{C}\alpha_s - \varepsilon \end{array}\right\} e^s = \frac{1}{C}\alpha_s + \varepsilon \cdot sign^*(\alpha_s) \tag{4.21}$$

Where,

$$sign^*(\alpha_s) = \begin{cases} 1 & if\ \ \alpha_s > 0 \\ \lambda \text{ with } \lambda \in [-1,1] & if\ \ \alpha_s = 0 \\ -1 & if\ \ \alpha_s < 0 \end{cases}$$

Where, $*$ means that the signum function is identical to the usual signum function except at $\alpha = 0$ values for which signum becomes multi-valued with $sign^*(\alpha) = \lambda$ with $\lambda \in [-1,1]$. On the other hand, $sign_\varepsilon^*(\alpha)$ is the ε-insensitive version of the signum function which is defined as:

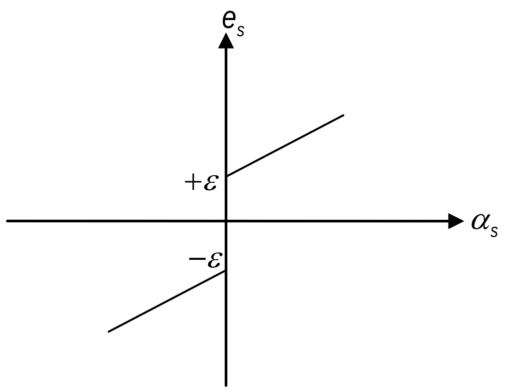


Figure 4.2 e^s graphics with respect to α_s

So, eliminating the variables $\boldsymbol{w}$ and e^s, the following formulation for the ε-insensitive LS-SVR is obtained.

$$\max_{\alpha \in R^L} J(\alpha_s) = \frac{1}{2}\sum_{s=1}^{L}\sum_{s=1}^{L} \alpha_s{}^T \emptyset(\mathbf{x}^s)^T \emptyset(\mathbf{x}^r)\alpha_r$$
$$+\frac{C}{2}\sum_{s \epsilon L_+}\left(\frac{1}{C}\alpha_s + \varepsilon\right)^2 + \frac{C}{2}\sum_{s \epsilon L_-}\left(\frac{1}{C}\alpha_s - \varepsilon\right)^2 + \frac{C}{2}\sum_{s \epsilon L_0} 0$$
$$-\sum_{s=1}^{L}\alpha_s\left(\sum_{s=1}\alpha_s\, \emptyset(\mathbf{x}^s)^T \emptyset(\mathbf{x}^s) + b\right.$$
$$\left. + \left[\sum_{s \epsilon L_+}\left(\frac{1}{C}\alpha_s + \varepsilon\right) + \sum_{s \epsilon L_-}\left(\frac{1}{C}\alpha_s - \varepsilon\right) + \sum_{s \epsilon L_0} 0\right] - y^s\right)$$

Note that, as $\sum_{s=1}^{L} \alpha_s = 0$, the bias term b is disappear.

$$\max_{\alpha\in R^L} J(\alpha_s) = -\frac{1}{2}\sum_{s=1}^{L}\sum_{s=1}^{L}\alpha_s{}^T \mathrm{K}(\mathbf{x}^s,\mathbf{x}^r)\alpha_r + \frac{C}{2}\sum_{s\epsilon L_+}\left(\frac{1}{C}\alpha_s + \varepsilon\right)^2 + \frac{C}{2}\sum_{s\epsilon L_-}\left(\frac{1}{C}\alpha_s - \varepsilon\right)^2 - \sum_{s=1}^{L}\alpha_s\left[\frac{1}{2}\sum_{s\epsilon L_+}\left(\frac{1}{C}\alpha_s + \varepsilon\right) + \frac{1}{2}\sum_{s\epsilon L_-}\left(\frac{1}{C}\alpha_s - \varepsilon\right)\right] + \sum_{s=1}^{L}\alpha_s y^s \tag{4.22}$$

Where, $L_+ \triangleq \{s \in \{1,\dots,L\} \mid e^s > \varepsilon\}$

$$L_- \triangleq \{s \in \{1,\dots,L\} \mid e^s < -\varepsilon\} \tag{4.23}$$

$L_0 \triangleq \{s \in \{1,\dots,L\} \mid -\varepsilon \le e^s \le \varepsilon\}$

Or alternatively, (4.22) can be rewritten as in the following minimization problem:

$$\min_{\alpha\in R^L} J(\alpha) = \frac{1}{2}\sum_{s=1}^{L}\sum_{s=1}^{L}\alpha_s \mathrm{K}(\mathbf{x}^s,\mathbf{x}^r)\alpha_r - \frac{C}{2}\left[\sum_{s\epsilon L_+}\left(\frac{1}{C}\alpha_s + \varepsilon\right)^2 + \sum_{s\epsilon L_-}\left(\frac{1}{C}\alpha_s - \varepsilon\right)^2\right] + \sum_{s=1}^{L}\alpha_s\left[\frac{1}{2}\sum_{s\epsilon L_+}\left(\frac{1}{C}\alpha_s + \varepsilon\right) + \frac{1}{2}\sum_{s\epsilon L_-}\left(\frac{1}{C}\alpha_s - \varepsilon\right)\right] - \sum_{s=1}^{L}\alpha_s y^s \tag{4.24}$$

subject to $\sum_{s=1}^{L}\alpha_s = 0$

As an alternative, using the compact representation in (4.21), (4.24) can be written as:

$$\min_{\alpha\in R^L} J(\alpha) = \frac{1}{2}\sum_{s=1}^{L}\sum_{s=1}^{L}\alpha_s \mathrm{K}(\mathbf{x}^s,\mathbf{x}^r)\alpha_r - \frac{C}{2}\sum_{s=1}^{L}\left(\frac{1}{C}\alpha_s + \varepsilon\cdot sign^*(\alpha_s)\right)^2 + \sum_{s=1}^{L}\alpha_s\left(\frac{1}{C}\alpha_s + \varepsilon\cdot sign^*(\alpha_s) - y^s\right) \tag{4.25}$$

Then, the following dual optimization formulation is obtained for the ε-Insensitive LS-SVR.

$$\min_{\alpha\in R^L} J(\alpha) = \frac{1}{2}\sum_{s=1}^{L}\sum_{s=1}^{L}\alpha_s \mathrm{K}(\mathbf{x}^s,\mathbf{x}^r)\alpha_r + \frac{1}{2C}\sum_{s=1}^{L}(\alpha_s)^2 - \sum_{s=1}^{L}\alpha_s y^s + \frac{C\varepsilon^2}{2}\sum_{s=1}^{L}(sign^*(\alpha_s))^2 \tag{4.26}$$

As can be seen from (4.26) and (4.10), the difference between LS-SVR and ε-insensitive LS-SVR is at the number support vector number due to the ε-insensitive tube. Because, the support vectors in the ε-insensitive tube are zero.

With the optimum values of α_s obtained by solving (4.26), one may determine the optimum value of $\boldsymbol{w}$**,** and so the approximation function as:

$$\begin{aligned} y^s &= \boldsymbol{w}^T \emptyset(\mathbf{x}^s) + b \\ &= \sum_{s=1}^{L}\alpha_s\, \emptyset^{\mathrm{T}}(\mathbf{x}^s).\emptyset(\mathbf{x}) + b = \sum_{s=1}^{L}\alpha_s\, \mathrm{K}(\mathbf{x}^s,\mathbf{x}) + b \end{aligned} \tag{4.27}$$

4.3 A qualitative analysis of $\boldsymbol{\varepsilon}$-insensitive LS-SVR

The analysis will be done for 1-dimensional and linear case for the sake of simplicity. Considering the sample set $\{(x^s,y^s)\}_{s=1}^{L}$ with $x^s,y^s \in R$, ε-insensitive LSSVR can be formulated as the minimization of the following cost function.

$$J(w,e_s) = \frac{1}{2}w^2 + \frac{C}{2}\sum_{s=1}^{L}(y^s - wx^s)_\varepsilon^2 \tag{4.28}$$

ε-insensitive squared loss function is continuously differentiable, so first order necessary condition for optimality can be stated as:

$$\frac{\partial J}{\partial w} = w + \frac{C}{2}\sum_{s=1}^{L}[2(y^s - wx^s) + | y^s - wx^s - \varepsilon | - | y^s - wx^s + \varepsilon |] \cdot (-x^s) = 0 \quad (4.29)$$

By using (4.29), one obtains w in terms of the x^s samples as:

$$w = \frac{C}{2}\left[\sum_{s=1}^{L}[2(y^s - wx^s) + | y^s - wx^s - \varepsilon | - | y^s - wx^s + \varepsilon |]\, x^s\right] \quad (4.30)$$

(4.30) is an implicit function of w. It is shown in the sequel, w can be obtained explicitly in terms of the samples. For this purpose, one can write (4.30) as follows:

$$w = \frac{C}{2}\sum_{s\epsilon L_0} 0 + \frac{C}{2}\sum_{s\epsilon L_+} 2(y^s - wx^s - \varepsilon)x^s + \frac{C}{2}\sum_{s\epsilon L_-} 2(y^s - wx^s + \varepsilon)x^s \quad (4.31)$$

Where, L_0, L_+ and L_- sets are defined as in (4.23).

Then, (4.31) can also be given as in (4.32).

$$w\left[1 + C\sum_{s\epsilon L_+}(x^s)^2 + C\sum_{s\epsilon L_-}(x^s)^2\right] = C\sum_{s\epsilon L_0} 0 + C\sum_{s\epsilon L_+}(y^s - \varepsilon)x^s + C\sum_{s\epsilon L_-}(y^s + \varepsilon)x^s \quad (4.32)$$

So, the optimal w is found as:

$$w = \frac{\left[C\sum_{s\epsilon L_0} 0 + C\sum_{s\epsilon L_+}(y^s - \varepsilon)x^s + C\sum_{s\epsilon L_-}(y^s + \varepsilon)x^s\right]}{\left[1 + C\sum_{s\epsilon L_+}(x^s)^2 + C\sum_{s\epsilon L_-}(x^s)^2\right]} \quad (4.33)$$

As an alternative, it can be written as:

$$w = \frac{\left[\sum_{s=1}^{L} S_1 \cdot y^s x^s - \varepsilon \sum_{s=1}^{L} S_2 \cdot x^s\right]}{\left[\frac{1}{C} + \sum_{s=1}^{L} S_1 \cdot (x^s)^2\right]} \tag{4.34}$$

Where,

$$S_1 = \frac{2 + sign(y^s - wx^s - \varepsilon) - sign(y^s - wx^s + \varepsilon)}{2}$$

$$S_2 = sign_\varepsilon(y^s - wx^s)$$

4.4 Comparison of Least Square Regression, LS-SVR and SVR

For a given a set of samples $\{(x^s, y^s)\}_{s=1}^{L}$ with $x^s \in R$ and $y^s \in R$, the linear regression with $b = 0$ can be defined as:

$$y^s = f(x^s) = wx^s + e \tag{4.35}$$

Where, $e \in R$ is the error variable.

Linear least squares regression for 1-dimensional case:

To determine the coefficient w, one can minimize the following sum of squared error for a given sample set.

$$\sum_{s=1}^{L} (e^s)^2 = \sum_{s=1}^{L} [y^s - wx^s]^2 \tag{4.36}$$

To use first order necessary conditions for optimality, one may differentiate the total squared error with respect to the coefficient w and then setting it to zero. As a result, the normal equations are obtained.

$$\frac{d}{dw}\left(\sum_{s=1}^{L}[y^s - wx^s]^2\right) = 0, \qquad s \in \{1,2,\cdots,L\} \tag{4.37}$$

which yields

$$-2\left(\sum_{s=1}^{L}[y^s - wx^s]\right)x^s = 0 \tag{4.38}$$

So, the optimal w is calculated as

$$w = \sum_{s=1}^{L}\left[\frac{y^s}{\sum_{s=1}^{L}(x^s)^2}\right]x^s \quad \text{or} \quad w = \frac{\sum_{s=1}^{L} y^s x^s}{\sum_{s=1}^{L}(x^s)^2} \tag{4.39}$$

Then, the approximate function is given as:

$$y = f(x) = \frac{\sum_{s=1}^{L} y^s x^s}{\sum_{s=1}^{L}(x^s)^2} \cdot x \tag{4.40}$$

LS-SVR for 1-dimensional linear case:

Assuming $b = 0$ and considering sample set $\{(x^s, y^s)\}_{s=1}^{L}$ with $x^s, y^s \in R$, LS-SVR is formulated as the following minimization problem:

$$\min_{w\in R} \frac{1}{2}w^2 + \frac{C}{2}\sum_{s=1}^{L}(e^s)^2$$
$$\text{subject to } y^s = wx^s + e^s, \qquad s \in \{1,\dots,L\} \tag{4.41}$$

One defines the Lagrangian form as:

$$J(w,e;\alpha_s) = \frac{1}{2}w^2 + \frac{1}{2}C\sum_{s=1}^{L}(e^s)^2 - \sum_{s=1}^{L}\alpha_s(wx^s + e^s - y^s) \tag{4.42}$$

Where, $\alpha_s \in R$ are the Lagrange multipliers. Its first order optimality conditions are:

$$\frac{dJ}{dw} = w - \sum_{s=1}^{L} \alpha_s x^s = 0 \rightarrow w = \sum_{s=1}^{L} \alpha_s x^s \tag{4.43}$$

$$\frac{dJ}{de^s} = Ce^s - \alpha_s = 0 \tag{4.44}$$

$$\frac{dJ}{d\alpha_s} = wx^s + e_s - y^s = 0 \tag{4.45}$$

To find the optimal w, Equation (4.44) and (4.45) can be substituted into the (4.43) and then solved as:

$$w = C\sum_{s=1}^{L} e_s x^s = C\sum_{s=1}^{L} (y^s - wx^s)\, x^s$$

$$w\left(1 + C\sum_{s=1}^{L}(x^s)^2\right) = C\sum_{s=1}^{L} y^s x^s$$

$$w = \sum_{s=1}^{L}\left(\frac{y^s}{\frac{1}{C} + \sum_{s=1}^{L}(x^s)^2}\right)x^s \qquad \text{or} \qquad w = \frac{\sum_{s=1}^{L} y^s x^s}{\frac{1}{C} + \sum_{s=1}^{L}(x^s)^2} \tag{4.46}$$

The following Lagrange multipliers α_s in (4.46) serve as the weights determining the contributions of the samples to the optimal weight.

$$\alpha_s = \frac{y^s}{\frac{1}{C} + \sum_{s=1}^{L}(x^s)^2} \tag{4.47}$$

The approximation obtained by LS-SVR is given as

$$y = f(x) = \sum_{s=1}^{L} \frac{y^s}{\frac{1}{C} + \sum_{s=1}^{L}(x^s)^2} x^s \cdot x \tag{4.48}$$

SVR for 1-dimensional linear case:

Considering sample set $\{(x^s, y^s)\}_{s=1}^{L}$ with $x^s, y^s \in R$ and assuming $b = 0$, SVR is formulated as the following minimization problem:

$$\min_{w \in R} \frac{1}{2} w^2 + C \sum_{s=1}^{L} |y^s - wx^s|_{\varepsilon} \tag{4.49}$$

The find the optimal coefficients for the minimization problem in (4.49), one can take the subgradient of the cost function with respect to w (Bertsekas, 1995). It is a fact that minimum of PWA convex function occurs at a vertex where 0 (zero) belongs to the subgradient (Bertsekas, 1995), i.e.

$$\nabla_w^* J(w, b) = w + C \sum_{s=1}^{L} sign_{\varepsilon}^*[y^s - wx^s](-x^s) = 0$$

On the other hand, $sign_{\varepsilon}^*(\alpha)$ is the ε-insensitive version of the signum function which is defined as:

$$sign_{\varepsilon}^*(\alpha) = \begin{cases} +1, & \text{for } \alpha > \varepsilon \\ -1, & \text{for } \alpha < -\varepsilon \\ 0, & -\varepsilon < \alpha < \varepsilon \\ \lambda \text{ with } \lambda \in [0,1], & \text{for } \alpha = \varepsilon \\ \lambda \text{ with } \lambda \in [-1,0], & \text{for } \alpha = -\varepsilon \end{cases}$$

Then, w can be solved in terms of the samples and specific values of λ^* as:

$$w = C \sum_{s=1}^{L} sign_{\varepsilon}^*[y^s - wx^s] x^s \tag{4.50}$$

The approximation obtained by SVR is given as:

$$y = f(x) = C \sum_{s=1}^{L} sign_{\varepsilon}^*[y^s - w^T x^s] x^s \cdot x \tag{4.51}$$

To optimal w parameters obtained by four different regression methods are given in Table 4.1 for understanding their differences in a clear way.

Table 4.1 Optimal w parameters for linear regression in three methods, namely, Least Square Regression (LSR) , LS-SVR and SVR.

Table 4.1 The value of optimal w in the least square, LS-SVR and SVR methods.

	LSR	LS-SVR	ε-Insensitive LS-SVR	SVR
w	$= \sum_{s=1}^{L} \left[\frac{y^s}{\left[\sum_{s=1}^{L} (x^s)^2\right]} \right] x^s$	$= \sum_{s=1}^{L} \left[\frac{y^s}{\frac{1}{C} + \sum_{s=1}^{L} (x^s)^2} \right] x^s$	$= \frac{\left[\sum_{s=1}^{L} S_1 \cdot y^s x^s - \varepsilon \sum_{s=1}^{L} S_2 \cdot x^s\right]}{\left[\frac{1}{C} + \sum_{s=1}^{L} S_1 \cdot (x^s)^2\right]}$	$= C \sum_{s=1}^{L} sign_{\varepsilon}^{*}[wx^s - y^s] x^s$

Where,

$$S_1 = \frac{2 + sign(y^s - wx^s - \varepsilon) - sign(y^s - wx^s + \varepsilon)}{2}$$

$$S_2 = sign_{\varepsilon}(y^s - wx^s)$$

As can be seen Table 4.1, one can conclude that data points out of the prescribed parameter epsilon bound are support vectors in SVR. The contributions of support vectors contained out of the ε-insensitive to the optimal parameter are the same. On the other hand, the difference between LS-SVR and LSR is only at the complexity parameter $\frac{1}{C}$. When C goes to infinity, LS-SVR solution approaches to LSR solution, more precisely, to the generalized inverse solution. In ε-insensitive LS-SVR, the contributions are only due to the support vectors which are the samples out of the ε-insensitive tube and the contributions changes from one support vector to another depending on its position on the sample space.

CHAPTER FIVE
SUPPORT VECTOR REGRESSION
WITH PIECEWISE AFFINE KERNEL

The performance of the SVR largely depends on the kernels and the loss functions chosen. Therefore many different kernel functions for mapping are used in literature such as polynomial kernel, Gaussian kernel, sigmoidal kernel, etc. This chapter presents a new type of kernels which is called piecewise affine kernels where feature space is explicitly given with a piece-wise linear mapping from the input space. Moreover, it is also presented how the Support Vector Regression (SVR) with piecewise affine kernels can be formulated for function approximation. The introduced Piecewise Affine Regression (PWA-SVR) models are inspired by the canonical representations available for PWA functions described in Subsection (2.1.5).

5.1 1-dimensional (PWA-SVR) models

Let a function $f(\cdot): R \to R$ be given by the samples $\{(x^s, y^s)\}_{s=1}^{L}$. Consider the following piecewise affine function $\emptyset(\cdot): R \to R^m$ mapping from a one-dimensional input space into an m-dimensional feature space.

$$\emptyset(x) = \begin{bmatrix} 1 \\ x \\ |x-\gamma^1| \\ |x-\gamma^2| \\ \vdots \\ |x-\gamma^l| \end{bmatrix} \tag{5.1}$$

Then, the mapped data set becomes

$$\{(\emptyset(x^s), y^s)\}_{s=1}^{L}, \qquad \emptyset(x^s) \in R^{l+2}, \qquad y^s \in R \tag{5.2}$$

One can use the following PWA approximation for obtaining a support vector regression model.

$$f(x) = w^T \emptyset(x) \tag{5.3}$$

It should be noted that the bias term b is not included in the representation of (5.3) since the first basis function in (5.1) is chosen as unity for providing a bias term in the function space. The bias term b will not be included in the kernel representations developed in this section except for the compact kernel (5.31).

This regression problem can be formulated as the following optimization:

$$\begin{aligned} &\min_{\boldsymbol{w}} \frac{1}{2} \parallel \boldsymbol{w} \parallel_2^2 + C \sum_{s=1}^{L} (\xi_s + \xi_s') \\ &\text{subject to} \begin{cases} y^s - \boldsymbol{w}^T \emptyset(\mathbf{x}^s) \le \varepsilon + \xi_s \\ -y^s + \boldsymbol{w}^T \emptyset(\mathbf{x}^s) \le \varepsilon + \xi_s' \\ \xi_s, \xi_s' \ge 0, \qquad s \in \{1, \dots, L\} \end{cases} \end{aligned} \tag{5.4}$$

Formulation of $\boldsymbol{w} \in R^m$ in terms of the dual variables is obtained by using Lagrange multipliers as mentioned in the Subsection (2.2.2.2).

$$\boldsymbol{w} = \sum_{s=1}^{L} (\alpha_s - \alpha_s') \, \emptyset(\mathbf{x}^s) \tag{5.5}$$

and the support vector regression function (5.3) is found as:

$$f(x) = \boldsymbol{w}^T \emptyset(\mathbf{x}) = \sum_{s=1}^{L} (\alpha_s - \alpha_s') \, \mathrm{K}(\mathbf{x}^s, \mathbf{x}) \tag{5.6}$$

Where, $\mathrm{K}(\mathbf{x}^s, \mathbf{x}) = \emptyset^{\mathrm{T}}(\mathbf{x}^s) \cdot \emptyset(\mathbf{x})$ is the PWA kernel function defined as the follows.

$$\mathrm{K}(\mathbf{x}^s, \mathbf{x}) = \emptyset^{\mathrm{T}}(\mathbf{x}^s) \cdot \emptyset(\mathbf{x})$$

$$= [1 \;\; x^s \;\; |x^s - \gamma^1| \;\; |x^s - \gamma^2| \;\; \cdots \;\; |x^s - \gamma^l|] \cdot \begin{bmatrix} 1 \\ x \\ |x - \gamma^1| \\ |x - \gamma^2| \\ \vdots \\ |x - \gamma^l| \end{bmatrix}$$

$$\begin{aligned} &= [1 + x^s . x + |x^s - \gamma^1| . |x - \gamma^1| + |x^s - \gamma^2| . |x - \gamma^2| + \cdots \\ &\qquad + |x^s - \gamma^l| . |x - \gamma^l|] \end{aligned} \tag{5.7}$$

Substuting (5.7) into (5.6), one can obtain PWA-SVR model.

$$f(x)=\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,[1+x^s.x+|x^s-\gamma^1|.|x-\gamma^1| \\ +|x^s-\gamma^2|.|x-\gamma^2|+\cdots+|x^s-\gamma^l|.|x-\gamma^l|] \tag{5.8}$$

Where, the function $f(x)$ is a PWA function whose canonical parameters can found as follows.

$$\begin{aligned} f(x) &= \sum_{s=1}^{L}(\alpha_s-\alpha_s')\,[1+x^s\cdot x+|x^s-\gamma^1|\cdot|x-\gamma^1|+|x^s-\gamma^2|\cdot|x-\gamma^2|+\cdots \\ &\qquad +|x^s-x^l|\cdot|x-x^l|] \\ &= \underbrace{\sum_{s=1}^{L}(\alpha_s-\alpha_s')}_{a_o}+\underbrace{\left[\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,x^s\right]}_{a_1}x+\underbrace{\left[\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,|x^s-\gamma^1|\right]}_{b_1}|x-\gamma^1| \\ &\qquad +\cdots+\underbrace{\left[\sum_{s=1}^{L}(\alpha_s-\alpha_s')\,|x^s-x^l|\right]}_{b_l}|x-x^l| \\ &= a_o+a_1x+b_1|x-\gamma^1|+b_2|x-\gamma^2|+\cdots+b_l|x-\gamma^l| \\ &= a_o+a_1x+\sum_{j=1}^{l}b_j|x-\gamma^j| \end{aligned} \tag{5.9}$$

When $\gamma^j=x^s$ for all $s\in\{1,\dots,L\}$ with $l=L$, the function in (5.9) can be represented as the following PWA canonical representation.

$$f(x)=a_o+a_1x+\sum_{s=1}^{L}b_s|x-x^s| \tag{5.10}$$

5.2 n-dimensional PWA-SVR models (lattice partition case)

Let a set of samples $\{(\mathbf{x}^s,y^s)\}_{s=1}^{L}$ be given. Where, $\mathbf{x}^s\in R^n$ are the domain samples and $y^s\in R$ are the range samples. Consider a piecewise affine mapping $\emptyset(\cdot):R^n\rightarrow R^m$ from the input space into the feature space.

$$\emptyset(\mathbf{x}) = \left[1 \;\; x_1 \cdots \; x_n \;\; |x_1 - \gamma_1^1| \; \cdots \; |x_1 - \gamma_1^{l_1}| \;\; |x_2 - \gamma_2^1| \cdots |x_2 - \gamma_2^{l_2}| \right. \\ \left. \cdots \; |x_n - \gamma_n^1| \cdots |x_n - \gamma_n^{l_n}| \right]^T \tag{5.11}$$

One can use the following PWA model for regression.

$$f(x) = \boldsymbol{w}^T \emptyset(\mathbf{x}^s) = \sum_{s=1}^{L} (\alpha_s - \alpha_s') \, \mathrm{K}(\mathbf{x}^s, \mathbf{x}) \tag{5.12}$$

Where, the PWA kernel $\mathrm{K}(\mathbf{x}^s, \mathbf{x})$ can be obtained as in (5.13) by using the mapping in (5.11).

$$\mathrm{K}(\mathbf{x}^s, \mathbf{x}) = \emptyset^{\mathrm{T}}(\mathbf{x}^s) \cdot \emptyset(\mathbf{x})$$

$$= \begin{bmatrix} 1 \\ x_1^s \\ \vdots \\ x_n^s \\ |x_1^s - \gamma_1^1| \\ \vdots \\ |x_1^s - \gamma_1^{l_1}| \\ |x_2^s - \gamma_2^1| \\ \vdots \\ |x_2^s - \gamma_2^{l_2}| \\ \vdots \\ |x_n^s - \gamma_n^1| \\ \vdots \\ |x_n^s - \gamma_n^{l_n}| \end{bmatrix}^T \cdot \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \\ |x_1 - \gamma_1^1| \\ \vdots \\ |x_1 - \gamma_1^{l_1}| \\ |x_2 - \gamma_2^1| \\ \vdots \\ |x_2 - \gamma_2^{l_2}| \\ \vdots \\ |x_n - \gamma_n^1| \\ \vdots \\ |x_n - \gamma_n^{l_n}| \end{bmatrix}$$

$$= \left(1 + x_1^s x_1 + \cdots + x_n^s x_n + |x_1^s - \gamma_1^1| \cdot |x_1 - \gamma_1^1| + \cdots \right. \\ + |x_1^s - \gamma_1^{l_1}| \cdot |x_1 - \gamma_1^{l_1}| + \cdots + |x_2^s - \gamma_2^1| \cdot |x_2 - \gamma_2^1| + \cdots \\ + |x_2^s - \gamma_2^{l_2}| \cdot |x_2 - \gamma_2^{l_2}| + \cdots + |x_n^s - \gamma_n^1| \cdot |x_n - \gamma_n^1| + \cdots \\ \left. + |x_n^s - \gamma_n^{l_n}| \cdot |x_n - \gamma_n^{l_n}| \right) \tag{5.13}$$

Substuting (5.13) into (5.12), one can obtain SVR with PWA kernel for having lattice structure.

$$f(x) = \sum_{s=1}^{L}(\alpha_s - \alpha'_s)\left(1 + x_1^s x_1 + \cdots + x_n^s x_n + |x_1^s - \gamma_1^1| \cdot |x_1 - \gamma_1^1| + \cdots + \left|x_1^s - \gamma_1^{l_1}\right| \cdot \left|x_1 - \gamma_1^{l_1}\right| + \cdots \right.$$
$$\left. + |x_2^s - \gamma_2^1| \cdot |x_2 - \gamma_2^1| + \cdots + \left|x_2^s - \gamma_2^{l_2}\right| \cdot \left|x_2 - \gamma_2^{l_2}\right| + \cdots + |x_n^s - \gamma_n^1| \cdot |x_n - \gamma_n^1| + \cdots + \left|x_n^s - \gamma_n^{l_n}\right| \cdot \left|x_n - \gamma_n^{l_n}\right|\right) \tag{5.14}$$

Where, the function $f(x)$ is a PWA function whose parameters can be found in terms of the dual variables as the follows.

$$f(x) = \underbrace{\sum_{s=1}^{L}(\alpha_s - \alpha'_s)1}_{a_0} + \underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha'_s)(\mathbf{x}^s)^T\right]}_{\boldsymbol{a}_1^T}\mathbf{x} + \underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha'_s)\,|\mathbf{x}^s - \boldsymbol{\gamma}^1|^T\right]}_{\boldsymbol{b}_1^T}|\mathbf{x} - \boldsymbol{\gamma}^1| + \cdots + \underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha'_s)\,|\mathbf{x}^s - \boldsymbol{\gamma}^l|^T\right]}_{\boldsymbol{b}_l^T}|\mathbf{x} - \boldsymbol{\gamma}^l|$$

$$f(x) = a_0 + \boldsymbol{a}_1^T\mathbf{x} + \boldsymbol{b}_1^T|\mathbf{x} - \boldsymbol{\gamma}^1| + \boldsymbol{b}_2^T|\mathbf{x} - \boldsymbol{\gamma}^2| + \cdots + \boldsymbol{b}_l^T|\mathbf{x} - \boldsymbol{\gamma}^l| \tag{5.15}$$
$$= a_0 + \boldsymbol{a}_1^T\mathbf{x} + \sum_{j=1}^{l}\boldsymbol{b}_j|\mathbf{x} - \boldsymbol{\gamma}^j|$$

Where, $|\mathbf{x}| = [|x_1|\ |x_2| \cdots |x_n|]^T$

When $\boldsymbol{\gamma}^s = \mathbf{x}^s$ for all $s \in \{1, \ldots, L\}$ with $l = L$, the function in (5.15) can be represented as in the following.

$$f(x) = a_0 + \boldsymbol{a}_1^T\mathbf{x} + \sum_{s=1}^{L}\boldsymbol{b}_s^T|\mathbf{x} - \mathbf{x}^s| \tag{5.16}$$

5.3 n-dimensional PWA-SVR models (general partition case)

Let a set of samples $\{(\mathbf{x}^s, y^s)\}_{s=1}^{L}$ be given. Where, $\mathbf{x}^s \in R^n$ are the domain samples and $y^s \in R$ are the range samples. Consider the following piecewise affine function $\boldsymbol{\Phi}(\cdot): R^n \to R^m$ for mapping the input space into the feature space.

$$\boldsymbol{\Phi}(\mathbf{x}) = [1 \;\; \mathbf{x}^{\mathrm{T}} \;\; |\boldsymbol{\alpha}_1^T\mathbf{x} - \boldsymbol{\gamma}^1| \;\; |\boldsymbol{\alpha}_2^T\mathbf{x} - \boldsymbol{\gamma}^2| \;\; \cdots |\boldsymbol{\alpha}_l^T\mathbf{x} - \boldsymbol{\gamma}^l|]^T \tag{5.17}$$

One can use the following PWA model for regression.

$$f(x) = \boldsymbol{w}^T\boldsymbol{\Phi}(\mathbf{x}) = \sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\mathrm{K}(\mathbf{x}^s, \mathbf{x}) \tag{5.18}$$

Where, the PWA kernel $\mathrm{K}(\mathbf{x}^s, \mathbf{x})$ can be obtained as in (5.19) by using the mapping in (5.17).

$$\begin{aligned}
\mathrm{K}(\mathbf{x}^s, \mathbf{x}) &= \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{x}^s) \cdot \boldsymbol{\Phi}(\mathbf{x}) \\
&= [1 \;\; (\mathbf{x}^s)^T \;\; |\boldsymbol{\alpha}_1^T\mathbf{x}^s - \gamma^1| \;\; |\boldsymbol{\alpha}_2^T\mathbf{x}^s - \gamma^2| \cdots \; |\boldsymbol{\alpha}_l^T\mathbf{x}^s - \gamma^l|] \begin{bmatrix} 1 \\ \mathbf{x} \\ |\boldsymbol{\alpha}_1^T\mathbf{x} - \gamma^1| \\ |\boldsymbol{\alpha}_2^T\mathbf{x} - \gamma^2| \\ \vdots \\ |\boldsymbol{\alpha}_l^T\mathbf{x} - \gamma^l| \end{bmatrix} \\
&= [1 + (\mathbf{x}^s)^T \cdot \mathbf{x} + |\boldsymbol{\alpha}_1^T\mathbf{x}^s - \gamma^1| \cdot |\boldsymbol{\alpha}_1^T\mathbf{x} - \gamma^1| \\
&\quad + |\boldsymbol{\alpha}_2^T\mathbf{x}^s - \gamma^2| \cdot |\boldsymbol{\alpha}_2^T\mathbf{x} - \gamma^2| + \cdots \\
&\quad + |\boldsymbol{\alpha}_l^T\mathbf{x}^s - \gamma^l| \cdot |\boldsymbol{\alpha}_l^T\mathbf{x} - \gamma^l|]
\end{aligned} \tag{5.19}$$

Substuting (5.19) into (5.18), one can obtain SVR with PWA kernel for having n dimensional general structure.

$$\begin{aligned}
f(\mathbf{x}) = \sum_{s=1}^{L}(\alpha_s - \alpha_s')\,[1 + (\mathbf{x}^s)^T \cdot \mathbf{x} + |\boldsymbol{\alpha}_1^T\mathbf{x}^s - \gamma^1|^T \cdot |\boldsymbol{\alpha}_1^T\mathbf{x} - \gamma^1| \\
+ |\boldsymbol{\alpha}_2^T\mathbf{x}^s - \gamma^2|^T \cdot |\boldsymbol{\alpha}_2^T\mathbf{x} - \gamma^2| + \cdots + |\boldsymbol{\alpha}_l^T\mathbf{x}^s - \gamma^l|^T \cdot |\boldsymbol{\alpha}_l^T\mathbf{x} - \gamma^l|]
\end{aligned} \tag{5.20}$$

Where, the function $f(x)$ is a PWA function whose coefficients can be found in terms of the dual variables as the follows.

$$f(x) = \sum_{s=1}^{L}(\alpha_s - \alpha_s')\,[1 + (\mathbf{x}^s)^T \cdot \mathbf{x} + |\boldsymbol{\alpha}_1^T \mathbf{x}^s - \gamma^1|^T \cdot |\boldsymbol{\alpha}_1^T \mathbf{x} - \gamma^1| + |\boldsymbol{\alpha}_2^T \mathbf{x}^s - \gamma^2|^T \cdot |\boldsymbol{\alpha}_2^T \mathbf{x} - \gamma^2| + \cdots + |\boldsymbol{\alpha}_l^T \mathbf{x}^s - \gamma^l|^T \cdot |\boldsymbol{\alpha}_l^T \mathbf{x} - \gamma^l|]$$

$$= \underbrace{\sum_{s=1}^{L}(\alpha_s - \alpha_s')}_{a_0} + \underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\mathbf{x}^{s^T}\right]}_{\boldsymbol{a}_1^T}\mathbf{x} + \underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,|\boldsymbol{\alpha}_1^T \mathbf{x}^s - \gamma^1|^T\right]}_{\boldsymbol{b}_1{}^T}|\boldsymbol{\alpha}_1^T \mathbf{x} - \gamma^1| + \cdots + \underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\,|\boldsymbol{\alpha}_l^T \mathbf{x}^s - \gamma^l|^T\right]}_{\boldsymbol{b}_l{}^T}|\boldsymbol{\alpha}_l^T \mathbf{x} - \gamma^l|$$

$$= a_0 + \boldsymbol{a}_1^T \mathbf{x} + \boldsymbol{b}_1{}^T|\boldsymbol{\alpha}_1^T \mathbf{x} - \gamma^1| + \boldsymbol{b}_2{}^T|\boldsymbol{\alpha}_2^T \mathbf{x} - \gamma^2| + \cdots + \boldsymbol{b}_l{}^T|\boldsymbol{\alpha}_l^T \mathbf{x} - \gamma^l|$$

$$= a_0 + \boldsymbol{a}_1^T \mathbf{x} + \sum_{j=1}^{l} \boldsymbol{b}_j{}^T|\boldsymbol{\alpha}_{\boldsymbol{j}}^{\boldsymbol{T}} \mathbf{x} - \gamma^j| \tag{5.21}$$

When $\gamma^j = (\mathbf{x}^s)^T \mathbf{x}^s$ and $\boldsymbol{\alpha}_{\boldsymbol{s}} = \mathbf{x}^s$ for all $s \in \{1, \ldots, L\}$ with $l = L$, the function in (5.21) can be represented as in the following.

$$f(x) = a_0 + \boldsymbol{a}_1^T \mathbf{x} + \sum_{s=1}^{L} \boldsymbol{b}_s{}^T|(\mathbf{x}^s)^T \mathbf{x} - (\mathbf{x}^s)^T \mathbf{x}^s| \tag{5.22}$$

The PWA kernel $\mathrm{K}(\mathbf{x}^s, \mathbf{x})$ can be written as in (5.23) by using (5.19)

$$\begin{aligned}\mathrm{K}(\mathbf{x}^s,\mathbf{x}) &= \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{x}^s)\cdot\boldsymbol{\Phi}(\mathbf{x})\\ &= [1+(\mathbf{x}^1)^T\cdot\mathbf{x}+|(\mathbf{x}^1)^T\mathbf{x}^s-(\mathbf{x}^1)^T\mathbf{x}^1|^T\cdot|(\mathbf{x}^1)^T\mathbf{x}-(\mathbf{x}^1)^T\mathbf{x}^1|\\ &\qquad +|(\mathbf{x}^2)^T\mathbf{x}^s-(\mathbf{x}^2)^T\mathbf{x}^2|^T\cdot|(\mathbf{x}^2)^T\mathbf{x}-(\mathbf{x}^2)^T\mathbf{x}^2|+\cdots\\ &\qquad +|(\mathbf{x}^l)^T\mathbf{x}^s-(\mathbf{x}^l)^T\mathbf{x}^l|^T\cdot\;|(\mathbf{x}^l)^T\mathbf{x}-(\mathbf{x}^l)^T\mathbf{x}^l|]\end{aligned} \tag{5.23}$$

5.4 First degree B-spline PWA-SVR model

Let a set of samples $\{(x^s, y^s)\}_{s=1}^{L}$ be given. Where, $x^s \in R^n$ are the domain samples and $y^s \in R$ are the range samples. Consider a piecewise affine function $\emptyset(\cdot): R \to R^l$ mapping from the input space into the feature space.

$$\emptyset(x) = \begin{bmatrix} -\frac{1}{2}|x-\gamma^1|+\frac{1}{2}|x-\gamma^2|+\frac{1}{2}sign(x-\gamma^1)\\ \frac{1}{2}|x-\gamma^1|-|x-\gamma^2|+\frac{1}{2}|x-\gamma^3|\\ \frac{1}{2}|x-\gamma^2|-|x-\gamma^3|+\frac{1}{2}|x-\gamma^4|\\ \vdots\\ \frac{1}{2}|x-\gamma^{l-2}|-|x-\gamma^{l-1}|+\frac{1}{2}|x-\gamma^l|\\ \frac{1}{2}|x-\gamma^{l-1}|-\frac{1}{2}|x-\gamma^l|-\frac{1}{2}sign(x-\gamma^{l-1})\end{bmatrix} \tag{5.24}$$

Then, the mapped data set becomes as:

$$\{(\emptyset(x^s), y^s)\}_{s=1}^{L}, \qquad \emptyset(x^s) \in R^l, \qquad y^s \in R \tag{5.25}$$

So, the regression function is obtained as in the following.

$$f(x) = \boldsymbol{w}^T\emptyset(\mathbf{x}^s) + b = \sum_{s=1}^{L}(\alpha_s - \alpha_s')\,\mathrm{K}(\mathbf{x}^s,\mathbf{x}) + b \tag{5.26}$$

As mentioned in the previous subsection, one can use the following PWA kernel function to obtain PWA-SVR model in terms of first order B-splines.

$$\mathrm{K}(x^s, x) = \emptyset^T(x^s) \cdot \emptyset(x)$$

$$= \begin{bmatrix} -\frac{1}{2}|x^s-\gamma^1| + \frac{1}{2}|x^s-\gamma^2| + \frac{1}{2}sign(x^s-\gamma^1) \\ \frac{1}{2}|x^s-\gamma^1| - |x^s-\gamma^2| + \frac{1}{2}|x^s-\gamma^3| \\ \frac{1}{2}|x^s-\gamma^2| - |x^s-\gamma^3| + \frac{1}{2}|x^s-\gamma^4| \\ \vdots \\ \frac{1}{2}|x^s-\gamma^{l-2}| - |x^s-\gamma^{l-1}| + \frac{1}{2}|x^s-\gamma^l| \\ \frac{1}{2}|x^s-\gamma^{l-1}| - \frac{1}{2}|x^s-\gamma^l| - \frac{1}{2}sign(x^s-\gamma^{l-1}) \end{bmatrix}^T \cdot \begin{bmatrix} -\frac{1}{2}|x-\gamma^1| + \frac{1}{2}|x-\gamma^2| + \frac{1}{2}sign(x-\gamma^1) \\ \frac{1}{2}|x-\gamma^1| - |x-\gamma^2| + \frac{1}{2}|x-\gamma^3| \\ \frac{1}{2}|x-\gamma^2| - |x-\gamma^3| + \frac{1}{2}|x-\gamma^4| \\ \vdots \\ \frac{1}{2}|x-\gamma^{l-2}| - |x-\gamma^{l-1}| + \frac{1}{2}|x-\gamma^l| \\ \frac{1}{2}|x-\gamma^{l-1}| - \frac{1}{2}|x-\gamma^l| - \frac{1}{2}sign(x-\gamma^{l-1}) \end{bmatrix} \tag{5.27}$$

$$\begin{aligned} &= \left(-\frac{1}{2}|x^s-\gamma^1| + \frac{1}{2}|x^s-\gamma^2| + \frac{1}{2}sign(x^s-\gamma^1)\right) \cdot \left(-\frac{1}{2}|x-\gamma^1| + \frac{1}{2}|x-\gamma^2| + \frac{1}{2}sign(x-\gamma^1)\right) \\ &\quad + \left(\frac{1}{2}|x^s-\gamma^1| - |x^s-\gamma^2| + \frac{1}{2}|x^s-\gamma^3|\right) \cdot \left(\frac{1}{2}|x-\gamma^1| - |x-\gamma^2| + \frac{1}{2}|x-\gamma^3|\right) \\ &\quad + \left(\frac{1}{2}|x^s-\gamma^2| - |x^s-\gamma^3| + \frac{1}{2}|x^s-\gamma^4|\right) \cdot \left(\frac{1}{2}|x-\gamma^2| - |x-\gamma^3| + \frac{1}{2}|x-\gamma^4|\right) \\ &\quad + \cdots + \left(\frac{1}{2}|x^s-\gamma^{l-2}| - |x^s-\gamma^{l-1}| + \frac{1}{2}|x^s-\gamma^l|\right) \cdot \left(\frac{1}{2}|x-\gamma^{l-2}| - |x-\gamma^{l-1}| + \frac{1}{2}|x-\gamma^l|\right) \\ &\quad + \left(\frac{1}{2}|x^s-\gamma^{l-1}| - \frac{1}{2}|x^s-\gamma^l| - \frac{1}{2}sign(x^s-\gamma^{l-1})\right) \cdot \left(\frac{1}{2}|x-\gamma^{l-1}| - \frac{1}{2}|x-\gamma^l| - \frac{1}{2}sign(x-\gamma^{l-1})\right) \end{aligned}$$

Substituting (5.27) into (5.26), one can obtain first degree B-spline PWA-SVR model.

$$
\begin{aligned}
f(x) &= \boldsymbol{w}^T \emptyset(\mathbf{x}^s) + b \\
&= \sum_{s=1}^{L} (\alpha_s - \alpha_s') \Bigg[-\frac{1}{2}|x^s - \gamma^1| + \frac{1}{2}|x^s - \gamma^2| + \frac{1}{2} sign(x^s - \gamma^1) \cdot \left(-\frac{1}{2}|x - \gamma^1| + \frac{1}{2}|x - \gamma^2| + \frac{1}{2} sign(x - \gamma^1) \right) \\
&\qquad + \left(\frac{1}{2}|x^s - \gamma^1| - |x^s - \gamma^2| + \frac{1}{2}|x^s - \gamma^3| \right) \cdot \left(\frac{1}{2}|x - \gamma^1| - |x - \gamma^2| + \frac{1}{2}|x - \gamma^3| \right) \\
&\qquad + \left(\frac{1}{2}|x^s - \gamma^2| - |x^s - \gamma^3| + \frac{1}{2}|x^s - \gamma^4| \right) \cdot \left(\frac{1}{2}|x - \gamma^2| - |x - \gamma^3| + \frac{1}{2}|x - \gamma^4| \right) + \cdots \\
&\qquad + \left(\frac{1}{2}|x^s - \gamma^{l-2}| - |x^s - \gamma^{l-1}| + \frac{1}{2}|x^s - \gamma^l| \right) \cdot \left(\frac{1}{2}|x - \gamma^{l-2}| - |x - \gamma^{l-1}| + \frac{1}{2}|x - \gamma^l| \right) \\
&\qquad + \left(\frac{1}{2}|x^s - \gamma^{l-1}| - \frac{1}{2}|x^s - \gamma^l| - \frac{1}{2} sign(x^s - \gamma^{l-1}) \right) \cdot \left(\frac{1}{2}|x - \gamma^{l-1}| - \frac{1}{2}|x - \gamma^l| - \frac{1}{2} sign(x - \gamma^{l-1}) \right) \Bigg] + b
\end{aligned}
\tag{5.28}
$$

Where, the function $f(x)$ is a PWA function whose parameters can be found in terms of the dual variables as follows.

$$
\begin{aligned}
f(x) = \underbrace{b}_{a_0} &+ \underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\left(-\frac{1}{2}|x^s - \gamma^1| + \frac{1}{2}|x^s - \gamma^2| + \frac{1}{2}sign(x^s - \gamma^1)\right)\right]}_{c_1}\frac{1}{2}sign(x - \gamma^1) \\
&+ \underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\left(-\frac{1}{2}|x^s - \gamma^{l-1}| + \frac{1}{2}|x^s - \gamma^l| + \frac{1}{2}sign(x^s - \gamma^{l-1})\right)\right]}_{c_2}\frac{1}{2}sign(x - \gamma^{l-1}) \\
&+ \underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\left(|x^s - \gamma^1| - |x^s - \gamma^2| + +\frac{1}{2}|x^s - \gamma^3| - \frac{1}{2}sign(x^s - \gamma^1)\right)\right]}_{b_1}|x - \gamma^1| \\
&+ \frac{3}{2}\underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\left(-|x^s - \gamma^1| + \frac{3}{2}|x^s - \gamma^2| - \frac{1}{2}|x^s - \gamma^3| + \frac{1}{2}sign(x^s - \gamma^1)\right)\right]}_{b_2}|x - \gamma^2| \\
&+ \frac{3}{2}\underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\left(|x^s - \gamma^1| - \frac{3}{2}|x^s - \gamma^2| + \frac{3}{2}|x^s - \gamma^3| - \frac{1}{2}|x^s - \gamma^4|\right)\right]}_{b_3}|x - \gamma^3| + \cdots \\
&+ \frac{3}{2}\underbrace{\left[\sum_{s=1}^{L}(\alpha_s - \alpha_s')\left(\frac{1}{2}|x - \gamma^{l-2}| - \frac{3}{2}|x^s - \gamma^{l-1}| + |x^s - \gamma^l| + \frac{1}{2}sign(x^s - \gamma^{l-1})\right)\right]}_{b_l}|x - x^l|
\end{aligned}
$$

$$f(x) = a_o + c_1 \left[\frac{1}{2} sign(x - \gamma^1)\right] + c_2 \left[\frac{1}{2} sign(x - \gamma^{l-1})\right] + b_1|x - \gamma^1| + b_2|x - \gamma^2| + \cdots + b_l|x - \gamma^l|$$

$$= a_o + \left[\frac{c_1}{2} sign(x - \gamma^1) + \frac{c_2}{2} sign(x - \gamma^{l-1})\right] + \sum_{j=1}^{l} b_j|x - \gamma^j| \qquad (5.29)$$

When $\gamma^j = x^s$ for all $s \in \{1, \dots, L\}$ with $l = L$, the function in (5.29) can be represented as the following PWA canonical representation.

$$f(x) = a_o + \left[\frac{c_1}{2} sign(x - \gamma^1) + \frac{c_2}{2} sign(x - \gamma^{l-1})\right] + \sum_{s=1}^{L} b_s|x - x^s| \qquad (5.30)$$

However, the above kernel does not have a compact representation. The following function is also proposed in the thesis as a more compact kernel.

$$\mathrm{K}(x^s, x) = \max\left\{0,\ p^1 - \frac{p^1}{p^2}|x^s - x|\right\} \qquad (5.31)$$

Where, $p^1$and $p^2$are user specified parameters which serve as the amplitude and, in some sense, variance of kernel.

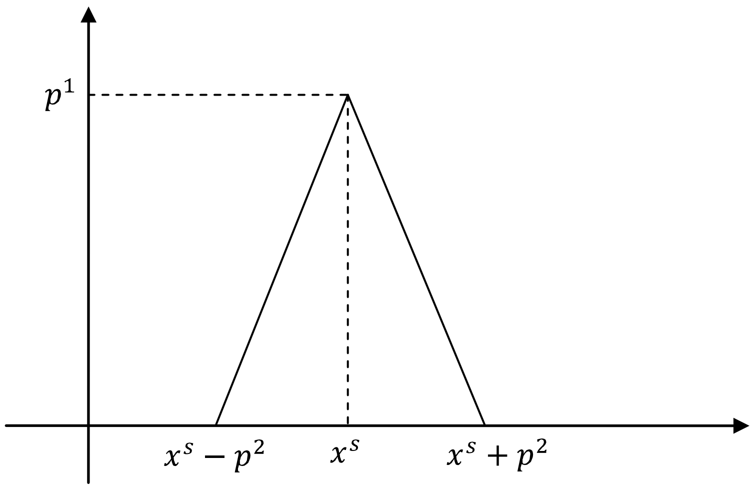


Figure 5.1 Compact PWA kernel

5.5 A quantitative analysis of the developed PWA regression models

This subsection presents performance analysis results of the developed PWA regression models. $\text{sinc}(x) = \frac{\sin(x)}{x}$ function is taken as the test function for all examined models. Table 5.1 and Table 5.2 present the regression performances of the SVR models with the developed PWA kernels in comparison with the Gaussian kernel for $C = 0.1$ and, respectively, for $C = 10$.

Table 5.1 A comparison of Gaussian kernel and the developed PWA kernels for sinc function in ε-insensitive SVR model with $C = 0.1$.

ε-insensitive SVR $C = 0.1$		Percentage of Root mean square Difference (PRD)			Root Mean Square Error (RMSE)			(Norm of w)			CPU time (In second)		
	Epsilon	0	0.01	0.1	0	0.01	0.1	0	0.01	0.1	0	0.0 1	0.1
Kernels													
Gaussian	Train	30.270	24.794	21.036	0.105	0.083	0.071	2.776	2.682	1.865	1.2	1.2	1.2
	Test	25.921	28.738	17.288	0.111	0.121	0.072						
	# of SV & (%)	12 (19.7%)	12 (19.7%)	11 (18%)									
Canonical PWA kernel in (5.7)	Train	12.113	12.94	21.756	0.042	0.047	0.071	0.040	0.039	0.024	1.3	1.3	1.3
	Test	11.729	10.548	19.185	0.050	0.047	0.078						
	# of SV & (%)	9 (14.8%)	10 (10.4%)	10 (16.4%)									
Max $\{0, p^1 - \frac{p^1}{p^2}\lvert x^s - x\rvert\}$	Train	4.827	4.888	24.145	0.019	0.018	0.095	0.854	0.788	0.537	1.8	1.0	1.1
	Test	5.552	9.945	20.058	0.028	0.045	0.103						
	# of SV & (%)	53 (86.9%)	51 (83.6%)	44 (72.1%)									

Table 5.2 A comparison of Gaussian kernel and the developed PWA kernels for sinc function in ε-insensitive SVR model with $C = 10$.

ε-insensitive SVR $C = 10$	Percentage of Root mean square Difference (PRD)				Root Mean Square Error (RMSE)			Norm of w			CPU time (In second)		
	Epsilon	0	0.01	0.1	0	0.01	0.1	0	0.01	0.1	0	0.01	0.1
Kernels													
Gaussian	Train	127.254	128.281	62.514	0.447	0.415	0.206	12.066	11.912	7.288	1.1	1.1	1.2
	Test	96.448	96.113	50.816	0.420	0.384	0.207						
	# of SV & (%)	10 (16.4%)	11 (18%)	6 (9.8%)									
Canonical PWA kernel in (5.7)	Train	132.704	13.966	19.128	0.504	0.048	0.066	0.0257	0.0392	0.0255	1.2	1.3	1.2
	Test	122.491	14.046	20.360	0.580	0.060	0.089						
	# of SV & (%)	6 (9.8%)	7 (11.5%)	11 (18%)									
Max $\{0, p^1 - \frac{p^1}{p^2}\lvert x^s - x\rvert\}$	Train	2.125	2.571	24.734	0.007	0.009	0.095	0.847	0.804	0.530	1.7	1.0	1.0
	Test	6.931	11.749	22.640	0.029	0.052	0.112						
	# of SV & (%)	47 (77.0%)	45 (73.8%)	44 (72 %)									

Table 5.3 and 5.4 presents the regression performances of the LS-SVR models with the developed PWA kernels in comparison with the Gaussian kernels for $C = 1$ and, respectively, for $C = 10$.

Table 5.3 A comparison of Gaussian kernel and the developed PWA kernels for sinc function in LS-SVR model with $C = 1$.

LS-SVR $C = 1$		Percentage of root mean square difference (PRD)	Rroot Mean Square Error (RMSE)	Norm of w
Kernel				
Gaussian	Train	27.417	0.109	1.196
	Test	21.975	0.108	
Canonical PWA kernel in (5.7)	Train	1.829	0.006	0.045
	Test	4.510	0.018	
Max $\{0, p^1 - \frac{p^1}{p^2}\lvert x^s - x\rvert\}$	Train	9.909	0.036	0.684
	Test	13.402	0.062	

Table 5.4 A comparison of Gaussian kernel and the developed PWA kernels for sinc function in LS-SVR model with $C = 10$.

LS-SVR $C = 10$		Percentage of root mean square difference (PRD)	Rroot Mean Square Error (RMSE)	Norm of w
Kernel				
Gaussian	Train	7.525	0.026	2.920
	Test	6.632	0.029	
Canonical PWA kernel in (5.7)	Train	0.327	0.001	0.049
	Test	2.569	0.011	
Max $\{0, p^1 - \frac{p^1}{p^2}\lvert x^s - x\rvert\}$	Train	1.097	0.004	0.830
	Test	7.683	0.036	

CHAPTER SIX

INPUT-OUTPUT CLUSTERING BASED DESIGN OF APPROXIMATE PWA FUNCTIONS

This chapter presents a new method for the PWA function representation. The idea behind the proposed method is to employ clustering to the given domain-range sample data set $\{(x^s, y^s)\}_{s=1}^{L}$ to determine the nonempty regions partitioned by a finite number of hyperplanes, and then to determine the coefficients associated to each affine function by using the samples belonging to this specified region. The number of the partitioned regions is assumed known a priori. In order to obtain the parameters of each affine function valid for a specific region, two methods are proposed: The least square regression and orthogonal regression.

In the least square regression, the mean square of error between the range samples and the regression value is considered as the error to be minimized. In the orthogonal regression, the sum of distances of the samples to the linear regression is considered as the error.

6.1 PWA function

Let a set of finite data pairs $\{(\mathbf{x}^s, \mathbf{y}^s)\}_{s=0}^{L}$, $\mathbf{x}^s \in R^n, \mathbf{y}^s \in R^m$ be given. As explained in Section 2.1.5.1. the PWA functions, which can be represented by the following canonical representation under mild conditions, can be used to approximate the given sample set.

$$f(\mathbf{x}) = a + A\mathbf{x} + \sum_{j=1}^{l} b_j \left| \boldsymbol{\alpha}_j^T \mathbf{x} - \gamma_j \right| \tag{6.1}$$

with $a, b_j \in R^m$, $\boldsymbol{\alpha_j} \in R^n$, $A \in R^{m x n}$ and $\gamma_j \in R^1$ if and only if it satisfies the consistent variation property.

This chapter gives a method which is based input-output clustering for determining the parameters of such a PWA regression function. Input-output clustering will be presented firstly and then the application of this clustering method will be given for determining affine regression models defined in each affine region and the linear partition of the domain space.

6.2 Input-Output Clustering

In the Input-Output Clustering (IOC), the vectors to which a clustering algorithm will be applied, are obtained by augmenting the weighted input sample vectors with the desired outputs as in (6.2) (Uykan, et. al.,2000).

$$(\boldsymbol{t}^s)^{I/O} = [\gamma(\mathbf{x}^s)^T \; (\mathbf{y}^s)^T]^T \qquad (6.2)$$

Where, γ is a weighting factor.

In this thesis, the IOC algorithm applies a batch mode clustering algorithm to the set of augmented vectors in (6.2) and then find the augmented centers of the clusters $\boldsymbol{m^j} = [\gamma\left(m_x^j\right)^T \left(m_y^j\right)^T]^T \in R^{n+m}$ where $\gamma m_x^j, \in R^n$ and $m_y^j \in R^m$. The first m entries are rescaled with $1/\gamma$. One can choose any metric for quantization error to be minimized by clustering. In this thesis study, Euclidean norm for quantization error and K-means clustering algorithm is used for partitioning the data in order to determine the linear partition of the PWA regression function under construction.

6.3 PWA representation by using input-output clustering method

The key point in the design of piecewise affine function representation is to specify the number and the locations of the breakpoints for one-dimensional case and hyperplanes partitioning the domain space for n-dimensional case. Once the locations of breakpoints and, in general, hyperplanes are determined, the optimal local regression function affine can be determined by using one of the linear regression methods, for instance, any method calculating generalized inverse solution.

The proposed IOC based PWA regression method is described in the flowchart of Figure 6.1. K-means used for clustering is just one of the possible choices. Different quantization error norms can be exploited to determine the linear partition of the domain space.

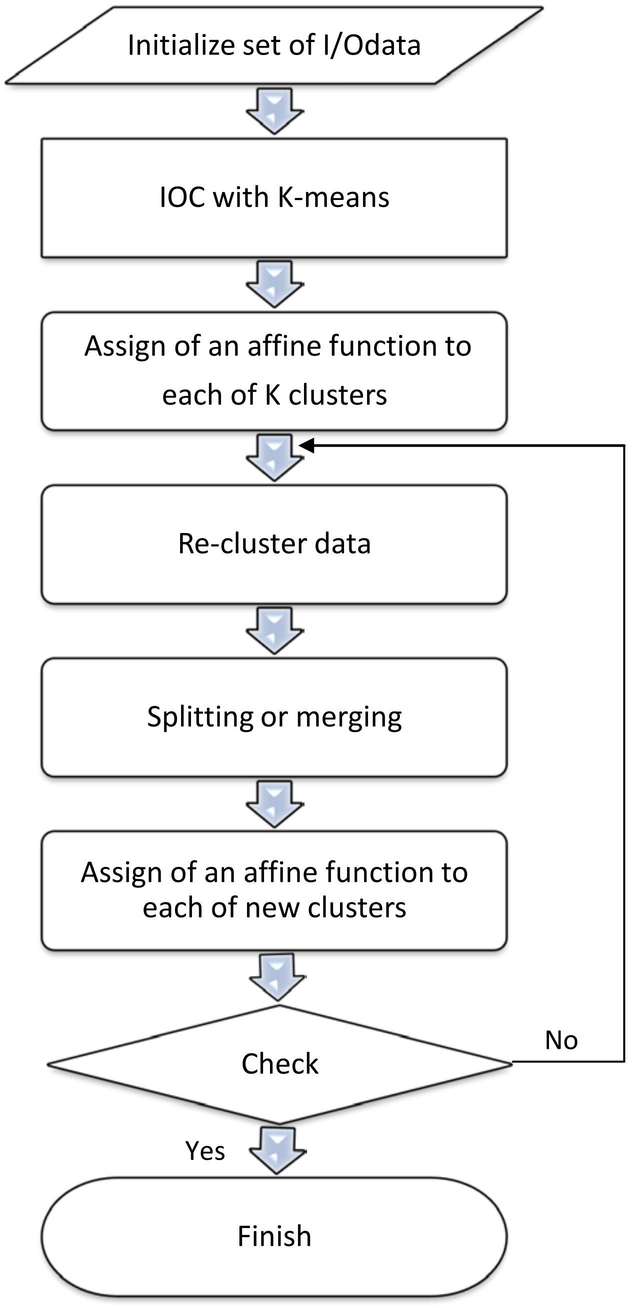

Initialize set of I/Odata
IOC with K-means
Assign of an affine function to each of K clusters
Re-cluster data
Splitting or merging
Assign of an affine function to each of new clusters
Check
No
Yes
Finish

Figure 6.1 Flowchart describing PWA function approximation with IOC

The steps of the proposed algorithm are given as follow.

I. The given set of finite data pairs $\{(x^s, y^s)\}_{s=0}^{L}$ is clustered for a given initial cluster numbers K by using IOC with K-means clustering algorithm.

II. After the given sample data are clustered, to assign an affine function to each of K clusters, two methods are proposed in the thesis: one is the orthogonal regression and the other is the least square regression.

III. In this step, the whole data is again clustered iteratively in two different ways: Either orthogonal distance or penalized orthogonal distance between any individual input-output sample and each of affine functions are calculated, or then the sample is assigned to the closest line representing the associated cluster.

IV. Iteration is terminated when the error measure is satisfied or the specified maximum iteration number is reached.

The developed method will be described, in the sequel, for the one dimensional case, i.e. the function to be approximated is considered as $f(\cdot): R \rightarrow R$. Below, the regression methods used for each affine region will be given firstly, and then the metrics i.e. orthogonal distance and penalized orthogonal distance used in the clustering will be explained.

Orthogonal regression

Consider the affine function $f(x) = wx + b$ with $w, b \in R$. For orthogonal regression, the coefficients w and b are given by

$$w = \frac{\sum\left(y^s - \mu_y\right)^2 - \sum(x^s - \mu_x)^2}{2\sum(x^s - \mu_x)\left(y^s - \mu_y\right)} + \frac{\sqrt{\left(\sum\left(y^s - \mu_y\right)^2 - \sum(x^s - \mu_x)^2\right)^2 + 4\left[\sum(x^s - \mu_x)\left(y^s - \mu_y\right)\right]^2}}{2\sum(x^s - \mu_x)\left(y^s - \mu_y\right)}$$

$$b = \sum\left(y^s-\mu_y\right)^2 - \frac{\sum\left(y^s-\mu_y\right)^2-\sum(x^s-\mu_x)^2}{2\sum(\mathrm{x}^s-\mu_x)\left(y^s-\mu_y\right)}\sum(x^s-\mu_x)^2 - \frac{\sqrt{\left(\sum\left(y^s-\mu_y\right)^2-\sum(x^s-\mu_x)^2\right)^2+4\left[2\sum(\mathrm{x}^s-\mu_x)\left(y^s-\mu_y\right)\right]^2}}{2\sum(\mathrm{x}^s-\mu_x)\left(y^s-\mu_y\right)}\sum(x^s-\mu_x)^2$$

Where, μ_x and μ_y are the mean value of x and y, respectively.

Least square regression:

The coefficients w and b are found by the following linear equation system.

$$\begin{bmatrix} L & \sum_{i=1}^{L} x^s \\ \sum_{s=1}^{L} x^s & \sum_{i=1}^{L} (x^s)^2 \end{bmatrix} \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{L} y^s \\ \sum_{i=1}^{L} y^s x^s \end{bmatrix}$$

Orthogonal distance

Take two points (end and last) on the i'<u>th</u> affine function $P_i^1 = (x_i^1, y_i^1)$ and $P_i^2 = (x_i^2, y_i^2)$, and a data point as $P^s = (x^s, y^s)$, then the orthogonal distance is given by the following formula:

$$d_i^s = \left|\frac{(x_i^2-x_i^1)(y_i^1-y^s)-(x_i^1-x^s)(y_i^2-y_i^1)}{\sqrt{(x_i^2-x_i^1)^2+(y_i^2-y_i^1)^2}}\right|$$

Penalized orthogonal distance

For penalized orthogonal distance, penalty parameter λ defined by the user is added to the formula given for orthogonal distance.

$$d_i^s = \frac{(x_i^2 - x_i^1)(y_i^1 - y^s) - (x^1 - x^s)(y_i^2 - y_i^1)}{\sqrt{(x_i^2 - x_i^1)^2 + (y_i^2 - y_i^1)^2}} + \lambda\begin{bmatrix} y^s - \mu_x^i & y^s - \mu_y^i \end{bmatrix}\begin{bmatrix} x^s - \mu_x^i \\ y^s - \mu_y^i \end{bmatrix}$$

The cost function for penalized orthogonal distance is defined as:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{K} \sum_{\begin{bmatrix} x^j \\ y^j \end{bmatrix} \in S_i} \|d_i^s\| + \lambda \left[\begin{bmatrix} x^j \\ y^j \end{bmatrix} - \mu^i\right]^T \left[\begin{bmatrix} x^j \\ y^j \end{bmatrix} - \mu^i\right] \text{ for } i \in \{1,2,\cdots,K\}$$

Where, S_i is the i'th cluster, and d_i^s is the orthogonal distance between (x^s, y^s) sample and the line belonging to the i'th cluster, and K is the number of cluster. For non penalized orthogonal distance, the penalty term λ disappears.

Note that there should exist at least two data points in each cluster. If the number of data is less than two within a cluster, that cluster is eliminated and is merged to the closest cluster. There are some clusters that may overlap in the input space (See Figure 6.2 for the application of the method on ECG data.). In this case, the partition where the overlapping occurs is labeled as a new cluster and also the rest of partition(s) is labeled as new cluster(s). Then, the affine functions are assigned to the new labeled clusters (See Figure 6.3 for ECG signal approximation.). Thus, the new affine functions assigned locally defined. That is, affine functions assigned for each clusters are not allowed to be overlapped in this algorithm.

By the above method, several affine functions each of which is defined locally in a bounded polyhedral region are obtained (See Figure 6.4 for one dimensional case where the regions are bounded intervals.). To obtain a continuous canonical PWA function, first and end points of these affine functions are used as breakpoints (See Figure 6.5). Coefficients of the PWA function can be found by using the generalized inverse solution. Finally, optimal continuous PWA function is obtained by minimizing the total squared error (See Figure 6.6).

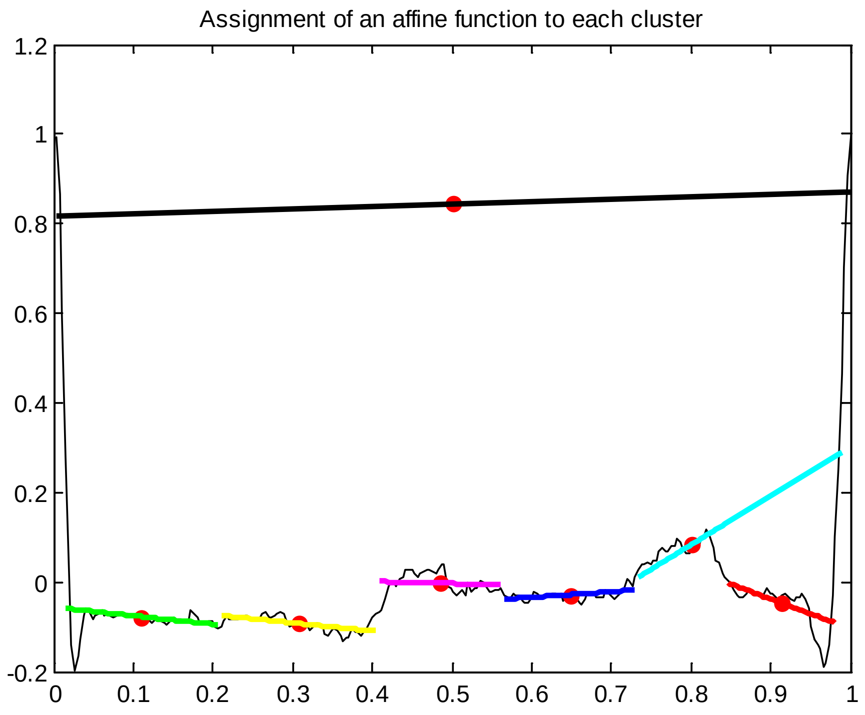


Figure 6.2 Clusters whose input space projections overlap and the associated affine regressors.

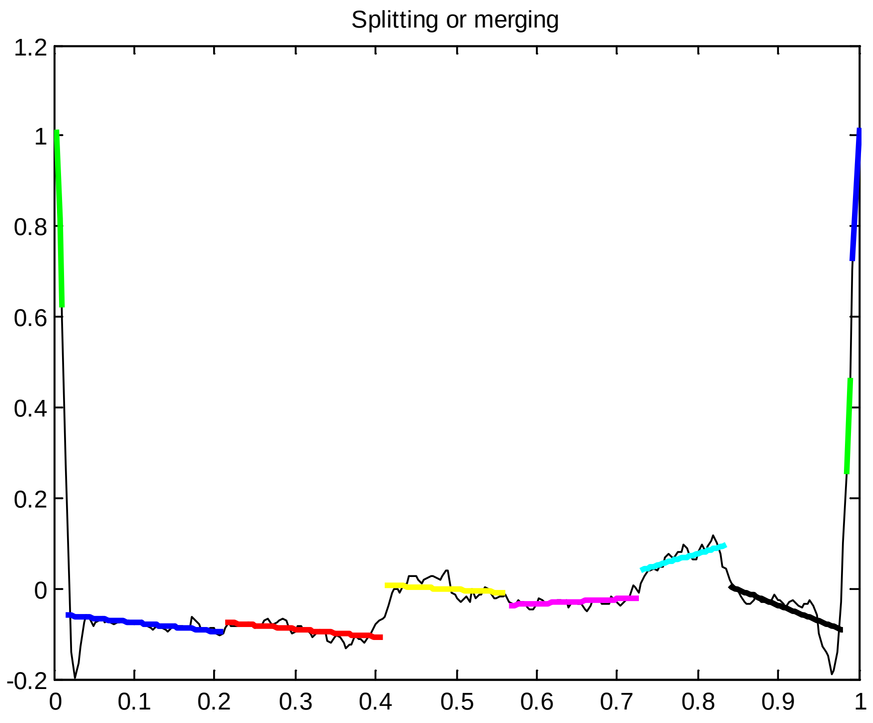


Figure 6.3 Obtaining the non-overlapping intervals by splitting or merging

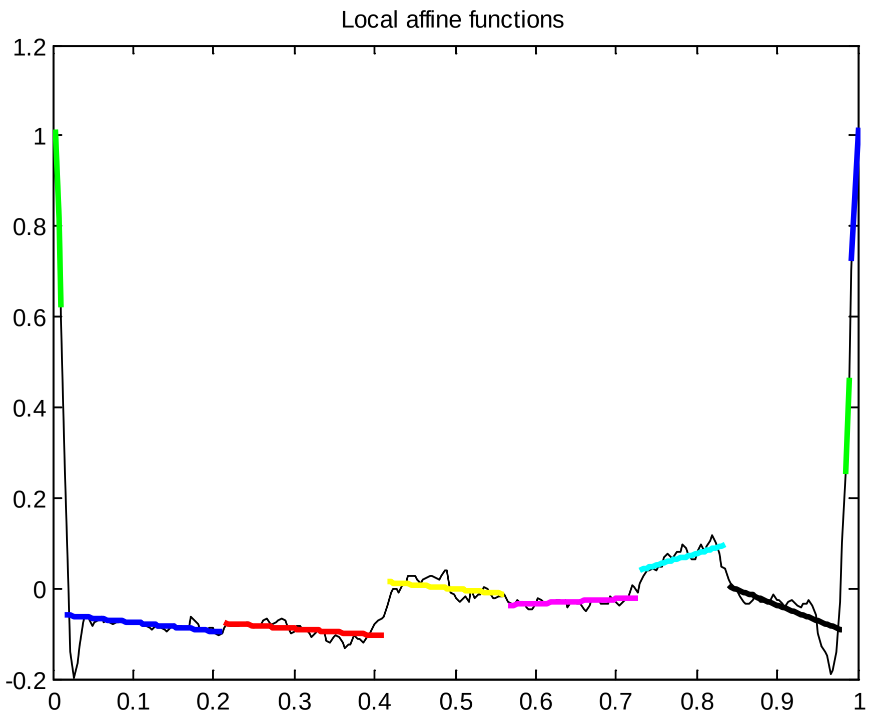


Figure 6.4 Reassigning data to the closest affine regressors and then redetermining the affine regressors associated to the non-overlapping intervals

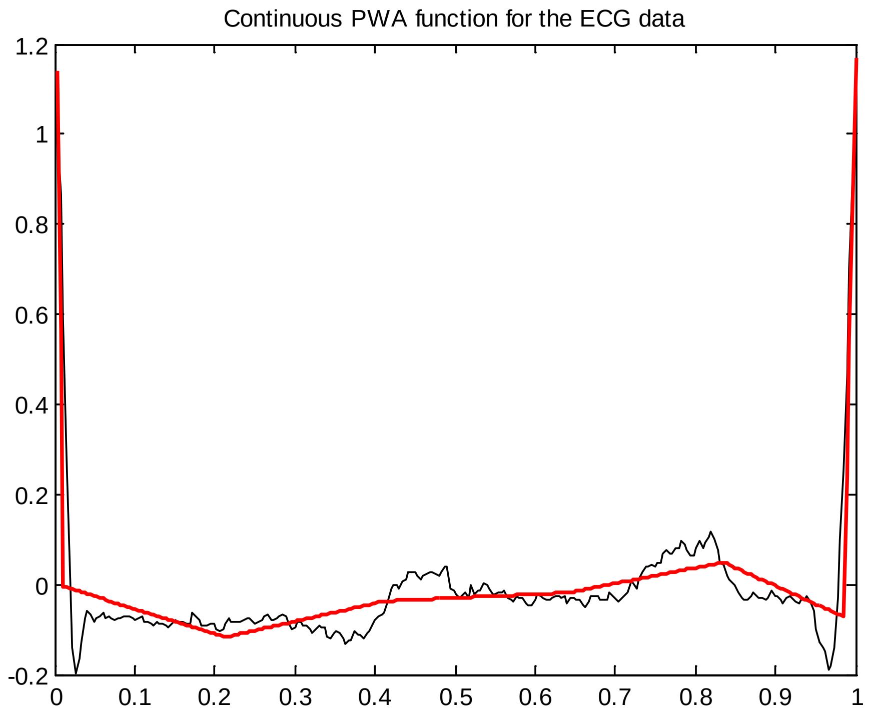


Figure 6.5 Determination of breakpoints of a continuous PWA regression by calculating the end points associated to the intervals defined by local affine regressors

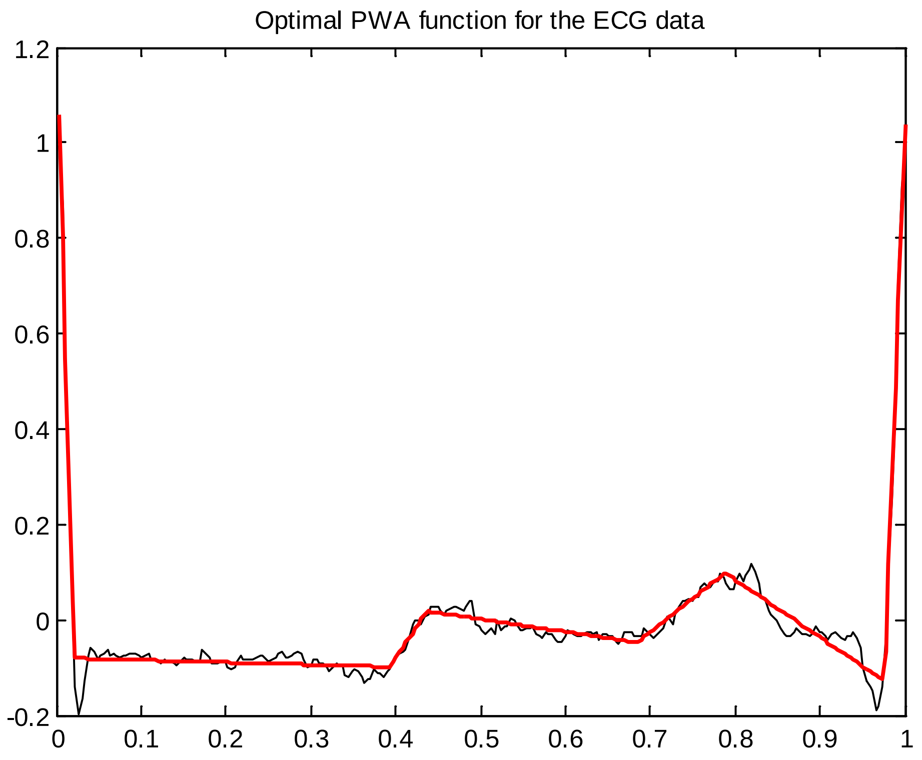


Figure 6.6 Construction of a continuous PWA function using the entire data by applying the least square PWA regression method for the obtained breakpoints.

CHAPTER SEVEN
CONCLUSION

In this thesis, four novel classes of regression models which are based on piecewise affine and/or support vector methods are proposed. The first class is the support vector regression models employing ℓ_p with $p \leq 1$ norms for model parameter cost in order to reduce model complexity and saturating and/or piecewise affine loss functions for rejection the contributions of outliers in determination of model parameters. The second class is the ε-insensitive least square support vector regression model class which is introduced as an extension of the least square support vector regression for reducing excessive number of support vectors appearing in the support vector approach. The third is the piecewise affine support vector regression model class which is derived by exploiting the canonical representations of piecewise affine functions and the first order B-spline basis functions. Finally, the piecewise affine models, which are designed by input-output clustering, are developed

The developed methods improve the performances of available support regression models in terms of the generalization ability and/or robustness against to outliers. The presented studies can be extended by future researches in two directions. More efficient optimization tools which fit better to the specific natures of the proposed models may be developed. The proposed regression models may be applied to new application domains such as signal compression and system identification.

REFERENCES

Aizerman, M. A., Braverman, E. M., & Rozonotr, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control, 25,* 821-837.

Bao, Y.K., Liu, Z.T., Guo, L., & Wang, W., (2005). *Forecasting STOCK composite index by fuzzy support vector machines regression*. In: Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, *6*, 18-21, 3535-3540.

Ben-Hur, A., Horn, D., Siegelmann, H.T., & Vapnik, V. (2001). Support vector clustering. *Journal of Machine Learning Research, 2,* 125-137.

Bergeron, C., Cheriet, F., Ronsky, J., Zernicke, R., & Labelle, H. (2005). Prediction of anterior scoliotic spinal curve from trunk surface using support vector regression. *Engineering Applications of Artificial Intelligence,18,* 8, 973-983.

Bertsekas, D. P. (1995). *Dynamic Programming and Optimal Control*, 1, Athena Scientific, Belmont, MA.

Bohr, H. (1947). *Almost Periodic Functions*. New York: Chelsea.

Burges, C.J.C. (1998). Geometry and invariance in kernel based methods. *In Advances in Kernel Methods - Support VectorLearning*, MIT Press, Cambridge, MA.

Chen, S. S. (1995). *Basis Pursuit*. Ph.D. Thesis, Department of Statistics, Stanford University, Stanford, CA, from http://www-stat.stanford.edu/˜atomizer/

Chen, S., Donoho, D., & Saunders, M. (1995). *Atomic decomposition by basis pursuit*. Technical Report 479, Department of Statistics, Stanford University.

Cheney, W., & Kincaid, D. (1985). *Numerical Mathematics and Computing* (2nd Ed.). U.S.A: Brooks.

Cherkassky, V., & Mulier, F.M. (2007). *Learning from Data: Concepts, Theory, and Methods*, Wiley-IEEE Press.

Chua, L. O., & Kang, S. (1977). Section-wise piecewise-linear functions: Canonical representation, properties and applications. *Proc. IEEE, 65*, 915–929.

Chua, L. O., & Deng, A. (1988). Canonical piecewise-linear representation. *IEEE Trans. Circuits Syst., 35,* 511–525.

Coifman, R.R. & Wickerhauser, M.V. (1992). Entropy-based algorithms for best-basis selection.IEEE Transactions on Information Theory, 38:713-718.

Colliez, J., Dufrenois, F., & Hamad, D. (2006). Optic flow estimation by support vector regression. *Engineering Applications of Artificial Intelligence, 19,* 7, 761-768.

Courant, R. and Hilbert, & D. (1953). *Methods of Mathematical Physics*. Interscience Publishers Inc., New York.

Cox, D. R. (1972). Regression Models and Life Tables (with Discussion). *Journal of the Royal Statistical Society*, Series B 34:187-220.

Cybenko, G. (1989). Approximation by superposition of a sigmoidal function. *Math. Control Systems Signals*, *2* (4): 303-314.

Daubechies, I. (1992). *Ten lectures on wavelets*. CBMS-NSF Regional Conferences Series in Applied Mathematics. SIAM, Philadelphia, PA,

De Boor, C. (1978). *A practical guid to splines*. New York, U.S.A.: *Springer-Verlag.*

Espinoza, M., Pelckmans, K., Hoegaerts, L., Suykens, J., & De Moor, B. (2004). *A comparative study of LS-SVMS applied to the silver box identification problem.* Symposium on nonlinear control systems (NOLCOS), 513–518, Stuttgart, Germany.

Espinoza, M., Suykens, J.A.K., & De Moor, B. (2005). Kernel Based Partially Linear Models and Nonlinear Identification. *IEEE Transactions on Automatic Control, 50*, 10.

Espinoza, M., Suykens, J.A.K., & De Moor, B. (2005). *Load forecasting using fixed-size least squares support vector machines*. 8th International Workshop on Artificial Neural Networks, IWANN 2005, *Lecture Notes in Computer Science, 3512* 1018-1026.

Espinoza, M., Suykens, J.A.K., & De Moor, B. (2006) Fixed-size least squares support vector machines: a large scale application in electrical load forecasting. *Computational Management Science, 3,* 113–129.

Ferrari-Trecate, G., & Muselli, M. (2002). *A new learning method for Piecewise Linear Regression*. ICANN '02 Proceedings of the International Conference on Artificial Neural Networks. LNCS 2415, 444-449.

Ferrari-Trecate, G., Muselli, M., Liberati, D., & Morari, M. (2001). *A Clustering Technique for the Identification of Piecewise Affine Systems.* In: Proceedings of HSSC 2001, 2304 of Lecture Notes in Computer Science Berlin: Springer-Verlag, 218-231.

Ferrari, S., & Stengel, R. F. (2005). Smooth Function Approximation by Neural Networks. *IEEE Transactions on Neural Networks,* 24-38.

Funahashi, K. (1989). On the approximate realization of continuous mapping by neural networks. *Neural Networks, 2*, 183-192.

Gallant , A. R., & Fuller, W. A. (1973). Fitting Segmented Polynomial Regression Models Whose Join Points have to be Estimated. *Journal of the American Statistical Association, 68*, No. 341. 144-147.

Goel A. & Pal, M. (2009). Application of support vector machines in scour prediction on grade-control structures. *Engineering Applications of Artificial Intelligence 22,* (2), 216–223.

Goethals, I., Pelckmans, K., Suykens, J.A.K., & De Moor, B. (2005). Identification of MIMO Hammerstein Models Using Least Squares Support Vector Machines *Automatica, 41*, 1263–1272.

Golub G., & Van Loan C., (1996). *Matrix Computations*. Baltimore: Johns Hopkins Univ Press,

Grossmann, A. & Morlet, J. (1984). Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM Journal of Analysis,15:* 723-736.

Gunn S.R. (1998). *Support Vector Machines for Classification and Regression.* University of Southampton Faculty of Engineering and Applied Science Department of Electronics and Computer Science, Technical Report pp 17-24, 41-46

Güzeliş, C., & Göknar, I. C. (1991). *A canonical representation for piecewise-affine maps and its applications to circuit analysis. IEEE Trans. Circuits Syst. I*, 38, 1342-1354.

Hao, J. (2008). *Adaptive feature selection in pattern recognition and ultra-wideband radar signal analysis.* Dissertation (Ph.D.), California Institute of Technology. http://resolver.caltech.edu/CaltechETD:etd-05302008-134607.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd Ed.). New Jersey, U.S.A: Prentice Hall.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. *Neural Networks, 2*, 359-366.

Huang, C.J., Lai, W.K., Luo, R.L., & Yan, Y.L. (2005). Application of support vector machines to bandwidth reservation in sectored cellular communications, *Engineering Applications of Artificial Intelligence, 18,*5, 585-594.

Hudson, D. J. (1966). Fitting Segmented Curves Whose Join Points Have to be Estimated. *Journal of the American Statistical Association, 61*, No. 316, 1097-1129.

Imai H., & Iri, M. (1986). An optimal algorithm for approximating a piecewise linear function, *Journal Information. Processing, 9,* No. 3, 159-162.

Jiang, J., Song, C., Zhao, H., Wu, C., & Liang, Y. (2009). Adaptive and Iterative Least Squares Support Vector Regression Based on Quadratic Renyi Entropy.

Jones, L.K. (1990). *Constructive approximations for neural networks by sigmoidal functions.* Proceedings of the IEEE, 78, 10, 1586–1589

Julian, P., Desages, A., & Agamennoni, O. (1999). High level canonical piecewise linear representation using a simplicial partition. *IEEE Trans. Circuits Syst. I*, 46, 463-480.

Kahlert, C., & Chua, L. O. (1990). A generalized canonical piecewise-linear representation. *IEEE Trans. Circuits Syst. I*, CAS-37, 373–383.

Kahlert C., & Chua, L. O. (1992). The complete canonical piecewise-linear representation-Part I: The geometry of the domain space. *IEEE Trans. Circuits Syst. I*, 39, 222-236.

Kang, S. M., & Chua, L. O. (1978). A global representation of multidimensional piecewise-linear functions with linear partitions. *IEEE Trans. Circuits Syst., 25*, 938–940.

Karush, W. (1939). *Minima of functions of several variables with inequalitiesas side constraints*. Master’s thesis, Dept. of Mathematics, Univ. of Chicago.

Kevenaar, T. Leenaerts, D., & van Bokhoven, W. (1994). Extensions to Chua's explicit piecewise-linear function descriptions. *IEEE Trans. Circuits, Syst. I*, 41, 308-314.

Konstantinides, K., & Natarajan, B. K. (1994). An Architecture for Lossy Compression of Waveforms Using Piecewise-Linear Approximation. *IEEE Transactions on Signal Processing, 42*. No. 9. 2449-2454.

Kuhn H.W. & Tucker A.W. (1951). *Nonlinear programming*. In: Proceeding.2nd Berkeley Symposium on Mathematical Statistics and Probabilistic, Berkeley. University of California Press, 481–492.

Lin, J. N., Xu, H., & Unbehahuen, R. (1994). A generalization of canonical piecewise-linear functions. *IEEE Trans. Circuits Syst. I*, 41,345-347.

Lin, J. N., & Unbehahuen, R. (1995). Canonical piecewise-linear networks. *IEEE Trans. Neural Netw., 6,* No. 1, 43-50.

Leenarts, D. (1999). On linear dynamic complementarity systems. *IEEE Trans. Circuits Systems:* Fundamantel. *Theory Application., 46*, No. 8, 1022-1026.

Lute, V., Upadhyay, A., & Singh, K. K. (2009). Support vector machine based aerodynamic analysis of cable stayed bridges, Advances in Engineering Software, 40, 9, 830-835.

Mallat, S., & Zhang, Z. (1993). Matching Pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing, 41*:3397-3415.

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A 209* (441–458): 415–446.

Mohammadi, M., & Gharehpetian, G. B. (2009). Application of Core Vector Machines for On-line Voltage Security Assessment using a DT-based Feature Selection Algorithm. *The Journal of IET, Generation, Transmission & Distribution, 3,* Issue 8, 701-712.

Montgomery, D.C., & Peck, E.A. (1992). *Introduction to linear regression analysis.* (2nd Ed.). New York, U.S.A.: John Wiley & Sons.

Moody, J. E., & Darken, C. J. (1989). Fast learning in networks of locally tuned processing units, *Neural Comput., 1,* 281–294.

Morlet, J.; Arens, G.; Fourgeau, E. & Giard, D. (1982). Wave propagation and sampling theory, Part1: Complex signal land scattering in multilayer media. *Journal of Geophysics,* 47: 203-221.

Mukherjee, S., Osuna, E. & Girosi, F. (1997). *Nonlinear Prediction of Chaotic Time Series using a Support Vector Machine*. Neural Networks for Signal Processing VII -Proceedings of the IEEE Workshop, New York.

Musavi, M. T., Ahmed, W., Chan K. H., Faris, K. B., & Hummels, D. M. (1992). On the training of radial basis function classifiers, *Neural Networks, 5,* 595-603.

Nilsson, N.J. (1965).*Learning machines: Foundations of Trainable Pattern Classifying Systems*. McGraw-Hill.

Olshausen, B.A. & Field, D.J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381:607-609.

Osowski, S. & Garanty, K. (2007). Forecasting of the daily meteorological pollution using wavelets and support vector machine. *Engineering Applications of Artificial Intelligence, 20*, 6, 745-755.

Parente, J.C. (1999). *The Characteristics of Spline Functions.* From http://www.psych.mcgill.ca/misc/fda/ex-basis-b1.html

Park, J., & Sandberg, I. (1991). Universal approximation using radial-basis-function networks. *Neural Computation, 3* (2):246-257.

Pelckmans, K., Suykens, J. A. K., & De Moor, B. (2005c). Building sparse representations and structure determination on LS-SVM substrates. *Neurocomputing, 64*, 137–159.

Pittman, J., & Murthy, C.A. (2000). Fitting Optimal Piecewise Linear Functions Using Genetic Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*, No. 7, 701-718.

Roger, D. F., & Adams, J. A. (1990). Mathematical elements for computer graphics (2nd Ed.). New York, U.S.A: McGraw-Hill.

Samanta, B., Al-Balushi, K.R., & Al-Araimi, S.A., (2001). *Use of genetic algorithm and artificial neural network for gear condition diagnostics*. Proceedings of COMADEM 2001, University of Manchester, UK, pp. 449–456.

Saunders, C., Gammerman, A., & Vovk,V. (1998). *Ridge regression learning algorithm in dual variables*. Proceedings of the 15th International Conference on Machine Learning (ICML-98), Madison, WI 515-521.

Schoenberg, I. J. (1967). *On spline functions. Inequalities.* New York, U.S.A: Academic Press, 255-291.

Schumaker, L.L. (1981). *Spline functions: Basic theory*. New York, U.S.A.: John Wiley & Sons.

Seber, G.A.F., & Wild, C.J. (2003). *Nonlinear Regression*. A. John Wiley&Sons, USA, 768s.

Smola, A., Schölkopf, B. & Müller, K.-R. (1998).General cost functions for support vector regression. In Ninth Australian Congress on Neural Networks.

Smola, A. J. & Schölkopf, B. (1998). *A Tutorial on Support Vector Regression*. Neurocolt Technical Series Report, 30, 3-8.

Sun, Z. H., & Sun, Y. X. (2003). *Fuzzy support vector machine for regression estimation.* In: IEEE International Conference on Systems, Man and Cybernetics, 4, (18-21), 3336-3341.

Suykens, J.A.K., Lukas, L., & Wandewalle, J. (2000). *Sparse approximation using least squares support vector machines*. In Proceeding of the IEEE International Symposium on Circuits and Systems (ISCAS 2000), 757-760.

Suykens, J.A.K. (2001). *Nonlinear modelling and support vector machines*. In: Proceedings of IEEE Instrumentation and measurement technology conference, Budapest, Hungary, 287– 294.

Suykens, J.A.K., Vandewalle, J., & De Moor, B. (2001). Optimal Control by Least Squares Support Vector Machines, *Neural Networks 14* (1) 23–35.

Suykens, J. A. K., De Brabanter, J., Lukas, L., & De Moor, B. (2002). Weighted least squares support vector machines: Robustness and sparse approximation. *Neurocomputing, 48* (1–4), 85–105.

Uykan, Z., Güzeliş, C., Çelebi M. E., & Koivo, H. N. (2000). Analysis of input-output clustering for determining centers of RBFN, *IEEE Trans. Neural Networks, 11* No:4, 851-858.

Tao, D. Tang, X., Li, X., & Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28,* 7, 1088-1099.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58, 1, 267-288.

Vapnik, V., & Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control 24,* 774-780.

Vapnik, V., & Chervonenkis, A. (1974). *Theory of Pattern Recognition (in Russian).* Moscow. Nauka.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. New York: Springer-Verlag.

Vapnik, V., Golowich, S. & Smola, A. (1996). Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems,* 9:281–287, Vapnik, V. & S. Golowich, & Smola A. (1997). *Support vector method for function approximation, regression estimation, and signal processing*. Cambridge, MA, 281–287, MIT Press.

Vong, C-M., Wong , P-K., & Li, Y-P. (2006). Prediction of automotive engine power and torque using least squares support vector machines and Bayesian inference. *Engineering Applications of Artificial Intelligence, 19,* 3, 277-287.

Wang, Y., Wang, S., & Lai, K. K. (2005). A new fuzzy support vector machine to evaluate credit risk. *IEEE Transactions on Fuzzy Systems, 13,* 6, 820-831.

Wu, C. G. (2006) *Study on Generalized Chromosome Genetic Algorithm and Iterative Least Squares Support Vector Machine Regression*, Doctoral Dissertation, Jilin University.

Weierstrass, K. (1885). *Über die analytische Darstellbarkeit sogenannter willkrlicher Functionen einer reellen Vernderlichen.* Sitzungsberichte der Kniglich Preuischen Akademie der Wissenschaften zu Berlin, 11.

Wu, Q., Yan, H., & Yang, H. (2008a). *A Hybrid Forecasting Model Based on Chaotic Mapping and Improved v-Support Vector Machine.* Proceedings of the 9th International Conference for Young Computer Scientists, 2701-2706, 18-21.

Wu, Q., Yan, H., & Yang, H. (2008b). *A Forecasting Model Based Support Vector Machine and Particle Swarm Optimization*. Proceedings of the 2008 Workshop on Power Electronics and Intelligent Transportation System, p.218-222, August 02-03.

Zhu, J., Rosset, S., Hastie, T. & Tibshirani, R. (2003), *1-norm support vector machines,* Neural Information Proceeding Systems 16.