A Study on the Fuzzy Modeling of Nonlinear Systems Using Kernel Machines

February 2007

Jongcheol Kim

A Study on the Fuzzy Modeling of Nonlinear Systems Using Kernel Machines

A Study on the Fuzzy Modeling of Nonlinear Systems Using Kernel Machines

by

Jongcheol Kim

School of Integrated Design Engineering

KEIO UNIVERSITY

A thesis submitted to the Faculty of Science and Technology, Keio University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the School of Integrated Design Engineering

Yokohama, Japan

2007. 2. 1.

Approved by

Yasuo Suga

Major Advisor

Abstract

This thesis presents new approaches to fuzzy inference system for modeling nonlinear systems based on input and output data using kernel machines. It is an important issue how to select the best structure and parameters of the fuzzy model from given input-output data. To solve this problem, this thesis proposes the state-of-the-art kernel machine as the fuzzy inference engine. The kernel machine contains two modules such as the machine learning and the kernel function. The machine learning is a learning algorithm. The kernel function projects input data into high dimensional feature space. In this thesis, an extended Support Vector Machine (SVM), an extended Feature Vector Selection (FVS) and an extended Relevance Vector Machine (RVM) as kernel machines are used.

In the proposed fuzzy system, the number of fuzzy rules and the parameter values of membership functions are automatically generated using extended kernel machines such as an extended SVM, an extended FVS and an extended RVM. The structure and learning algorithm of the FIS using an extended SVM, an extended FVS and an extended RVM are presented, respectively. The learning algorithm of the extended FVS is faster than the extended SVM. The extended FVS consists of the linear transformation part of input variables and the kernel mapping part. The linear transformation of input variables is used to solve problem selecting the best shape of the Gaussian kernel function. The extended RVM generates the smaller number of fuzzy rules than the extended SVM. The extended RVM does not need the linear transformation of input variables because the basis function of the extended RVM is not restricted within the limitation of the kernel function.

As the basic structure of the proposed fuzzy inference system, the Takagi-Sugeno (TS) fuzzy model is used. After the structure is selected, the parameter values in the consequent part of TS fuzzy model are determined by the least square estimation method. In particular, the number of fuzzy rules can be reduced by adjusting the

linear transformation matrix or the parameter values of kernel functions using a gradient descent method.

Some examples involving benchmark nonlinear systems are included to illustrate the effectiveness of the proposed techniques.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

Conventional mathematical modeling approaches have difficulty in modeling many systems because of the lack of exact knowledge, highly nonlinear behaviors or performance limitation.

To overcome this problem, the neuro-fuzzy system has been popularly developed for modeling of nonlinear systems based on input and output data [1] [2] [3] [4] [5] [6]. The advantage of integrating neural networks and fuzzy inference system (FIS) is that neuro-fuzzy systems are able not only to describe target systems using fuzzy logic and reasoning of fuzzy system but also to decide its parameters using the learning and adaptive capability of neural network. Generally, neuro-fuzzy modeling from numeric data consists of two parts that are structure identification and parameter identification. The process of structure identification determines the number of fuzzy rules or variables selection. The process of parameter identification decides the parameters of membership functions in antecedent parts and coefficients of linear equations in consequent parts.

However, if training data set for learning has measurement noise and (or) available data size is too small in the real system modeling, neural network can bring out over-fitting problem which is a factor of poor generalization. It is an important problem to select the optimal structure of the neuro-fuzzy model for good generalization, such as the number of fuzzy rules, parameters of membership functions and coefficients in consequent part.

In this thesis, we propose new approaches to FIS for modeling nonlinear system based on input and output data using kernel machines such as an extended Support Vector Machine (SVM) [7], an extended Feature Vector Selection (FVS) [8] and an extended Relevance Vector Machine (RVM) [9].

The proposed FIS performs system optimization and generalization simultaneously. As the basic structure of the proposed fuzzy inference system, the Takagi-Sugeno (TS) fuzzy model [10] is used. In the proposed fuzzy system, the number of fuzzy rules and the parameter values of membership functions are automatically generated. In addition, the number of fuzzy rules can be reduced by adjusting the linear transformation matrix or the parameter values of kernel functions using a gradient descent method. After the structure fuzzy system is determined, the parameter values in the consequent parts of TS fuzzy model are determined by the least square estimation method.

1.2 Previous Research

The main issue in neuro-fuzzy modeling is how to decide the best structure and parameters from a given input-output data of the particular systems, such as the number of fuzzy rules, parameters of membership functions in antecedent parts and coefficients in consequent parts. If a fuzzy model has too many rules, it decreases the error between a given data output and fuzzy model output, but can cause overfitting and decrease computational power. By contrast, if a fuzzy model has too small rules, it increases computational power and prevents overfitting but can increase error.

The conventional structure identification of neuro-fuzzy modeling is closely related to the partitioning of input space for fuzzy rule generation.

Table 1.1 Various input space partition methods for fuzzy rule extraction

Group	Method	Disadvantage
Partition	Grid Partition	Course of dimensionality
	Tree Partition	Number of rule exponential increasing
	Scatter Partition	Completeness not guaranteed
	GA Algorithm based Partition	Long learning time
Clustering	Fuzzy C-mean Clustering	Predetermined the number of clustering
	Mountain Clustering	Let perception grid points as the candidate for clustering center
	Hybrid Clustering	Depending on implementation

1.2.1 Partitioning of input space

There are two kinds of groups for fuzzy rule generation from the data such as partition and clustering as shown in Table 1.1. One group is the partition of input space. The partition of input space can be categorized into the following methods.

- **Grid Partition** [11] [12] : As shown in Fig. 1.1(a), input space is divided into grid partition using grid type.
- **Tree Partition** [13] : As shown in Fig. 1.1(b), each region is uniquely specified along a corresponding decision tree.
- **Scatter Partition** [14] : As shown in Fig. 1.1(c), scatter partition is illustrated as the subset of the whole input space.
- **GA Algorithm based Partition** [15] : As shown in Fig. 1.1(d), GA Algorithm based partition is presented as the partition method using GA algorithm which divides the input space into disjoint decision areas.

The other group is the clustering of input space. The clustering method is classified into the following methods.

- **Fuzzy C-mean Clustering** [16] : Fuzzy C-mean clustering partitions the collection of n vector $\mathbf{x}_j, (j = 1, ..., n)$ into C groups $G_i, (i = 1, ..., c)$ and finds a

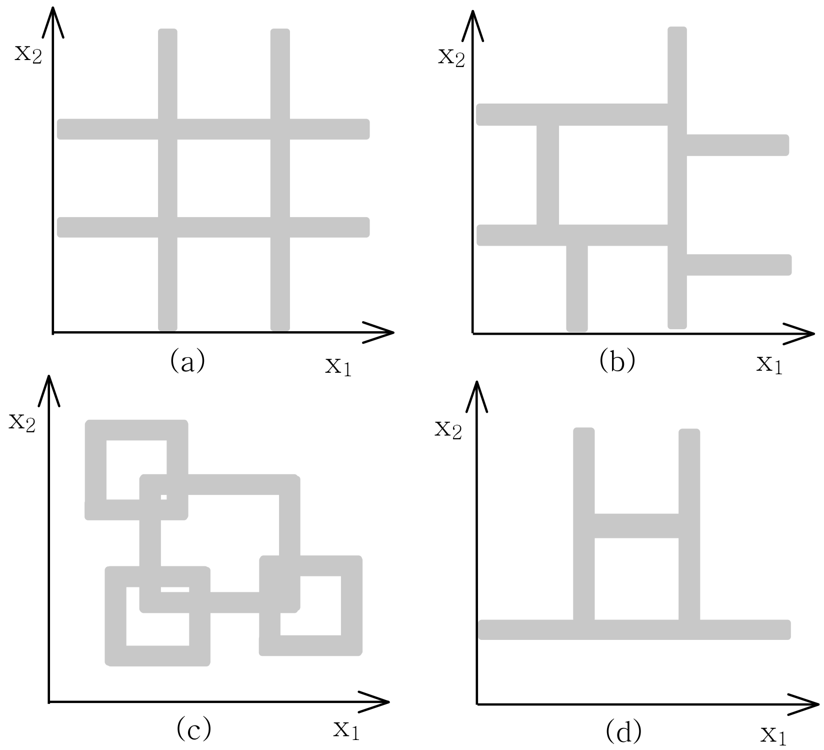


Fig. 1.1 Various input space partitioning methods, (a) Grid partition, (b) Tree partition, (c) Scatter partition and (d) GA algorithm based partition

cluster center in each group such that a cost function of dissimilarity measure is minimized.

- **Mountain Clustering** [17] : Mountain clustering is a relatively simple and effective approach to approximate estimation of cluster centers on the basis of a density measure.

Table 1.1 summaries various input space partitioning methods for fuzzy rule extraction and these disadvantages. The conventional partition of input space in structure identification is separated from parameter identification determining the value of parameter. Besides this process is isolated system optimization involving parameter and structure optimization. In particular, partition methods have disadvantages, which include the course of dimensionality [11], an exponential increase in the number of rules [13], unpredictable completeness [14] or computation cost

[15]. In clustering techniques, the number of cluster must be known in advance [16] [18] , or previously settled grid points of grid lines can function as candidates for cluster centers [17] [19]. Traditional sequential learning approaches of structure identification and parameter identification are adequate for off-line learning instead of on-line learning [20].

1.2.2 Statistical techniques based neuro-fuzzy modeling

Recently, the state-of-the-art kernel machine has actively applied to various fields [21] [22] [23] [24] [25] [26] [27]. The kernel machine is derived from the statistical learning theory. The kernel machine contains two modules such as the machine learning and the kernel function. The machine learning is a learning algorithm. The kernel function projects input data into high dimensional feature space to increase the computational power.

In kernel machine, the most popular method is Support Vector Machine (SVM). The SVM [21] has delivered good performance in various application. In particular, the SVM has been used in order to find the number of network nodes or fuzzy rules based on given error bound [28] [29] [30]. The Support Vector Neural Network (SVNN) was proposed to select the appropriate structure of radial based function network for the given precision [28]. Support vector learning mechanism for fuzzy rule-based inference system was presented in [29].

However, these methods have same Gaussian kernel parameters, completeness is not guaranteed. It means that the number of fuzzy rules is not really simplified. In this thesis, the number of rules is reduced by adjusting the parameter values of membership functions using a gradient descent algorithm during the learning process. Once a structure is selected, the parameter values in consequent part of TS fuzzy model are determined by the least square estimation method.

1.2.3 Kernel machines

The kernel machine is the large class of learning algorithms with kernel function. The kernel machine generally deals with trade-off between fitting the training data and simplifying model capacity. Recently, kernel machines have been popularly used in many applications including face recognition [31] [32] [33] [34], bioinformatics [35] [36] [37], text categorization [38] [39], time series analysis [40] [41] [42] [43], machine vision [44] [45], signal processing [46] and nonlinear system identification [47] [48].

As kernel machines, Support Vector Machine (SVM), Feature Vector Selection (FVS) and Relevance Vector Machine (RVM) are noticeable methodologies. These kernel machines are summarized as follows:

Support Vector Machine (SVM) [21]

The SVM has strong mathematical foundations in statistical learning theory. It is a learning system designed to trade-off the accuracy obtained particular training set and the capacity of the system. The structure of the SVM is the sum of weighted kernel functions. The SVM determines support vectors and weights by solving a linearly constrained quadratic programming problem in a number of coefficients equal to the number of data points. The SVM is generally divided into Support Vector Classification (SVC) [49] and Support Vector Regression (SVR) [50].

Feature Vector Selection (FVS) [26]

The FVS is based on kernel method. It performs a simple computation optimizing a normalized Euclidean distance into the feature space. The FVS technique is to select feature vector being a basis of data subspace and capturing the structure of the entire data into feature space. Once the feature vector is selected, the output of FVS is calculated using a kernel function approximation algorithm. The FVS is also used for classification [37] and regression [26].

Relevance Vector Machine (RVM) [27]

The RVM has an exploited probabilistic Bayesian learning framework. It acquires relevance vectors and weights by maximizing a marginal likelihood. The structure of the RVM is described by the sum of product of weights and kernel functions. The kernel function means a set of basis function projecting the input data into a high dimensional feature space. The RVM is also presented for classification and regression.

Table 1.2 Compared results of SVM, FVS and RVM

	SVM	FVS	RVM
Sparsity	Bad	Middle	Good
Generalization	Good	Bad	Good
Computation time	Middle	Short	Long
Flexibility of kernel	No	No	Yes

Now, we compare the characteristics of the SVM, FVS and RVM. The compared results are listed in Table 1.2.

In sparsity, the number of the extracted support vectors grows linearly with the size of training set. On the contrary, the RVM achieves sparsity because the posterior distributions of many of weights are sharply peaked around zero. The FVS has middle sparsity because it extracts feature vector as a basis of data subspace. Both SVM and RVM deal with the generalization, but the FVS do not achieve generalization. The RVM has long computation time because it has order $O(M^3)$ complexity with the M number of basis function. Because the SVM solves the quadratic programming problem, the computation time of SVM is longer than the FVS. Both SVM and FVS must the Mercer's condition of kernel function. It means that the kernel function is symmetric positive finite definite. But contrast, because the RVM has only basis function as kernel function, it's kernel function does not need to satisfy the Mercer's condition.

In following chapters, we will present SVM, FVS and RVM in detail, respectively.

1.3 Original Contributions

In this thesis, we describe new approaches to fuzzy inference system (FIS) for modeling nonlinear systems based on input and output data using kernel machines. As the basic structure of the proposed fuzzy inference system, the Takagi-Sugeno (TS) fuzzy model is used.

We have the following original contributions in the areas of fuzzy modeling using the state-of-the-art kernel machines, such as the extended Support Vector Machine (SVM) [7], the extended Feature Vector Selection (FVS) [8] and the extended Relevance Vector Machine (RVM) [9].

- We propose the FIS using an extended SVM for modeling the nonlinear systems. In the proposed FIS, the number of fuzzy rules and the parameter values of fuzzy membership functions are automatically generated using an extended SVM. In particular, the number of fuzzy rules can be reduced by adjusting the parameter values of the kernel functions using the gradient descent method.
- We propose the FIS using an extended FVS for modeling the nonlinear systems. In the proposed FIS, the number of fuzzy rules and the parameter values of fuzzy membership functions are also automatically determined using an extended FVS. In addition, the number of fuzzy rules can be reduced by adjusting the linear transformation matrix of input variables and the parameter values of the kernel function using the gradient descent method.
- We propose the FIS using an extended RVM for modeling nonlinear systems with noise. In the proposed FIS, the number of fuzzy rules and the parameter values of fuzzy membership functions are automatically decided using an extended RVM. In particular, the number of fuzzy rules can be reduced under the process of optimizing a marginal likelihood by adjusting parameter values of kernel functions using the gradient ascent method.

The kernel machine already works fine system modeling from input and output. However, there are several advantages of the proposed FIS using the extended SVM, FVS and RVM, respectively.

- The SVM, FVS and RVM describe only input and output of system as black-box. It is difficult to make out interior state of system. On the contrary, because the FIS describes system using if-then rules with membership functions qualitatively, it can help us to grasp the system.
- Once the black-box system is presented as the FIS, it is easy to design the controller. As an example, the well known parallel distributed compensation (PDC) can be utilized [51].

- If we model the nonlinear system as TS fuzzy model, we can prove the stability of system [52].

1.4 Thesis Overview

This thesis presents the fuzzy inference systems of nonlinear systems using kernel machines such as the extended Support Vector Machine (SVM), the extended Feature Vector Selection (FVS) and the extended Relevance Vector Machine (RVM). Each of the original contributions described in the previous section is presented in the following separated chapters.

Chapter 1 describes the background, motivation, contribution and the outline of this work.

Chapter 2 describes the preliminaries of the fuzzy system, statistical learning theory and kernel-induced feature space. In particular, the fuzzy set and logic, fuzzy reasoning and Takagi-Sugeno fuzzy model in fuzzy system are introduced. In statistical learning theory, generalization error, empirical risk minimization and structure risk minimization principle are presented. In kernel-induced feature space, learning in feature space and kernel function are described.

Chapter 3 describes the fuzzy inference system using an extended SVM. The extended SVM is introduced as fuzzy inference engine. The structure and learning algorithm of the FIS using an extended SVM are proposed. The proposed FIS is tested in three numerical examples.

Chapter 4 describes the fuzzy inference system using an extended FVS. The extended FVS is also proposed as fuzzy inference engine. The learning algorithm of the extended FVS is faster than the extended SVM. The extended FVS consists of the linear transformation part of input variables and the kernel mapping part. The linear transformation of input variables is used to solve problem selecting the best shape of the Gaussian kernel function. The proposed FIS is evaluated in the examples of two nonlinear functions.

Chapter 5 describes the fuzzy inference system using an extended RVM. The extended RVM is also proposed as fuzzy inference engine. The extended RVM generates the smaller number of fuzzy rules than the extended SVM. The extended

RVM does not need the linear transformation of input variables because the basis function of the extended RVM is not restricted within the limitation of the kernel function. The structure and learning algorithm of the FIS using an extended RVM are presented. The proposed FIS is evaluated in the examples of nonlinear dynamic systems and robot arm data.

Chapter 6 summarizes the results of this thesis and discusses future research initiatives.

CHAPTER 2

Preliminaries

This chapter introduces fuzzy system, statistical learning theory and kernel-based feature space. In fuzzy system, fuzzy set and fuzzy logic are presented. The Takagi-Sugeno (TS) fuzzy model known as one of the most outstanding fuzzy systems is also introduced. In statistical learning theory, generalization error, empirical and structure risk minimization principle are presented. In kernel-based feature space, learning in feature space and the properties of kernel function are illustrated.

2.1 Fuzzy Systems

A fuzzy system is a rule-based system that uses fuzzy set and fuzzy logic to reason about data. Fuzzy logic is a computational paradigm that provides a mathematical tool for representing information in a way that resembles human linguistic communication and reasoning processes [53] [54] [55] [56] [57] [58] [59] [60].

2.1.1 Fuzzy set and fuzzy logic

Lotfi Zadeh established the foundation of fuzzy logic in a seminal paper entitled "Fuzzy Sets" [61]. In [61], fuzzy sets were imprecisely defined as sets and classes "play an important role in human thinking, particularly in the domains pattern

Table 2.1 The equivalence between isomorphic domains

Set	Logic	Algebra
Membership	Truth	Value
Member ($\in$)	True (T)	1
Non-member	False (F)	0
Intersection ($\cap$)	AND ($\wedge$)	Product($\times$)
Union ($\cup$)	OR ($\vee$)	Sum ($+$)
Complement ($-$)	NOT ($\sim$)	Complement ($'$)

recognition, communication of information, and abstraction." Fuzzy sets are the generalization of crisp sets with crisp boundaries.

Let us now basic definitions concerning fuzzy sets.

Definition 2.1.1 *[55] [62] If X is a collection of objects denoted generically by x, then a **fuzzy set** A in a universe of discourse X is defined as a set of ordered pairs:*

$$A = \{(x, \mu_A(x))|x \in X\} \tag{2.1}$$

*where $\mu_A(x)$ is called the **membership function (MF)** for the fuzzy set A. The MF is a mapping*

$$\mu_A(x) : X \longrightarrow [0, 1]. \tag{2.2}$$

Note that each element of X is mapped to a membership grade between 0 and 1.

*The operation that assigns a membership function $\mu_A(x)$ to a given value x is called **fuzzification**.*

The most commonly used membership functions are triangular, trapezoidal, Gaussian, generalized bell and sigmoidal MFs.

The rules of FIS are expressed as the logical form of **if ... then** statements. J. M. Mendel pointed out fuzzy logic system as "It is well established that propositional logic is isomorphic to set theory under the appreciate correspondence between components of these two mathematical system. Furthermore, both of these systems are isomorphic to a Boolean algebra." [55] [63]. Some of the most important equivalence between these isomorphic domains are shown in Table 2.1.

In fuzzy domains, fuzzy operators are needed such as crisp operators. The following fuzzy operators most commonly used in the frame of fuzzy systems [55] [62].

Operators for intesection/AND operations ($\mu_{A\cap B}(x) = \mu_A(x) \wedge \mu_B(x)$): The intersection/AND of two fuzzy sets A and B is defined as the following T-norm operators,

$$\begin{aligned}
minimum &: min(\mu_A(x), \mu_B(x)) \\
algebraic\ product &: \mu_A(x) \cdot \mu_B(x) \\
bounded\ product &: max(0, \mu_A(x) + \mu_B(x) - 1) \\
drastic\ product &: \begin{cases} \mu_A(x) & , \texttt{ if } \mu_B(x) = 1 \\ \mu_B(x) & , \texttt{ if } \mu_A(x) = 1 \\ 0 & , \texttt{ if } \mu_A(x), \mu_B(x) < 1. \end{cases}
\end{aligned}$$

Operators for union/OR operations ($\mu_{A\cup B}(x) = \mu_A(x) \vee \mu_B(x)$): The union/OR of two fuzzy sets A and B is defined as the following T-conorm operators,

$$\begin{aligned}
maximum &: max(\mu_A(x), \mu_B(x)) \\
algebraic\ sum &: \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x) \\
bounded\ sum &: min(1, \mu_A(x) + \mu_B(x)) \\
drastic\ product &: \begin{cases} \mu_A(x) & , \texttt{ if } \mu_B(x) = 0 \\ \mu_B(x) & , \texttt{ if } \mu_A(x) = 0 \\ 1 & , \texttt{ if } \mu_A(x), \mu_B(x) > 1. \end{cases}
\end{aligned}$$

Operators for complement/NOT ($\mu_{\overline{A}}(x) = \mu_{\sim A}(x)$): The complement/NOT of fuzzy sets A is defined as the following fuzzy complement,

$$fuzzy\ complement \ : \ 1 - \mu_A(x)$$

2.1.2 Fuzzy inference system

Zadeh pointed out that conventional techniques for system analysis are intrinsically suited for dealing with humanistic systems [64]. Zadeh introduced the concept of linguistic variable as an alternative approach to modeling human thinking.

In fuzzy inference system, fuzzy if-then rules have the form [62],

$$\text{if } x \text{ is } A \text{ then } y \text{ is } B, \tag{2.3}$$

where A and B are linguistic values defined by fuzzy sets on universes of discourse X and Y, respectively. The input condition "$x\ is\ A$" is called the *antecedent* or *premise*. The output assignment "$y\ is\ B$" is called the *consequent* or *conclusion*.

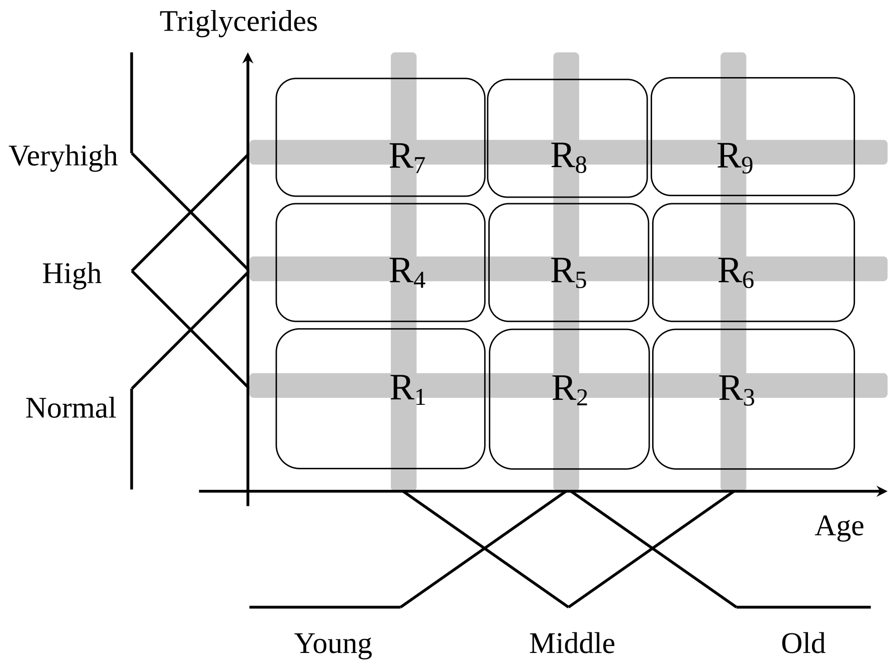


Fig. 2.1 Fuzzy rule vs input space partition

The generation of fuzzy if-then rules is related to the partitioning of input space partition. Figure 2.1 shows the example of the 2-dimensional input space partitioning [55]. In Fig. 2.1, the fuzzy linguistic variable Age has three membership functions: **Young**, **Middle** and **Old**. The fuzzy linguistic variable $Triglycerides$ has also three membership functions: **Normal**, **High** and **Veryhigh**. The total number of fuzzy rules is nine as shown in Fig. 2.1.

The 9-th fuzzy rule is described as follows:

$$\begin{aligned} R_9 \quad : \quad & \text{If } \ Age \text{ is } Old \ \text{ and } \ Triglycerides \text{ is } Veryhigh, \\ & \text{Then } \ Cardiac\ risk \text{ is } Dangerous. \end{aligned} \tag{2.4}$$

where **Dangerous** is linguistic fuzzy output variable.

In fuzzy inference system, fuzzy reasoning is necessary. Fuzzy reasoning is an inference procedure that derives a reasonable output and conclusion from a set of fuzzy if-then rules and known facts.

The inference procedure of fuzzy reasoning (approximate reasoning) is defined as follows:

Definition 2.1.2 *[62] Let A, A' and B be fuzzy sets of X, X and Y, respectively. Assume that the fuzzy implication A ⟶B is expressed as a fuzzy relation R on X × Y. Then the fuzzy set B' induced by "x is A'" and the fuzzy rule "if x is A $then$ y is B" is defined by*

$$\begin{aligned} \mu_{B'}(y) &= A' \circ R = A' \circ (A \longrightarrow B) \\ &= max_x min[\mu_{A'}(x), \mu_R(x,y)] \\ &= \vee_x[\mu_{A'}(x) \wedge \mu_R(x,y)] \end{aligned} \tag{2.5}$$

where a composition operator ∘ *means the* $max - min$ *composition.*

The fuzzy implication $A \longrightarrow B$ is defined as commonly operators, *minimum* and *product*. The most of composition operators have used the $max - min$ composition or the $max - product$ composition.

The output of FIS is crisp value. The process that extracts the best crisp output from a fuzzy output as a representative value is called *defuzzification*. Many defuzzification methods have been developed in literature. The most commonly have used method is the $Center\ of\ Gravity$ (COG), also called $Center\ of\ Areas$ (COA) or $Centroid$.

Given an output fuzzy set $A = \mu_A(x)$ defined in the universe X of the variable x, the defuzzified output y is given as follows:

- $Center\ of\ Gravity$ (COG):

$$y_{COG} = \frac{\int_X \mu_A(x)x\ dx}{\int_X \mu_A(x)\ dx} \tag{2.6}$$

  where $\mu_A(x)$ is the aggregated output MF.

Figure 2.2 shows graphically the operation fuzzy reasoning for two rules with two antecedents. Two fuzzy if-then rules with two antecedents are presented as follows:

$$\begin{aligned} R_1 &: \text{ if } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } z \text{ is } C_1, \\ R_2 &: \text{ if } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } z \text{ is } C_2, \end{aligned} \tag{2.7}$$

Two firing strength $\mu_{A_i}(x)$ and $\mu_{B_i}(y)$ $(i = 1, 2)$ indicate degrees to which the antecedent part of the fuzzy rule is satisfied. They are calculated using **AND** operator

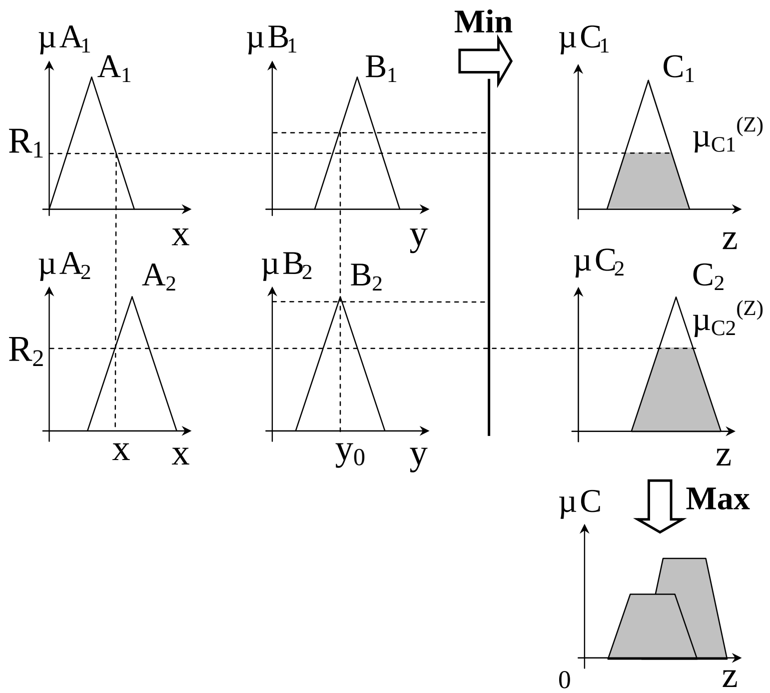


Fig. 2.2 Fuzzy reasoning for two rules with two antecedents

$(min(\mu_{A_i}(x), \mu_{B_i}(x)))$. Two induced consequent membership functions $\mu_{C_1}(z)$ and $\mu_{C_2}(z))$ are combined using **Union** operator $(max(\mu_{C_1}(z), \mu_{C_2}(z)))$. Once fuzzy reasoning is achieved, defuzzifer follows.

The basic structure of a fuzzy system consists of four conceptual components as shown in Fig. 2.3 (1) a **knowledge base**, which consists of a **database** that defines the membership functions used in the fuzzy rules, a **rule base** that contains a selection of fuzzy rules; (2) a **fuzzifier**, that translates crisp inputs into fuzzy values; (3) an **inference engine**, which applies the fuzzy reasoning mechanism; (4) **defuzzifier**, that extracts a crisp value from fuzzy output.

2.1.3 Takagi-Sugeno fuzzy model

Fuzzy Inference System (FIS)s have powerful capability for modeling complex nonlinear systems [1] [10] [16]. One of the most outstanding FISs, proposed by Takagi and Sugeno [10] [57], is known as the TS model. The TS fuzzy model consists of fuzzy if-then rules which map the input space into fuzzy regions and approxi-

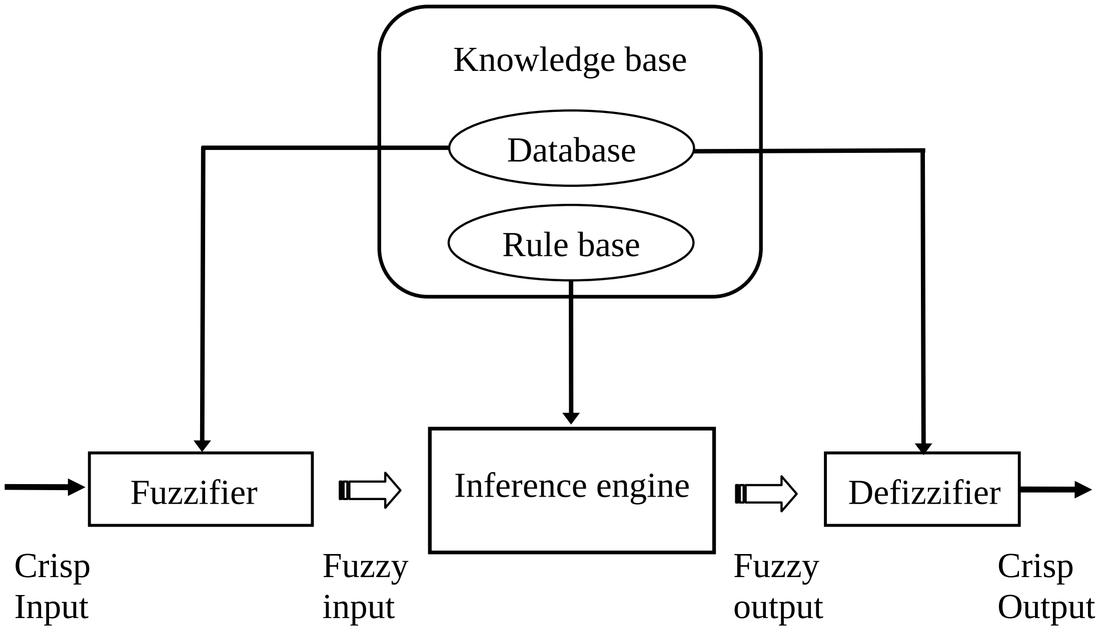


Fig. 2.3 The structure of a FIS

mate the system in every region by a local model corresponding to various operating points. The structure of TS fuzzy model is the combination of interconnected systems with linear models.

The TS fuzzy model suggested a systematic approach for generating fuzzy rules from a given input and output data set. This fuzzy model is presented as follows:

$$\begin{aligned} R_1 \quad &: \text{ If } x_1 \text{ is } M_{11} \text{ and ... and } x_D \text{ is } M_{1D}, \\ &\quad \text{Then } \; f_1 = a_{10} + a_{11}x_1 + \cdots + a_{1D}x_D \\ R_2 \quad &: \text{ If } x_1 \text{ is } M_{21} \text{ and ... and } x_D \text{ is } M_{2D}, \\ &\quad \text{Then } \; f_2 = a_{20} + a_{21}x_1 + \cdots + a_{2D}x_D \\ &\quad \vdots \qquad \vdots \qquad \vdots \\ R_n \quad &: \text{ If } x_1 \text{ is } M_{n1} \text{ and ... and } x_D \text{ is } M_{nD}, \\ &\quad \text{Then } \; f_n = a_{n0} + a_{n1}x_1 + \cdots + a_{nD}x_D. \end{aligned} \tag{2.8}$$

where n is the number of fuzzy rules, D is the dimension of input variables, $x_j(j = 1, 2, ..., D)$ is an input variable, f_i is the i-th local output variable, $M_{ij}(i = 1, 2, ..., n, j =$

$1, 2, ..., D)$ is a fuzzy set and $a_{ij}(i = 1, 2, ..., n, j = 0, 1, ..., D)$ is a consequent parameter.

The final output of TS fuzzy model is obtained as follows:

$$\begin{aligned} f(x) &= \frac{\sum\limits_{i=1}^{n} w_i f_i}{\sum\limits_{i=1}^{n} w_i}, \\ &= \frac{\sum\limits_{i=1}^{n} w_i (a_{i0} + a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{iD}x_D)}{\sum\limits_{i=1}^{n} w_i}, \\ &= \sum_{j=0}^{D} h_i a_{ij} x_j, \end{aligned}$$

where $x_0 = 1$,

$$h_i = \frac{w_i}{\sum\limits_{i=1}^{n} w_i}, \quad w_i = \prod_{j=1}^{D} M_{ij}(x_j), \tag{2.9}$$

w_i is the weight of the i-th If-then rule for input and $M_{ij}(x_j)$ is the membership grade of x_j in M_{ij}.

Sugeno-Kang proposed the procedure of TS fuzzy modeling as a nonlinear modeling framework. The methods of structure and parameter identifications were introduced. These methods had influence on the self-organizing fuzzy identification algorithm (SOFIA) [59] and neuro-fuzzy modeling techniques [1] [60].

2.2 Statistical Learning Theory

In this section, we introduce a statistical learning theory. Recently, statistical learning theory has been popularly developed in many application [21] [22] [23] [24] [25] [49] [50].

2.2.1 Generalization error

Generalization error is the sum of estimation error and approximation error as shown in Fig. 2.4 [65].

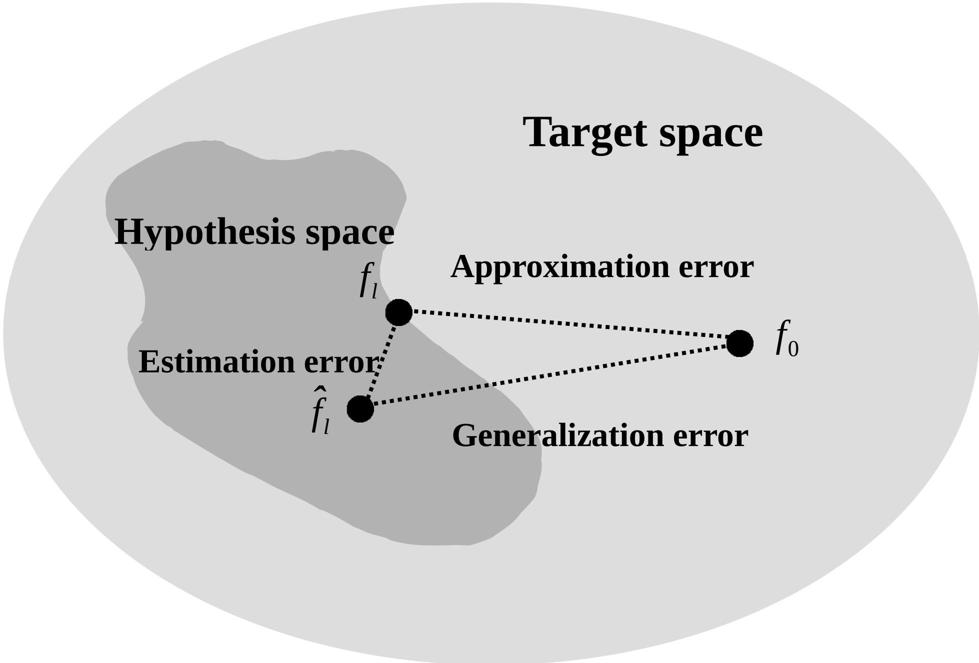


Fig. 2.4 Generalization error

- Approximation error is the one due to approximation from hypothesis space into target space.
- Estimation error is the one due to the learning procedure which results in a technique selecting the non-optimal model from the hypothesis space.

2.2.2 Empirical risk minimization principle

In statistical learning theory, the standard way to solve the learning problem is to define risk function, which measures the average amount of error associated with an estimator [66].

- Classical Regularization Networks

$$V(y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2 \tag{2.10}$$

- Support Vector Machines Regression

$$V(y_i, f(\mathbf{x}_i)) = |y_i - f(\mathbf{x}_i)|_\epsilon \tag{2.11}$$

Training data is $D_l \equiv \{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^{l}$, obtained by sampling l times the set $X \times Y$ according to $P(\mathbf{x}, y)$. If $V(y_i, f(\mathbf{x}_i))$ is the loss function measuring the error, when we predict y using $f(\mathbf{x})$, then the average error is called $expected\ risk$:

$$R[f] = \int_{X,Y} V(y_i, f(\mathbf{x}_i))P(\mathbf{x}, y))d(\mathbf{x})dy. \tag{2.12}$$

Let f_0 be the function which minimizes the expected risk in F:

$$f_0 = arg \min_F R[f]. \tag{2.13}$$

The function f_0 is ideal estimator, and it is often called $target\ function$. However, the probability distribution $P(\mathbf{x}, y)$ defining the expected risk is unknown. To overcome this problem, Vapnik [49] suggests $empirical\ risk\ minimization\ principle$,

$$R_{emp}[f] = \frac{1}{l}\sum_{i=1}^{l} V(y_i, f(\mathbf{x}_i)). \tag{2.14}$$

Formally, the theory answers the question of finding under which conditions the method of empirical risk minimization principle satisfies:

$$\lim_{l \to \infty} R_{emp}[\hat{f}_l] = \lim_{l \to \infty} R[\hat{f}_l] = R[f_0], \tag{2.15}$$

where $\hat{f}_l$ is the minimizer of the empirical risk (2.14) in F.

2.2.3 Structure risk minimization principle

The $Vapnic\ Chervoenkis(VC)\ dimension\ h$ is defined as follows:

Definition 2.2.1 *[49] The capacity of a set of function with logarithmic bounded growth function can be characterized by the coefficient h. The coefficient h is called the VC dimension of a set of indicator functions. It characterizes the capacity of a set of functions. When the growth function is linear, the VC dimension is defined to be infinite.*

The important outcome of the work of Vapnik and Chervonenkis is that the uniform deviation between empirical risk and expected risk in a hypothesis space can be bounded in terms of the VC-dimension, as shown in the following theorem [66]:

Theorem 2.2.1 *[66] [67] Let $A \leq V(y, f(\boldsymbol{x})) \leq B, f \in F$,with A and $B < \infty$, F be a set of bounded functions and h the VC-dimension of V in F. Then, with probability at least $1-\eta$, the following inequality holds simultaneously for all the elements f of F :*

$$R_{emp}[f]-(B-A)\sqrt{\frac{h\ln\frac{2l}{h}-\ln(\frac{\eta}{4})}{l}} \leq R[f] \leq R_{emp}[f]+(B-A)\sqrt{\frac{h\ln\frac{2l}{h}-\ln(\frac{\eta}{4})}{l}}. \quad (2.16)$$

The quantity $R[f] - R_{emp}[f]$ is often called the *estimation error*. Since the space F where the loss function V is defined is usually very large, one typically considers smaller hypothesis spaces H. The cost associated with restricting the space is called the *approximation error*. In the literature, space F where V is defined is called the *target space*, while H is so called the *hypothesis space* [66].

We define the set of nested subsets of hypothesis spaces $H_1 \subset H_2 \subset \cdots \subset H_{n(l)}$. If h_i is the VC dimension of space H_i, then $h_1 \leq h_2 \leq \cdots \leq h_{n(l)}$. (2.16) is rewritten as

$$R[f] \leq R_{emp}[f] + (B-A)\sqrt{\frac{h\ln\frac{2l}{h}-\ln(\frac{\eta}{4})}{l}}. \quad (2.17)$$

The idea of the structural risk minimization induction principle is the following. To provide the given set of functions with an admissible structure and then to find the function that minimize guaranteed risk (2.17) over given elements of the structure.

In Fig. 2.5, the relationship between approximation error, estimation error and generalization error about VC dimension is illustrated.

2.3 Kernel-Induced Feature Space

In this section, the learning in feature space and kernel function are introduced. The kernel technique performs a nonlinear mapping which projects input space into high dimensional feature space.

2.3.1 Learning in feature space

In general, the preprocessing step in learning machine contains representation of given input-output data [23]:

$$\mathbf{x} = (x_1, ..., x_n) \;\mapsto\; \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), ..., \phi_n(\mathbf{x})).$$

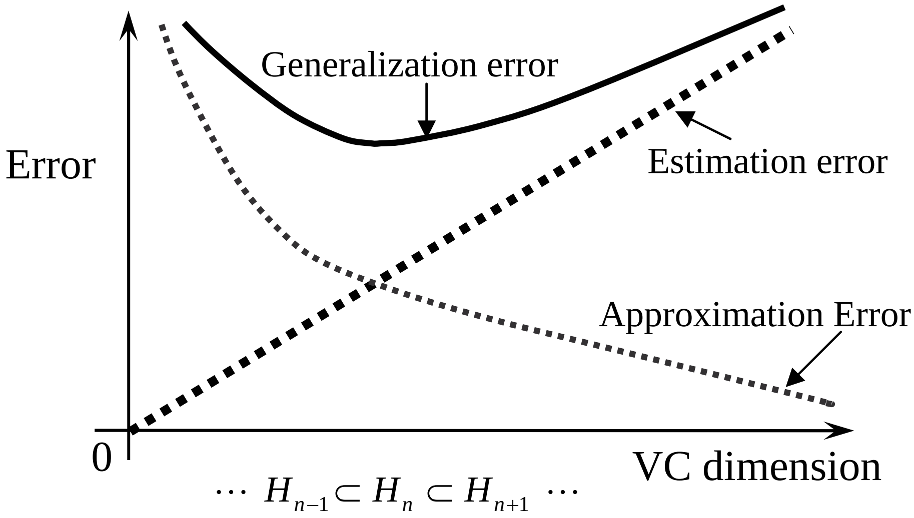


Fig. 2.5 The relationship between approximation error, estimation error and generalization error

This step is equivalent to mapping the input space X into a new space, $F = \{\phi(\mathbf{x})|\mathbf{x} \in X\}$.

To project the given data into hypothesis space can increase computational power in learning machine and can supply various methods for extracting relevant information through new representation of data. The quantities introduced to describe the data are called $features$, while original quantities are called $attributes$. The work of selecting the best suitable representation is known as the $feature\ selection$. The space X is referred to as the $input\ space$, while $F = \{\phi(\mathbf{x})|\mathbf{x} \in X\}$ is called the $feature\ space$ [23].

Figure 2.6 shows the example of a nonlinear mapping the training data in input space into a higher-dimensional feature space via ϕ. In input space, data can not be separated by linear function, but in feature space, data can be separated by linear function.

2.3.2 Kernel function

We present the definition and characteristic of kernel function. Firstly, finitely positive semi-definite function is defined.

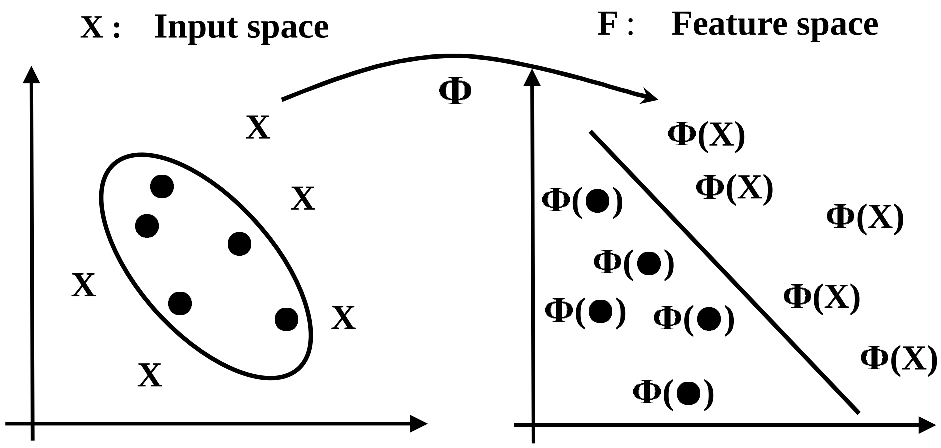


Fig. 2.6 Nonlinear mapping

Definition 2.3.1 *[25] A function*

$$K : X \times X \longrightarrow \mathbb{R} \tag{2.18}$$

satisfies the finitely positive semi-definite property if it is a symmetric function for which the matrices formed by restriction to any finite subset of the space X are positive semi-definite.

Definition 2.3.2 *[23] A kernel is a function K, such that for all* $\boldsymbol{x},\ \boldsymbol{z} \in X$

$$K(\boldsymbol{x}, \boldsymbol{z}) \quad = \quad < \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{z}) > \tag{2.19}$$

where ϕ is a mapping from X to an (inner product) feature space F.

Its arguments followed by the evaluation of the inner product in F if and only if it satisfies the finitely positive semi-finite property.

The following Mercer's theorem provides characterization when a function $K(\mathbf{x}, \mathbf{z})$ is a kernel.

Theorem 2.3.1 *[23] [68] Let X be a compact subset of $\mathbb{R}^n$. Suppose K is a continuous symmetric function such that the integral operator $T_k : L_2(X) \rightarrow L_2(X)$,*

$$T_k f(\cdot) := \int_X K(\cdot, \boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} \tag{2.20}$$

Table 2.2 Kernel function and type

Kernel Function	Type
$K(\mathbf{x},\mathbf{y}) = ((\mathbf{x}\cdot\mathbf{y})+1)^d$	Polynomial of degree d
$K(\mathbf{x},\mathbf{y}) = exp(-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2})$	Gaussian RBF
$K(\mathbf{x},\mathbf{y}) = exp(-\frac{\lvert\mathbf{x}-\mathbf{y}\rvert}{2\sigma^2})$	Exponential RBF
$K(\mathbf{x},\mathbf{y}) = tanh(a(\mathbf{x}\cdot\mathbf{y})-b)$	Multi-layer perceptron
$K(\mathbf{x},\mathbf{y}) = \frac{sin(N+\frac{1}{2})(\mathbf{x}-\mathbf{y})}{sin(\frac{1}{2}(\mathbf{x}-\mathbf{y}))}$	Fourier series

is positive. That is

$$\int_{X\times X} K(\boldsymbol{x},\boldsymbol{z})f(\boldsymbol{x})f(\boldsymbol{z})d\boldsymbol{x}d\boldsymbol{z} \geq 0, \tag{2.21}$$

for all $f \in L_2(X)$. *Then we can expand K(x,z) in a uniformly convergent series (on* $X \times X$*) in terms of* T_k*'s eigen-functions* $\phi_j \in L_2(X)$*, normalized in such a way that* $\|\phi_j\|_{L2} = 1$*, and positive associated eigenvalue* $\lambda_j \geq 0$*,*

$$K(\boldsymbol{x},\boldsymbol{z}) = \sum_{j=1}^{\infty} \lambda_j \phi_j(\boldsymbol{x})\phi_j(\boldsymbol{z}). \tag{2.22}$$

From these definition and theorem, we can summary kernel function as follows,

$$K(\mathbf{x},\mathbf{z}) = <\phi(\mathbf{x})\cdot\phi(\mathbf{z})> = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x})\phi_i(\mathbf{z}). \tag{2.23}$$

The following example in [21] gives brief understanding.

Example (Quadratic feature in [21] $\mathbb{R}^2$): Consider the map $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ with

$$\phi(\mathbf{x}) = \phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2), \tag{2.24}$$

where $x_1 \; and \; x_2 \in \mathbb{R}^2$, for instance, the polynomial kernel $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y})^d$.

For $d = 2$, and $\mathbf{x},\mathbf{y} \in \mathbb{R}^2$, we have

$$\begin{aligned}(\mathbf{x}\cdot\mathbf{y})^2 &= \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\cdot\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right)^2 \\ &= \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}\cdot\begin{pmatrix} y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{pmatrix} \\ &= (\phi(\mathbf{x})\cdot\phi(\mathbf{y})).\end{aligned} \tag{2.25}$$

In Table 2.2, the most commonly used kernel functions are presented.

We use the the following kernel matrix as kernel function in learning algorithm. Give a input data $\mathbf{X} = \{x_1, ..., x_l\}$ and kernel function K, the following kernel or Gram matrix $\mathbf{K}_{ij}$ is presented,

$$\mathbf{K}_{ij} = K(x_i, x_j), \text{ for } i, j = 1, ..., l. \tag{2.26}$$

The kernel matrix acts as an interface between the data input module and learning algorithm.

CHAPTER 3

Fuzzy Inference System Using an Extended SVM

This chapter describes the fuzzy inference system (FIS) using an extended Support Vector Machine (SVM) for modeling the nonlinear systems based on input and output data. The SVM is a learning system designed to trade-off the accuracy obtained particular training set and the capacity of the system [21] [65]. The structure of the SVM is the sum of weighted kernel functions. In the proposed FIS, the number of fuzzy rules and the parameter values of fuzzy membership functions are automatically generated using an extended SVM. In an extended SVM, the parameter values of the kernel function are adjusted using the gradient descent method. The number of fuzzy rules can be reduced by the extended SVM.

3.1 Introduction

Recently, the neural fuzzy approach has become one of the most popular research fields in system modeling describing the system's nature and behaviors [1] [3] [69]. The principle purpose of a neuro-fuzzy system is to apply learning technique of neural network to find and tune both the structure and the parameter of system based on FIS.

Main design issues of neuro-fuzzy system from numeric data are how to appropriately determine the number of fuzzy rule and how to precisely decide the membership function in antecedent parts and the value of parameter in consequent parts. Neuro-fuzzy modeling consists of structure identification and parameter identification. Structure identification methods determining the number of fuzzy rules have been variously introduced by [16] [17] [18] [19]. Parameter identification methods have generally used the gradient descent method.

Recently, the SVM has been popularly used for the system modeling and identification [70] [71] [72] [73] [74]. In particular, the SVM has been used in order to find the proper number of rules for the given precision [28] [75]. A Support Vector Neural Network (SVNN) using a radial basis function network was introduced [28]. In [75], a SVM was applied for simplifying FIS. However, because both papers used the same Gaussian kernel parameters, the number of fuzzy rules was not really minimized for the given precision.

To overcome this limitation, we propose a new FIS based on Takagi-Sugeno (TS) fuzzy model using an extended SVM. We uses an extended SVM without any bias. The number of new fuzzy rules can be reduced further by adjusting the parameter values of membership functions using a gradient descent method during the learning process. The proposed FIS can easily present a given system by nonlinear mapping which projects input space into high dimensional feature space. The structure of the proposed FIS is founded first by solving a constrained quadratic programming problem for a given modeling error. After the structure is selected, the parameter values in consequent part of TS fuzzy model are determined by the least square estimation method.

3.2 Support Vector Machines (SVM)

The SVM is derived from statistical learning theory [21]. Support Vector Machines (SVMs) are learning systems that use a hypothesis space of linear functions in a high dimensional kernel induced feature space. It determines support vectors and weights by minimizing an upper bound of generalization error [76] [77]. The output of the SVM is the sum of weighted kernel function. Kernel function projects the data into a high dimensional feature space to increase the computational power of the linear machine.

The SVM is generally divided into Support Vector Classification (SVC) [49] used to describe classification and Support Vector Regression (SVR) [50] used to describe regression. This section describes the SVR problem.

Consider the structure on the nonlinear function for approximating

$$f(x) = w \cdot \Phi(\mathbf{x}), \tag{3.1}$$

with $\Phi : \ R^l \ \rightarrow F, \ \ w \in F$, when Φ is nonlinear mapping, w is the associated weight and F is a feature space.

Suppose we have given data

$$(x_1, y_1), ..., (x_l, y_l).$$

Nonlinear function for approximating the set of data is presented as follows:

$$\begin{aligned} f(x) &= \sum_{i=1}^{l} (\alpha_i^* - \alpha_i)(\Phi(x_i) \cdot \Phi(\mathbf{x})), \\ &= \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) K(x_i, \mathbf{x}), \end{aligned} \tag{3.2}$$

where l is the number of data, $\mathbf{x} = [x_1, x_2, ..., x_l]$ is input data, α_i^* and α_i are Lagrange multipliers.

Let $\Phi(\mathbf{x})$ and w be the nonlinear mapping and the associated weight, respectively. The kernel function $K(x_i, \mathbf{x})$ is defined as a linear dot product of nonlinear mapping,

$$K(x_i, \mathbf{x}) = \Phi(x_i) \cdot \Phi(\mathbf{x}). \tag{3.3}$$

The parameters α and α^* of (3.2) are obtained by minimizing the following regularized risk functional $R_{eg}[f]$,

$$R_{eg}[f] = \frac{1}{2}\|w\|^2 + C \cdot R_{emp}[f], \tag{3.4}$$

where $\|w\|^2$ is a term which characterizes the model complexity, the second term is a empirical risk, $R_{emp}[f] = \sum_{i=1}^{l} L_\varepsilon(y)$, and C is a constant determining the trade-off and ε is the given precision.

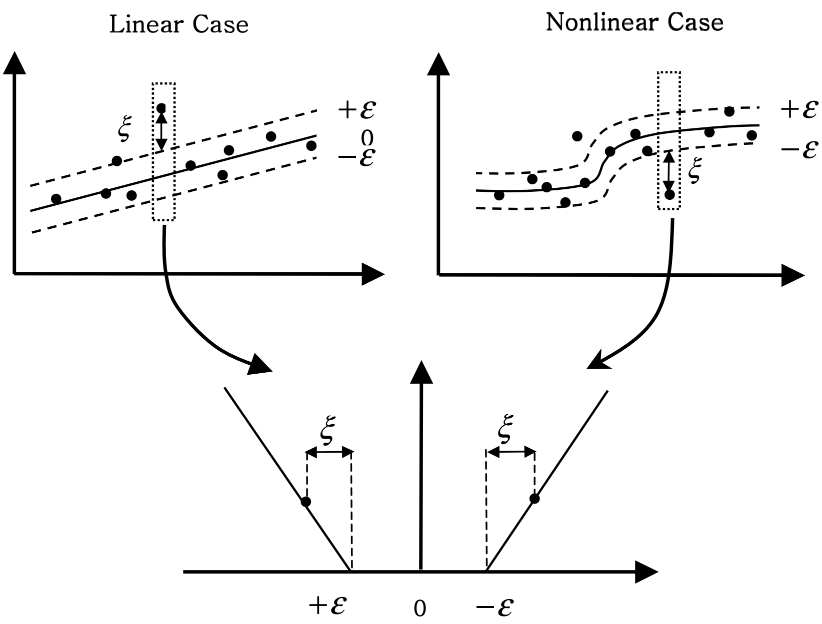


Fig. 3.1 The ε-insensitive band and loss for linear and nonlinear regression problem

Definition 3.2.1 *[23] The ε-insensitive loss function $L_\varepsilon(y)$ is defined by*

$$L_\varepsilon(y) = \begin{cases} 0 & \textit{for} \;\; |f(x) - y| < \varepsilon \\ |f(x) - y| - \varepsilon & \textit{otherwise}, \end{cases} \tag{3.5}$$

where f is a real-valued function on a domain X, $x \in X$ and $y \in \mathbb{R}$.

Figure 3.1 shows the form of ε-insensitive losses for zero and nonzero ε as a function of $y - f(x)$. The minimization of (3.4) is equal to the following constrained optimization problem,

$$minimize \quad \tau(w, \xi^*, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i^* + \xi_i), \tag{3.6}$$

$$subject\ to \begin{cases} y_i - w \cdot \Phi(x) & \leq \varepsilon + \xi_i^* \\ -y_i + w \cdot \Phi(x) & \leq \varepsilon + \xi_i \\ \xi_i^*, \xi_i & \geq 0, \;\; i = 1, ..., l, \end{cases} \tag{3.7}$$

where ξ_i^* and ξ_i are slack variables representing lower and upper constraints on the outputs of the systems.

To solve the optimization problem with constraints of inequality type can be converted to find the saddle point of the Lagrange functional

$$\begin{aligned} L(w,\xi^*,\xi,\alpha^*,\alpha,\beta^*,\beta) &= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i^*+\xi_i) - \sum_{i=1}^{l}(\beta_i^*\xi_i^*+\beta_i\xi_i) \\ &- \sum_{i=1}^{l}\alpha_i[y_i - w\cdot\Phi(x_i)+\varepsilon+\xi_i] \\ &- \sum_{i=1}^{l}\alpha_i^*[w\cdot\Phi(x_i)-y_i+\varepsilon+\xi_i^*], \end{aligned} \tag{3.8}$$

where Lagrange multipliers $\alpha_i^* \geq 0,\ \alpha_i \geq 0,\ \beta_i^* \geq 0,\ \beta_i \geq 0.$
The minimum with respect to $w,\ \xi^*,\ \xi$ of Lagrangian L implies the following conditions

$$\begin{aligned} \frac{\partial L}{\partial w} &= 0 \implies w = \sum_{i=1}^{l}(\alpha_i^*-\alpha_i)\Phi(x_i), \\ \frac{\partial L}{\partial \xi^*} &= 0 \implies 0 \leq \alpha^* \leq C, \\ \frac{\partial L}{\partial \xi} &= 0 \implies 0 \leq \alpha \leq C. \end{aligned} \tag{3.9}$$

The dual problem is given by

$$\begin{aligned} \min_{\alpha^*,\alpha} W(\alpha^*,\alpha) &= \min_{\alpha^*,\alpha} \frac{1}{2}\sum_{i=0}^{l}\sum_{j=0}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(x_i,x_j) \\ &- \sum_{i=0}^{l}(\alpha_i^*-\alpha_i)y_i + \sum_{i=0}^{l}(\alpha_i^*-\alpha_i)\varepsilon, \end{aligned} \tag{3.10}$$

with constraints

$$0 \leq \alpha, \alpha^* \leq C, \quad i = 1, ..., l.$$

In summary, approximate function from the set [65] is

$$f(x) = \sum_{i=0}^{l}(\alpha_i^*-\alpha_i)K(x_i,\mathbf{x}). \tag{3.11}$$

The optimization problem, $\min_{\alpha^*,\alpha}\ W(\alpha^*,\alpha)$, can be expressed in matrix notation as,

$$\min_{x} \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{c}^T\mathbf{x}, \tag{3.12}$$

where

$$\mathbf{H} = \begin{bmatrix} XX^T & -XX^T \\ -XX^T & XX^T \end{bmatrix}, \mathbf{c} = \begin{bmatrix} \varepsilon + Y \\ \varepsilon - Y \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \alpha \\ \alpha^* \end{bmatrix}, \tag{3.13}$$

with constraints

$$\alpha_i,\ \alpha_i^* \geq 0,\ \ i = 1, ..., l, \tag{3.14}$$

and

$$X = \begin{bmatrix} x_i \\ \vdots \\ x_l \end{bmatrix},\ \ Y = \begin{bmatrix} y_i \\ \vdots \\ y_l \end{bmatrix}. \tag{3.15}$$

In the process of solving this optimization problem, the vector from the training set that associate with nonzero Lagrange multipliers is called the *support vector*.

3.3 New Fuzzy Inference System Using an Extended SVM

This section describes the structure of the FIS using a SVM, the structure and learning algorithm of the FIS using an extended SVM, and input space partition method. In the proposed FIS, the number of fuzzy rules and the parameter values of fuzzy membership functions are automatically generated using an extended SVM. In particular, the number of fuzzy rules can be reduced by adjusting the parameter values of the kernel function using the gradient descent method.

3.3.1 The structure of the FIS using a SVM

Let us suppose that we have given input and output data

$$(x_1, y_1), ..., (x_n, y_n),$$

where $x_i (i = 1, 2, ..., n)$ is input data and $y_i (i = 1, 2, ..., n)$ is output data.

The proposed TS fuzzy model with fuzzy if-then rules can be represented as follows:

$$\begin{array}{lcl} \text{Rule 1} & : & \text{If } x_{11} \text{ is } M_{11} \text{ and ... and } x_{1D} \text{ is } M_{1D},\ \text{Then } f_1 = \theta_1 \\ \text{Rule 2} & : & \text{If } x_{21} \text{ is } M_{21} \text{ and ... and } x_{2D} \text{ is } M_{2D},\ \text{Then } f_2 = \theta_2 \\ \cdots & & \qquad\qquad \cdots \qquad \cdots \\ \text{Rule n} & : & \text{If } x_{n1} \text{ is } M_{n1} \text{ and ... and } x_{nD} \text{ is } M_{nD},\ \text{Then } f_n = \theta_n, \end{array} \tag{3.16}$$

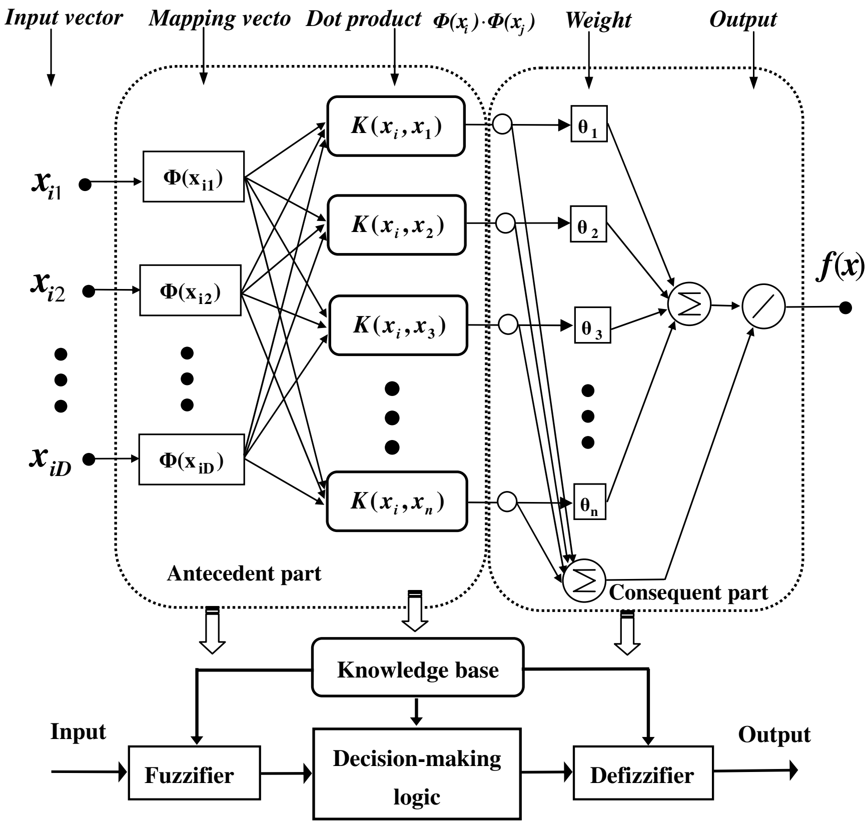


Fig. 3.2 The structure of the proposed FIS

where x_{ij} is input variable, f_i is local output variable, M_{ij} is fuzzy set and θ_i is consequent parameter. It is a simple Takagi-Sugeno (TS) type used singleton in consequent parts.

Now, we describe the structure of FIS using the SVM. It consists of four layers as shown in Fig. 3.2. The four layers involved in the proposed FIS are presented as follows:

Layer 1: Input space is nonlinearly mapped into feature space by a map Φ.

$$\mathbf{x} = (x_{i1}, ..., x_{iD}) \ \mapsto \ \Phi(\mathbf{x}) = (\Phi(x_{i1}), ..., \Phi(x_{iD})).$$

Layer 2: Dot products are computed with the mapped input $\mathbf{x}$ and the support vector (SV) being subset of input vector $\mathbf{x}$. It corresponds to evaluating kernel functions at locations $K(x_i, x)$. The modified Gaussian kernel function is used as follows:

$$K(x_i, \mathbf{x}) = \exp\Big(- \frac{(\mathbf{x} - x_i)^2}{2\sigma_i^2} \Big), \tag{3.17}$$

x_i is a SV and σ_i is called a kernel parameter. This kernel function is a Gaussian membership function in fuzzy inference system.
Layers 1 and 2 are the stage of fuzzifier.

Layer 3: In the nonlinear function considered for approximating the set of data,

$$f(x) \;=\; \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) K(x_i, \mathbf{x}). \tag{3.18}$$

Weights $(\alpha_i^* - \alpha_i)$ and support vectors (SVs) x_i are found by the constrained optimization for a given precision ε. The obtained SV becomes the center of the Gaussian membership function.
This layer means a decision-making logic determining the number of fuzzy rule by solving optimization problem from knowledge base being a minimum regularized risk functional $R_{eg}[f]$ in (3.4).

Layer 4: The defuzzification using center of gravity (COG) method is performed as follows:

$$COG \;:\; \frac{\sum_{i=1}^{n} w_i \theta_i}{\sum_{i=1}^{n} w_i}. \tag{3.19}$$

In (3.18), the $f(x)$ should be modified for the defuzzification of the COG.
Let

$$\lambda(x_i, \mathbf{x})) \;=\; \sum_{i=1}^{n} K(x_i, \mathbf{x}), \tag{3.20}$$

$$\Psi \;=\; \begin{bmatrix} K(x_1, x_1) & \dots & K(x_n, x_1) \\ \vdots & \vdots & \vdots \\ K(x_1, x_n) & \dots & K(x_n, x_n) \end{bmatrix}, \tag{3.21}$$

$$L \;=\; \begin{bmatrix} \lambda(x_1, x_1) & & 0 \\ & \ddots & \\ 0 & & \lambda(x_n, x_n) \end{bmatrix}. \tag{3.22}$$

Weight $\theta_1, ..., \theta_n$ can be expressed in terms of α and $\alpha*$, as

$$\theta \;=\; (\Psi^T \Psi)^{-1} \Psi^T L \Psi (\alpha^* - \alpha). \tag{3.23}$$

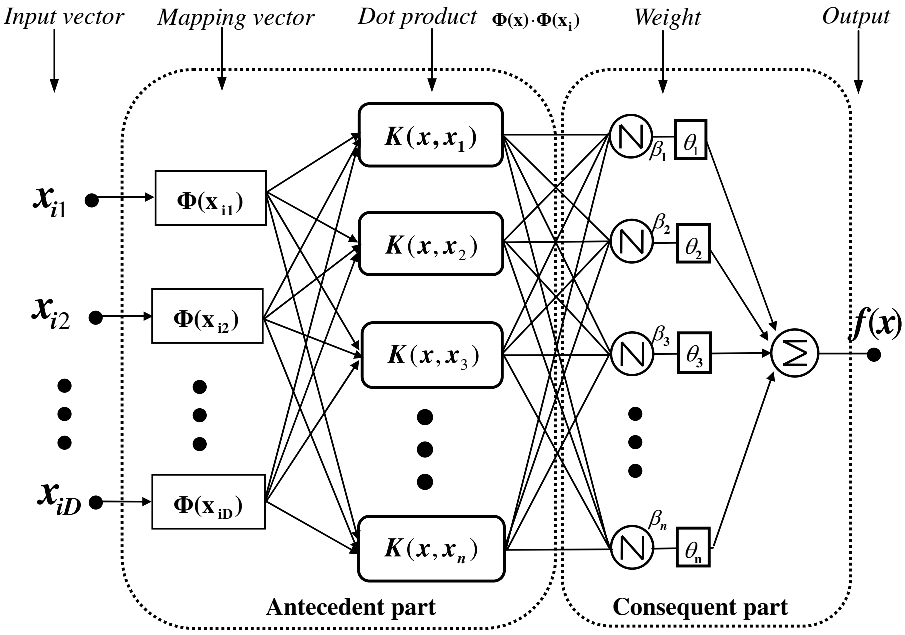


Fig. 3.3 The simple structure of the proposed FIS using a SVM

The modified function $f(x)$ is obtained,

$$\begin{aligned} f(\mathbf{x}) &= \frac{\sum_{i=1}^{n} K(x_i, \mathbf{x})\theta_i}{\sum_{i=1}^{n} K(x_i, \mathbf{x})} \\ &= \sum_{i=1}^{n} \beta_i \theta_i. \end{aligned} \tag{3.24}$$

More simple structure for learning algorithm is shown in Fig. 3.3.

Once a structure is selected, the parameter values in consequent part of TS fuzzy model are determined by the least square estimation (LSE) method or the recursive least square estimation (RLSE) algorithm.

3.3.2 The structure of the FIS using an extended SVM

The Takagi-Sugeno (TS) fuzzy model which is suitable for highly nonlinear systems has been one of the major topics in theoretical studies and practical applications of fuzzy modeling and control. The basic idea of the TS fuzzy models is to transform the input space into fuzzy regions and to approximate the system in every region by

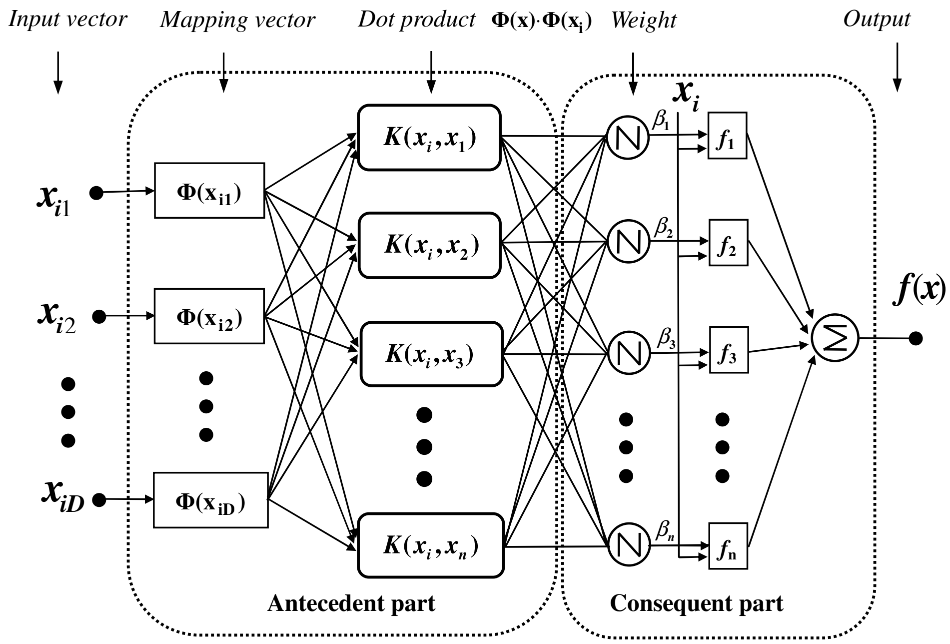


Fig. 3.4 The structure of the proposed FIS using an extended SVM

a local model. The overall fuzzy model consists of the combination of interconnected systems with linear models. Then the output of the whole fuzzy model is calculated as the weighted sum of the local models using the defuzzification scheme based on the Center Of Gravity (COG) method.

The TS fuzzy model using an extended SVM consists of the following If-Then rules:

$$\begin{aligned}
R_1 \quad &: \quad \text{If } x_{11} \text{ is } M_{11} \text{ and ... and } x_{1D} \text{ is } M_{1D}, \\
&\quad \text{Then } \; f_1 = a_{10} + a_{11}x_1 + \cdots + a_{1D}x_D \\
R_2 \quad &: \quad \text{If } x_{21} \text{ is } M_{21} \text{ and ... and } x_{2D} \text{ is } M_{2D}, \\
&\quad \text{Then } \; f_2 = a_{20} + a_{21}x_1 + \cdots + a_{2D}x_D \\
&\quad \vdots \qquad \vdots \qquad \vdots \\
R_n \quad &: \quad \text{If } x_{n1} \text{ is } M_{n1} \text{ and ... and } x_{nD} \text{ is } M_{nD}, \\
&\quad \text{Then } \; f_n = a_{n0} + a_{n1}x_1 + \cdots + a_{nD}x_D. \qquad (3.25)
\end{aligned}$$

The structure of the FIS using an extended SVM based on TS fuzzy model is illustrated in Fig. 3.4.

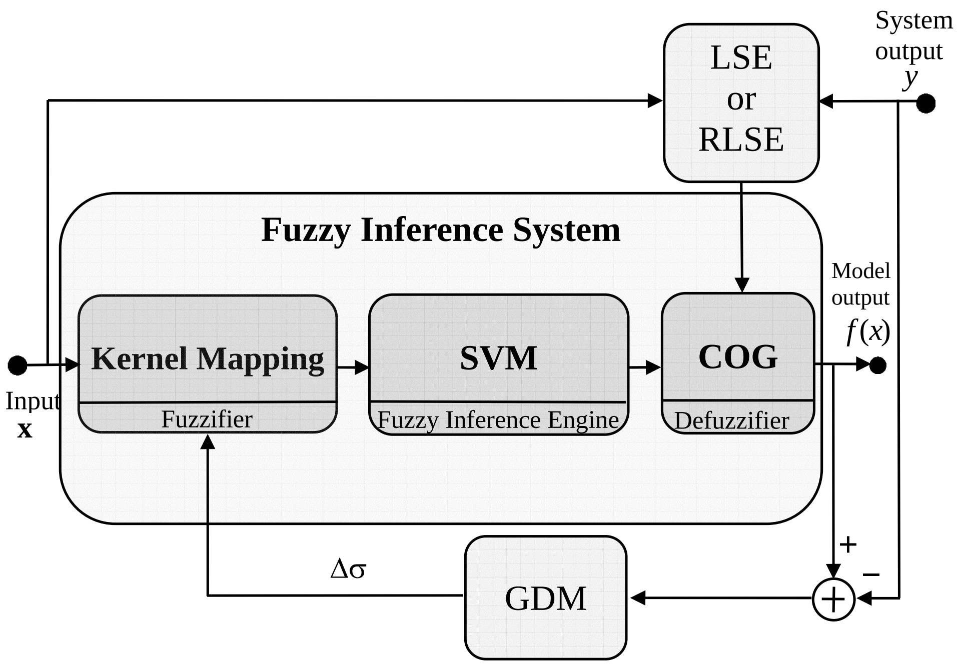


Fig. 3.5 The learning algorithm of the proposed FIS using an extended SVM

The proposed FIS is also divided into four Layers. The following function of this FIS is similar to the previous FIS using a SVM.

Layer 1 $\sim$ **Layer 3**: The functions of these Layers are equal to the previous FIS using a SVM.

Layer 4: For the overall output of the fuzzy model constructed, defuzzification using center of gravity (COG) method is performed as follows:

$$\begin{aligned} f(x) &= \frac{\sum_{i=1}^{n} K(x_i, \mathbf{x})\, f_i}{\sum_{i=1}^{n} K(x_i, \mathbf{x})}, \qquad K(x_i, x) = \mathrm{MIN}_{j=1}^{D} M_{ij}(x_{ij}), \\ &= \sum_{i=1}^{n} \beta_i \left(a_{i0} + a_{i1}x_{i1} + a_{i2}x_{i2} + \cdots + a_{iD}x_{iD}\right), \\ \text{where, } \beta_i &= \frac{K(x_i, \mathbf{x})}{\sum_{i=1}^{n} K(x_i, \mathbf{x})}. \end{aligned} \tag{3.26}$$

Table 3.1 Two passes in the learning algorithm for the proposed FIS

Forward	Backward
SVM learning	Gradient descent algorithm
LSE	

3.3.3 The learning algorithm of the FIS using an extended SVM

The learning algorithm of the FIS using an extended SVM is shown in Fig. 3.5. We present a recursive support vector learning algorithm which adjusts Gaussian kernel parameters and estimates the consequent parameters using the LSE or RLSE. It can be achieved by the following iterative procedure.

Step 1: Initialize precision ε, trade-off constant C, and kernel parameter σ_i.

Step 2: Using the following SVM algorithm, find support vectors (SVs) x_i^* that is the center c_i of Gaussian membership function.

$$\min_x \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{c}^T\mathbf{x}. \tag{3.27}$$

Step 3: Using either the LSE or RLSE [10], estimate the parameter a_{ij} of linear equation f_i in the consequent parts.

Step 4: Using a gradient descent algorithm [17], update the kernel parameter σ_i such that error is minimized.

Step 5: Go to step 2 or stop

Table 3.1 shows two passes in the learning algorithm for the proposed FIS. In step 3, because of estimating the parameter of linear equation β_i in the consequent parts, the LSE [10] or RLSE [10] starts from minimizing the squared error measure defined by

$$E = \sum_{i=1}^{n}(y_i - y_i^d)^2, \tag{3.28}$$

where y_i^d is the desired output corresponding to the i-the input $x_i = (x_{i1}, x_{i1}, ..., x_{iD})$ and y_i is the output of the constructed fuzzy model.

The output of the constructed fuzzy model can be determined by

$$\begin{aligned} y_i &= \frac{\sum_{i=1}^{n} K(x_i, x)\, f_i}{\sum_{i=1}^{n} K(x_i, x)}, \quad K(x_i, x) = \mathrm{MIN}_{j=1}^{D} M_{ij}(x_{ij}), \\ &= \sum_{i=1}^{n} \beta_i \left(a_{i0} + a_{i1}x_{i1} + a_{i2}x_{i2} + \cdots + a_{iD}x_{iD}\right). \end{aligned} \tag{3.29}$$

Let

$$\begin{aligned} &Y = \begin{bmatrix} y_1^d & y_2^d & \cdots & y_n^d \end{bmatrix}^T, \\ &A = \begin{bmatrix} a_{10} & a_{11} & \cdots & a_{1D} & \cdots\cdots & a_{n0} & a_{n1} & \cdots & a_{nD} \end{bmatrix}^T, \\ &\text{and} \\ &W = \begin{bmatrix} \beta_{11} & \beta_{11}x_{11} & \cdots & \beta_{11}x_{1D} & \cdots\cdots & \beta_{n1} & \beta_{n1}x_{11} & \cdots & \beta_{n1}x_{1D} \\ \beta_{12} & \beta_{12}x_{21} & \cdots & \beta_{12}x_{2D} & \cdots\cdots & \beta_{n2} & \beta_{n2}x_{21} & \cdots & \beta_{n2}x_{2D} \\ \vdots & \vdots & \cdots & \vdots & \cdots\cdots & \vdots & \vdots & \cdots & \vdots \\ \beta_{1n} & \beta_{1n}x_{n1} & \cdots & \beta_{1n}x_{nD} & \cdots\cdots & \beta_{nn} & \beta_{nn}x_{n1} & \cdots & \beta_{nn}x_{nD} \end{bmatrix}. \end{aligned} \tag{3.30}$$

If $(W^T W)$ is nonsingular, the parameter vector A is calculated by

$$A = (W^T W)^{-1} W^T Y. \tag{3.31}$$

Also we can apply the RLSE algorithm having on-line learning ability. Let b_k $(k = 1, 2, ..., n)$ row vector of the matrix W. Then A is recursively calculated as follows:

$$\begin{aligned} A_{k+1} &= A_k + S_{k+1} \cdot b_{k+1}^T \cdot (y_{k+1}^d - b_{k+1} \cdot A_k), \\ S_{k+1} &= S_k - \frac{S_k \cdot b_{k+1}^T \cdot b_{k+1} \cdot S_k}{1 + b_{k+1} \cdot S_k \cdot b_{k+1}^T}, \quad k = 0, 1, \cdots, n-1, \\ A_0 &= 0, \\ S_0 &= \gamma I, \end{aligned} \tag{3.32}$$

where γ is a positive large number and I is the identity matrix of dimensions $(n \cdot D + 1) \times (n \cdot D + 1)$. The consequent parameter values are determined by the recursive least-squares estimates $A = A_n$ of the algorithm.

In step 4, the kernel parameters σ_i is adjusted by minimizing given E_i,

$$E_i = (y_i - y_i^d)^2, \tag{3.33}$$

$$\begin{aligned}\text{Let } e &= y_i - y_i^d \\ &= \sum_{i=1}^{n} \beta_i \left(a_{i0} + a_{i1}x_{i1} + a_{i2}x_{i2} + \cdots + a_{iD}x_{iD}\right) - y_i^d,\end{aligned} \tag{3.34}$$

Gaussian kernel function,

$$K(x_i, x) = \exp\Big(-\frac{(x - x_i)^2}{2\sigma_i^2}\Big). \tag{3.35}$$

According to the gradient descent method [17], learning rule for adjusting kernel parameter σ_i in antecedent parts is presented as follows:

$$\Delta\sigma_i = -\eta\nabla\sigma_i E_i, \tag{3.36}$$

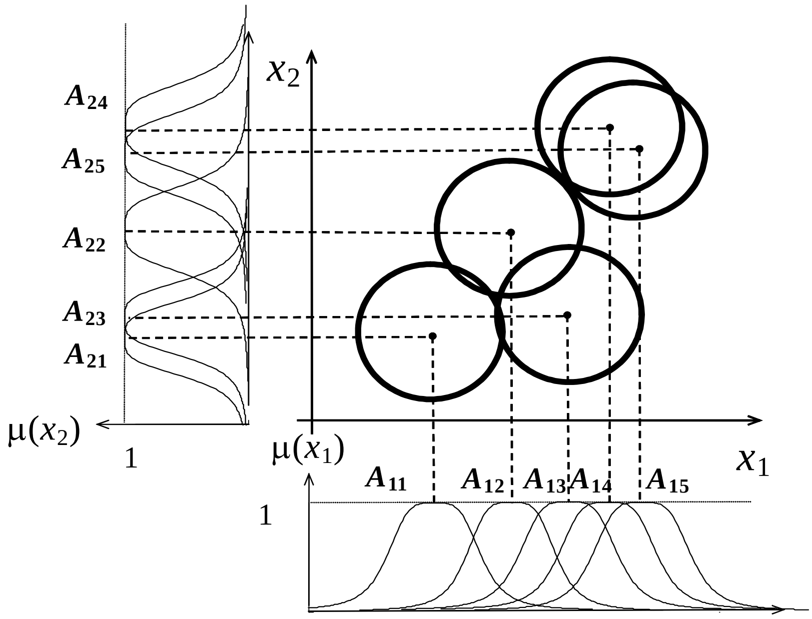
$$\nabla\sigma_i E_i = \frac{\partial E_i}{\partial\sigma_i} = -2e\beta_i K(x_i, x)\|x - x_i\|^2\sigma_i^{-3}. \tag{3.37}$$

3.3.4 The input space partition of the FIS using an extended SVM

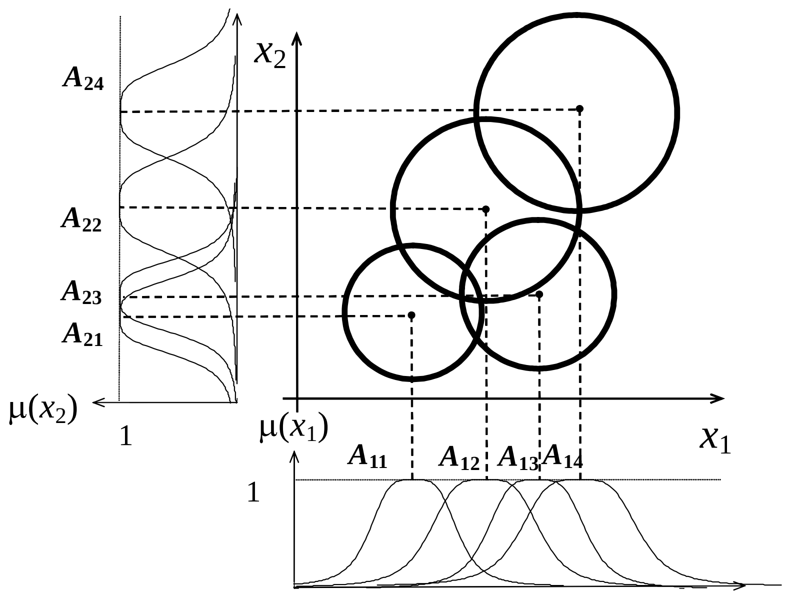
Now, we discuss the input space partitioning of proposed FIS using an extended SVM. The structure of fuzzy modeling is closely related to the partitioning of input space for fuzzy rule generation. Input space partition approach of the proposed FIS is a clustering-based method.

Figure 3.6 shows input space partition method of two-dimensional input space. Figure 3.6 (a) and (b) show the input space partitioning using the SVNN [28] with same Gaussian variance and the proposed FIS with each different Gaussian variance, respectively. Because each cluster leads to hidden layer and fuzzy rule, five hidden layers and four rules are generated in Fig. 3.6 (a) and (b), respectively. The support vector (SV) as the center of Gaussian kernel function becomes the center of Gaussian membership function. Figure 3.6 illustrates how the method using the extended SVM can reduce the number of rules and membership function. The five clusters which are generated using the SVNN with the same Gaussian variance in Fig. 3.6 (a) can be merged into four rules using the extended SVM with a different Gaussian variance σ_{ij} in Fig. 3.6 (b).

The proposed FIS through the generalization strategy of the SVM estimates the nonlinear system and determines fuzzy rules and parameters of membership functions automatically.



(a) The input space partitioning of the SVNN



(b) The input space partitioning of the proposed FIS

Fig. 3.6 The input space partitions of the SVNN and the proposed FIS

3.4 Examples

In this section, simulation results of the proposed FIS for the modeling of three nonlinear systems are described. We compare modeling results of the proposed FIS with the results of three others modeling methods such as an ordinary fuzzy system, general SVM and SVNN. The ordinary fuzzy system havs initial grid even partition without learning algorithm. It's center and variance are the mean and half a size of each partition respectively. The SVM is used the method proposed by Vapnik [21]. The SVNN with the same Gaussian variance is employed [28]. In the proposed FIS, the extended SVM is used as a learning algorithm. The Gaussian kernel function is employed as a kernel function.

The modeling error is defined as Root Mean Square Error (RMSE):

$$E = \sqrt{\frac{\sum_{k=1}^{N}(y_k - \widehat{y_k})^2}{N}}, \tag{3.38}$$

where N is the number of data, y_k and $\widehat{y_k}$ are the system and the model output.

3.4.1 Example 1: modeling of 1-input nonlinear function

The example was taken from Z. UyKan et al. [78]. The nonlinear system is as follows:

$$F_1(x) = 0.5(\sin(2\pi x/5) + \sin(2\pi x/3)). \tag{3.39}$$

From 0.1 interval point of the range $[0, 10]$ within the input space of the above function, 100 training data pairs were obtained firstly. The proposed FIS using an extended SVM for modeling of $F_1(x)$ extracts the 7 SVs, so that it has 7 fuzzy rules as follows:

$$\text{Rule i}: \text{ If } x \text{ is } M_i, \text{ Then } f_i = a_{i0} + a_{i1}x, \;\; i = 1, ..., 7, \tag{3.40}$$

where x and f_i are the input and output values, respectively.

The structure of the proposed FIS is shown in Fig. 3.7. The parameter values of antecedent and consequent parts are listed in Table 3.2. The c_{ij} and θ_{ij} are the center and variance of Gaussian membership function, respectively. The $(a_{i0},\ a_{i1})$ are the consequent parameters of the TS fuzzy model.

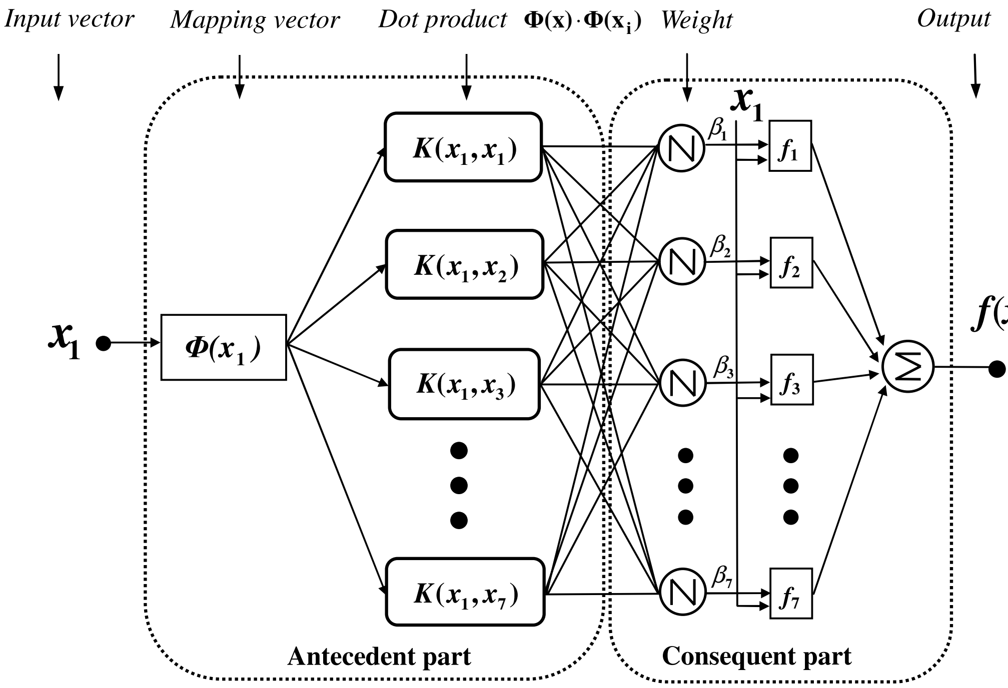
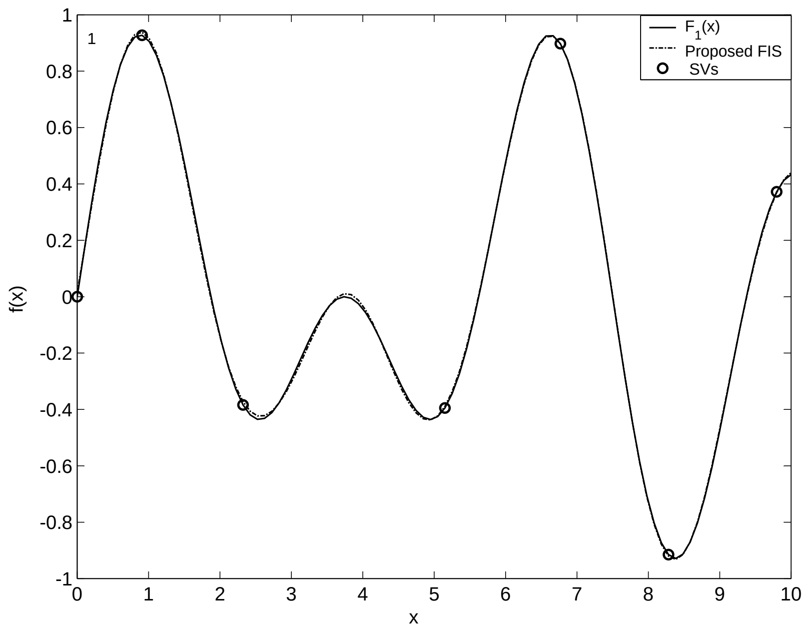


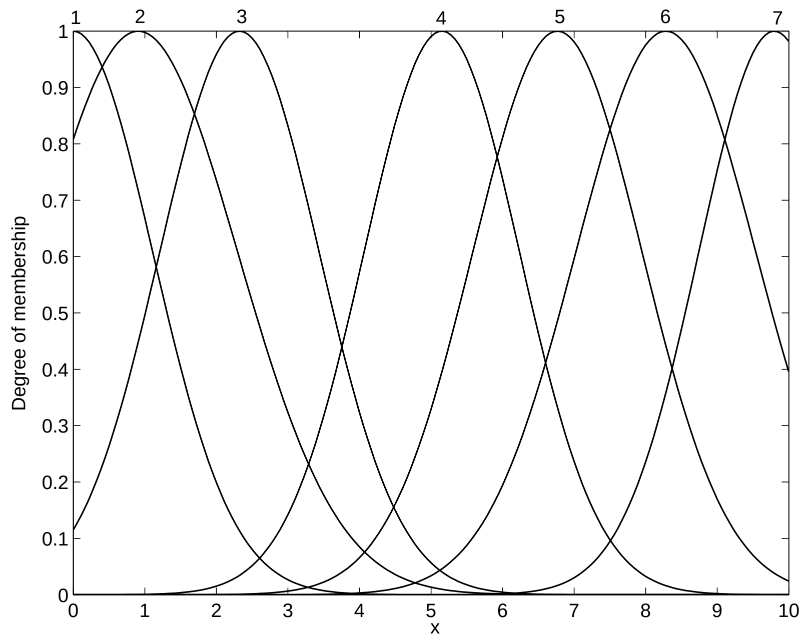
Fig. 3.7 The structure of the FIS for modeling of $F_1(x)$

Table 3.2 The parameter values of the proposed FIS for modeling of $F_1(x)$

Rule	Antecedent part		Consequent part
	c_{ij}	θ_{ij}	$(a_{i0},\ a_{i1})$
1	0	1.0024	-34.3285, -9.7038
2	0.9091	1.0589	48.1501, -23.7999
3	2.3232	1.0371	12.7341, -2.8346
4	5.1515	1.0354	9.4054, -2.4138
5	6.7677	1.0868	21.5803, -2.7576
6	8.2828	1.1129	23.3162, -3.2517
7	9.7980	1.0233	28.7624, -2.5254



(a) The output results of the proposed FIS with 7 rules for modeling of $F_1(x)$



(b) The membership functions of proposed FIS

Fig. 3.8 The output results of the proposed FIS for modeling of $F_1(x)$

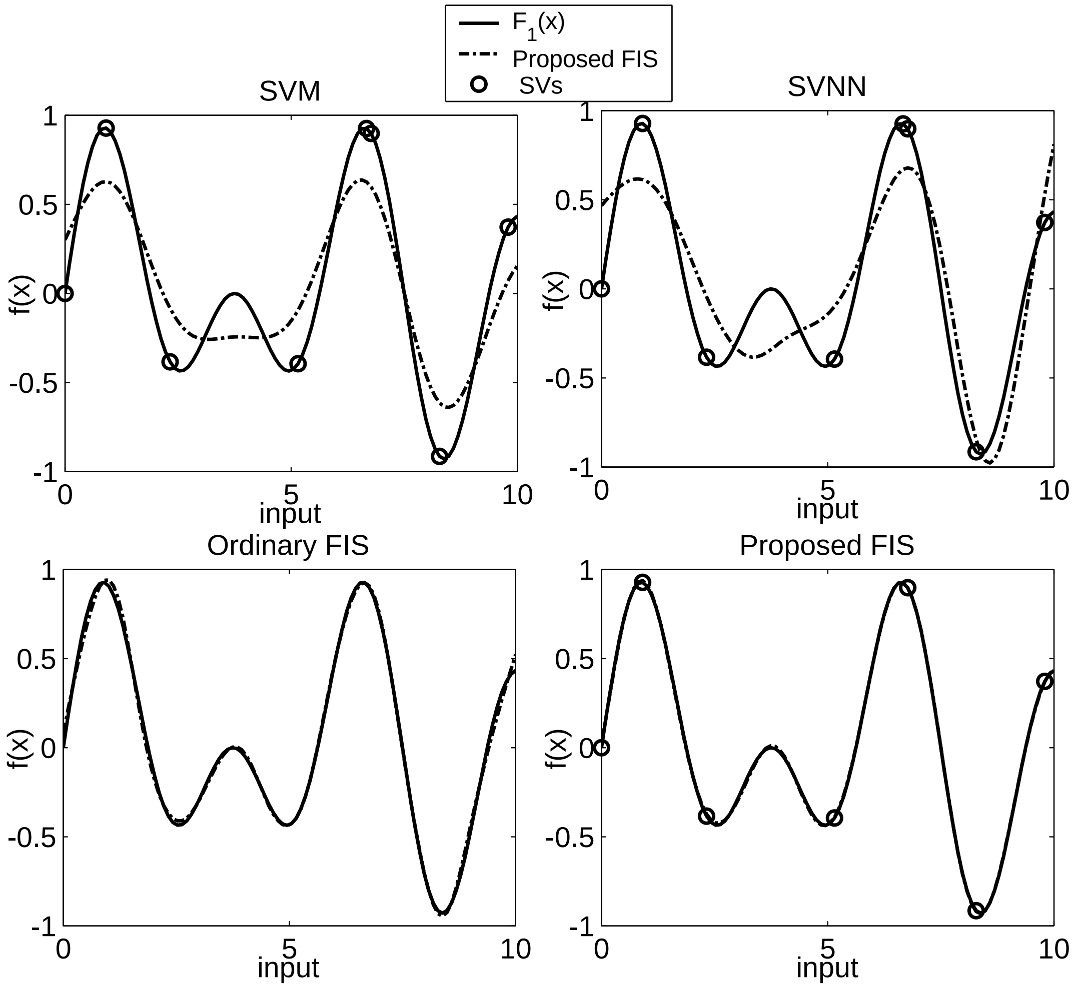


Fig. 3.9 Performance results using four algorithms for F_1 modeling

The output results of the proposed FIS and 7 support vectors (SVs) are shown in Fig. 3.8 (a). The membership functions of the proposed FIS are also shown in Fig. 3.8 (b).

The method in the literature applied to the same nonlinear function. In the initial condition of simulation, Given precision ε is 0.3 and the constant of trade-off is 300. In the SVM and SVNN, the fixed variance is 1. Figure 3.9 shows performance results for F_1 modeling using four algorithms such as the SVM, SVNN, ordinary FIS and proposed FIS. The results listed on the Table 3.3. The modeling error is the RMSE. Compared with the number of rules and modeling error, the proposed method using the extended SVM shows the smaller number of rules and modeling error than the others methods shown in Table 3.3.

Table 3.3: Compared results of modeling nonlinear function $F_1(x)$

Type	Rules(or SVs)	RMSE
SVM [21]	8	0.2098
SVNN [28]	8	0.2192
Ordinary FIS	7	0.0257
Proposed FIS	7	0.0062

3.4.2 Example 2: modeling of 2-input nonlinear function

The training data in this examples

$$F_2(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1.5})^2, \tag{3.41}$$

which was used by Ryu *et al.* [12]. From input ranges $[1, 5] \times [1, 5]$ within the input space of (3.41), 50 training data pairs were obtained firstly. The proposed FIS using an extended SVM for modeling of $F_2(x_1, x_2)$ extracts the 5 SVs, so that it has 5 fuzzy rules as follows:

$$\text{Rule i}: \text{ If } x_1 \text{ is } M_{i1}, x_2 \text{ is } M_{i2} \text{ Then } f_i = a_{i0} + a_{i1}x_1 + a_{i2}x_2, \ \ i = 1, ..., 5. \tag{3.42}$$

The structure of the proposed FIS is shown in Fig. 3.10. The parameter values of antecedent and consequent parts are listed in Table 3.4. The c_{ij} and θ_{ij} are the center and variance of Gaussian membership function, respectively. The $(a_{i0},\ a_{i1},\ a_{i2})$ are the consequent parameters of the TS fuzzy model.

In the initial condition of simulation, given precision ε is 0.7 and the constant of trade-off is 300. In the SVM and SVNN, the fixed variance is 3.2. The membership functions of the proposed FIS for modeling of $F_2(x_1, x_2)$ are shown in Fig. 3.11.

Figure 3.12 shows the output results of the proposed FIS with 5 rules for modeling of $F_2(x_1, x_2)$. To investigate the performance of the proposed FIS, the method in the literature also applied to the same nonlinear function. The comparison of our FIS with others methods is presented in Table 3.5. Compared with the number of rules and modeling error, the proposed method using the extended SVM shows the smaller number of rules and modeling error than the others methods shown in Table 3.5.

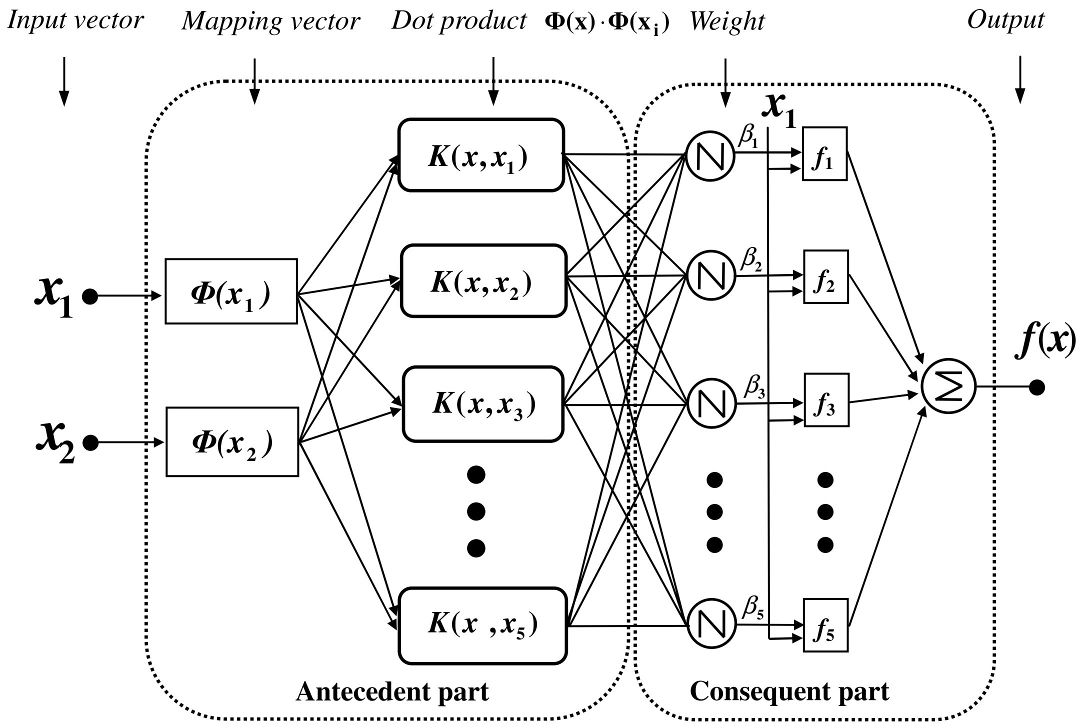
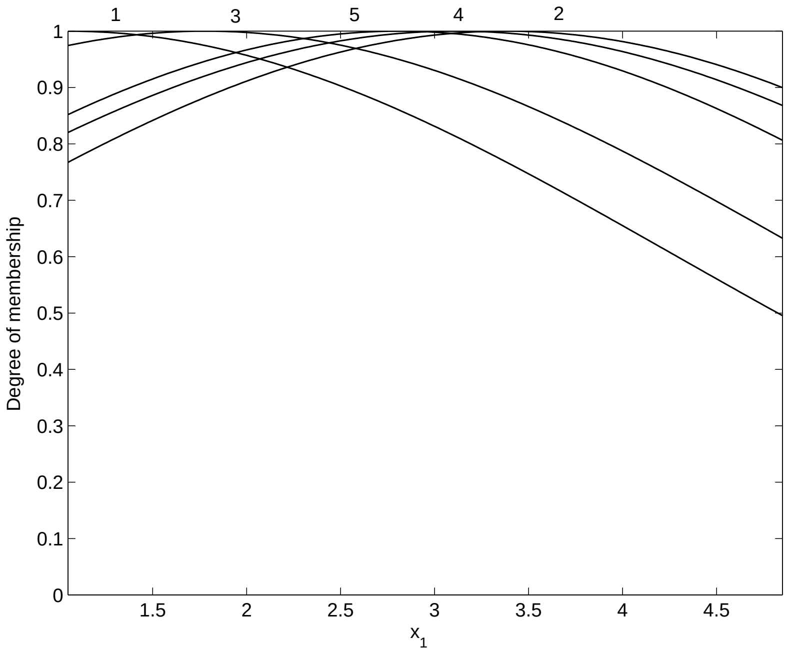


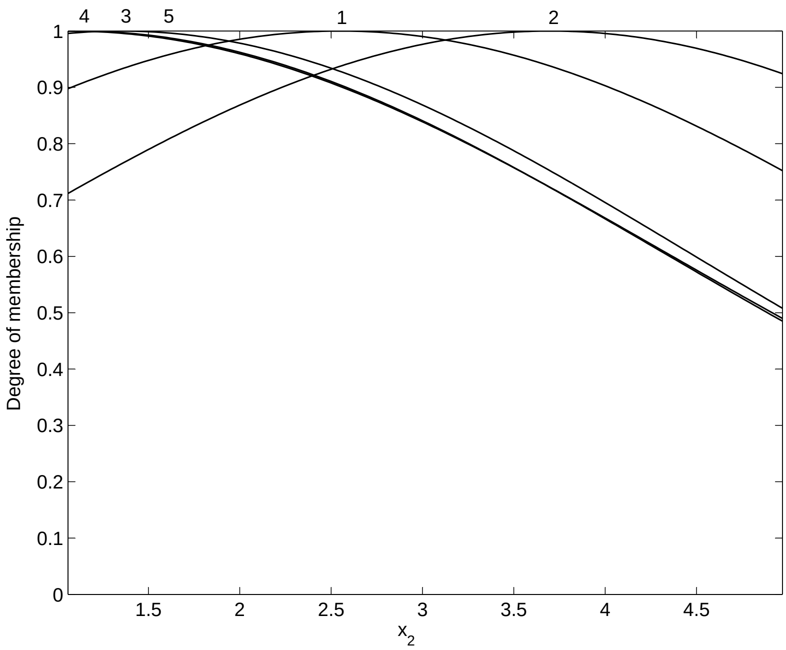
Fig. 3.10 The structure of the FIS for modeling of $F_2(x_1, x_2)$

Table 3.4 The parameter values of the proposed FIS for modeling of $F_2(x_1, x_2)$

Rule	Antecedent part		Consequent part
	c_{ij}	θ_{ij}	$(a_{i0},\ a_{i1}, a_{i2})$
1	(1.0500, 2.5500)	3.2066	597, -218, -95
2	(3.3800, 3.7000)	3.2014	-8355, 37, 507
3	(1.7800, 1.1100)	3.2087	-71756, -3509, -885
4	(3.1100, 1.0600)	3.2728	81762, -3396, 1871
5	(2.8100, 1.3500)	3.1100	17651, -2723, 200



(a) The membership functions of x_1



(b) The membership functions of x_2

Fig. 3.11 The membership functions of the proposed FIS for modeling of $F_2(x_1, x_2)$

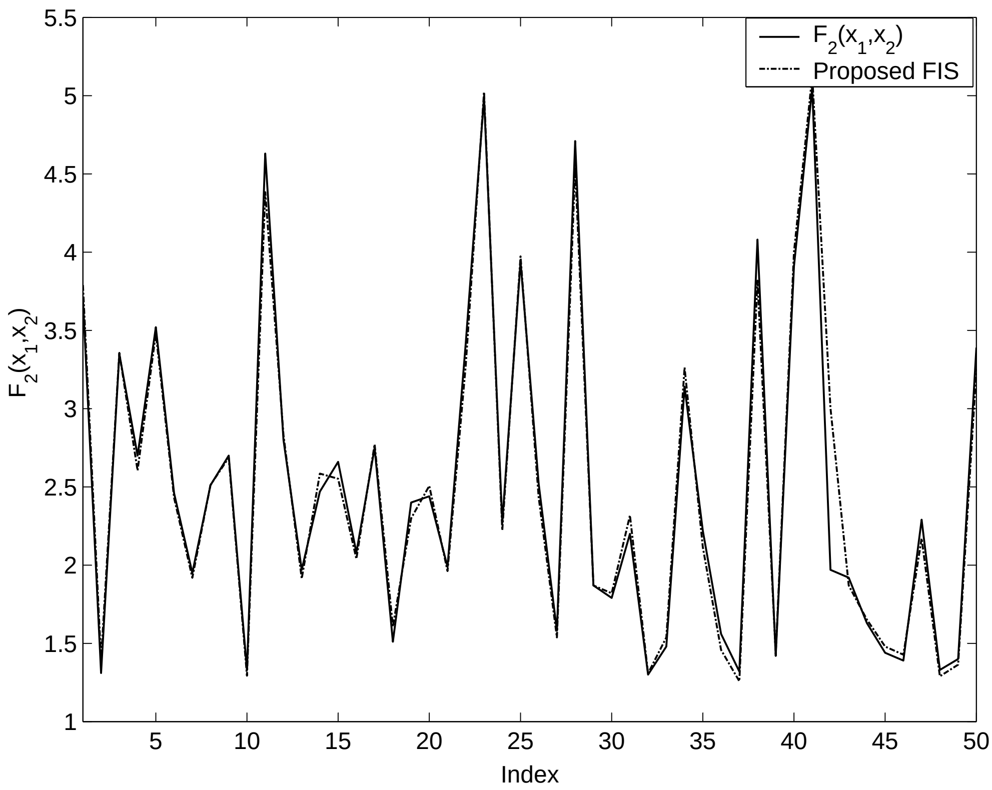


Fig. 3.12 The output results of the proposed FIS with 5 rules for modeling of $F_2(x_1, x_2)$

Table 3.5: The compared results of modeling of nonlinear function $F_2(x_1, x_2)$

Type	Rules(or SVs)	RMSE
M. Sugeno [16]	6	0.281
A.F.G.Skarmeta [79]	5	0.266
S. Kim [80]	7	0.293
SVNN [28]	6	0.324
Proposed FIS	5	0.171

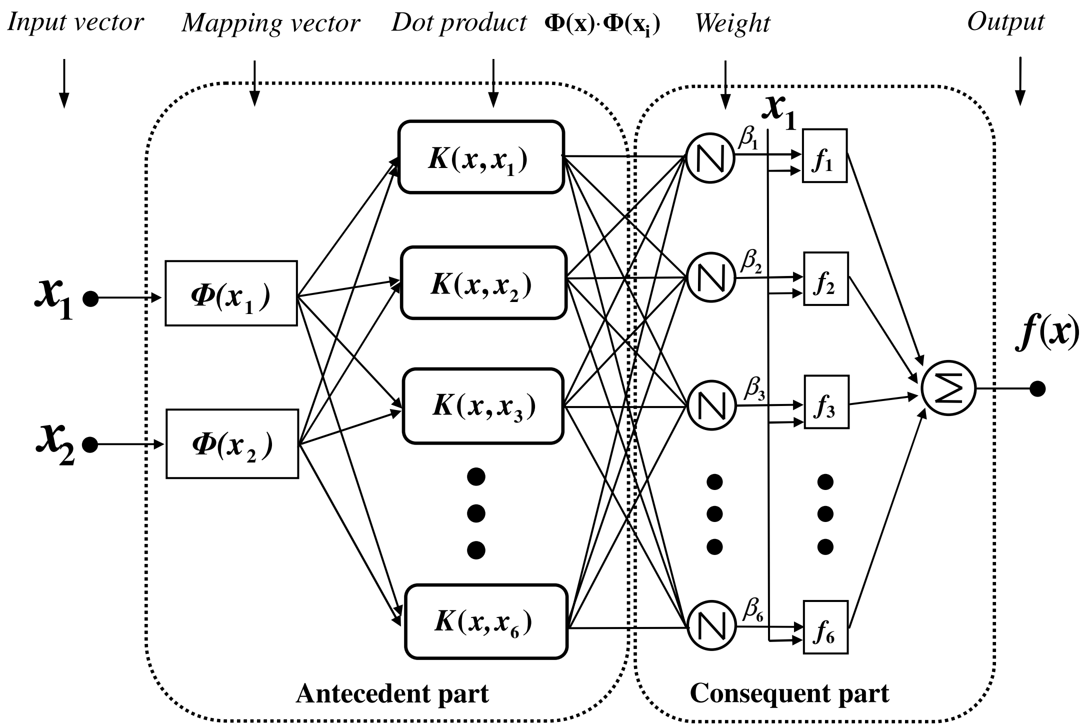


Fig. 3.13 The structure of the proposed FIS for modeling of $F_3(x_1, x_2)$

3.4.3 Example 3: modeling of 2-input nonlinear function

The third examples was taken from C. C. Wong's works [81]. The nonlinear function is as follows:

$$F_3(x_1, x_2) = \sin(\pi x_1)\sin(\pi x_2), \tag{3.43}$$

from the distributed grid points of input range $[-1, 1] \times [0, 1]$ with input space of the nonlinear function $f_3(x_1, x_2)$, $21 \times 11 = 231$ training data pairs were obtained. The proposed FIS using an extended SVM for modeling of $F_3(x_1, x_2)$ extracts the 6 SVs, so that it has 6 fuzzy rules:

$$\text{Rule i}: \text{ If } x_1 \text{ is } M_{i1}, x_2 \text{ is } M_{i2} \text{ Then } f_i = a_{i0} + a_{i1}x_1 + a_{i2}x_2, \quad i = 1, ..., 6. \tag{3.44}$$

The structure of the proposed FIS is shown in Fig. 3.13. The original nonlinear function with 6 SVs is shown in Fig. 3.14. The parameter values of antecedent and consequent parts are listed in Table 3.6. The c_{ij} and θ_{ij} are the center and

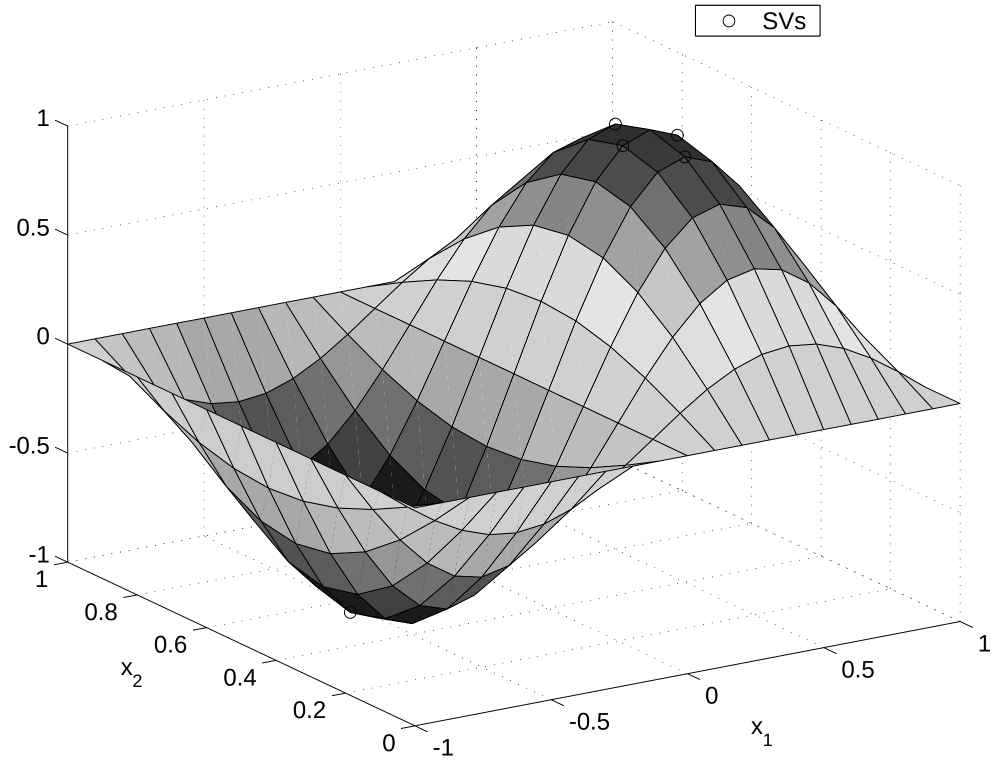


Fig. 3.14 Nonlinear function $F_3(x_1, x_2)$ with 6 SVs

variance of Gaussian membership function, respectively. The $(a_{i0},\ a_{i1},\ a_{i12})$ are the consequent parameters of the TS fuzzy model.

The output results of the proposed FIS and 6 support vectors (SVs) for modeling of $F_3(x_1, x_2)$ are shown in Fig. 3.15. The membership functions of the proposed FIS are also shown in Fig. 3.15.

The method in the literature applied to the same nonlinear function. In the initial condition of simulation, Given precision ε is 0.3 and the constant of trade-off is 300.

Figure 3.17 shows performance results for modeling of $F_3(x_1, x_2)$ using four algorithms such as the SVM, SVNN, ordinary FIS and proposed FIS. The results listed on the Table 3.7. The modeling error is the RMSE. Compared with the number of rules and modeling error, the proposed method using the extended SVM shows the smaller number of rules and modeling error than the others methods shown in Table 3.7.

Table 3.6 The parameter values of the FIS for modeling of $F_3(x_1, x_2)$

Rule	Antecedent part		Consequent part
	c_{ij}	θ_{ij}	$(a_{i0},\ a_{i1},\ a_{i12})$
1	(-0.6, 0.5)	0.4346	0.7344, 0.3549, -0.0004
2	(-0.4, 0.5)	0.2056	-1.0968, 2.5749, -0.0001
3	(0.4, 0.5)	0.2751	1.0895, 5.3986, -0.0488
4	(0.5, 0.4)	0.2062	-2.9958, 2.9944, 2.9426
5	(0.5, 0.6)	0.2056	0.1471, 3.0127, -3.1304
6	(0.6, 0.5)	0.1954	2.2016, -4.0753, -0.0395

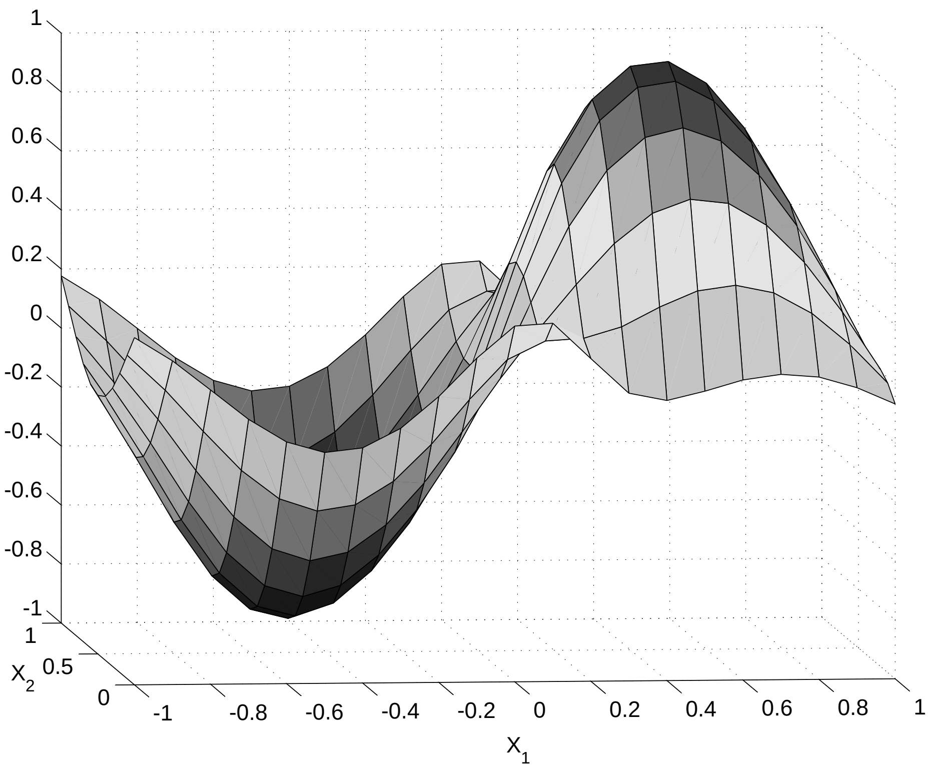
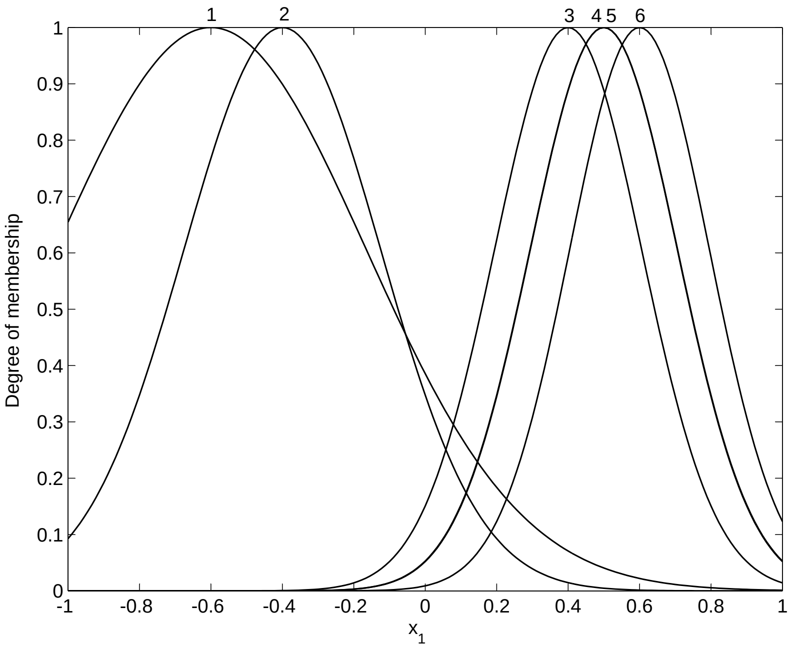
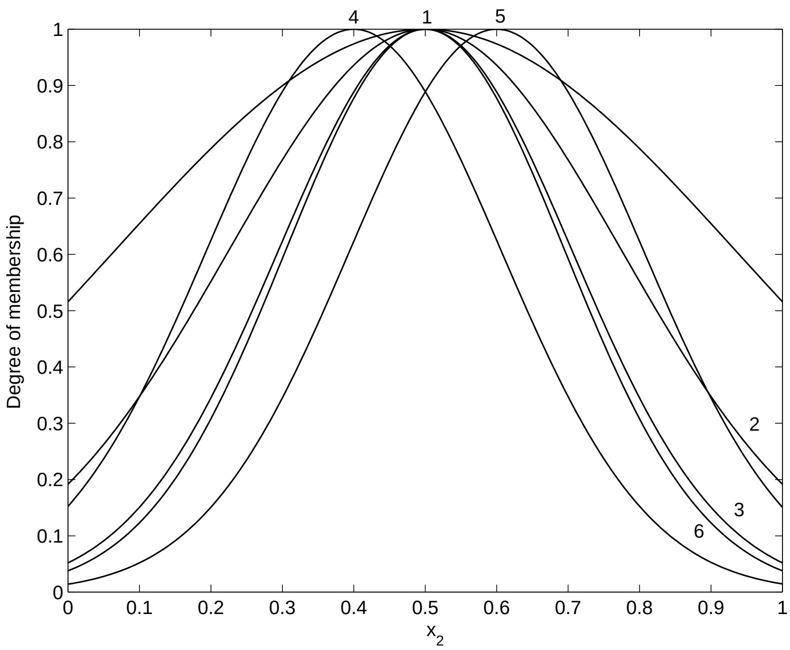


Fig. 3.15 The output results of the proposed FIS with 6 rules for modeling of $F_3(x_1, x_2)$



(a) The membership functions of x_1



(b) The membership functions of x_2

Fig. 3.16 The membership functions of the proposed FIS for modeling of $F_3(x_1, x_2)$

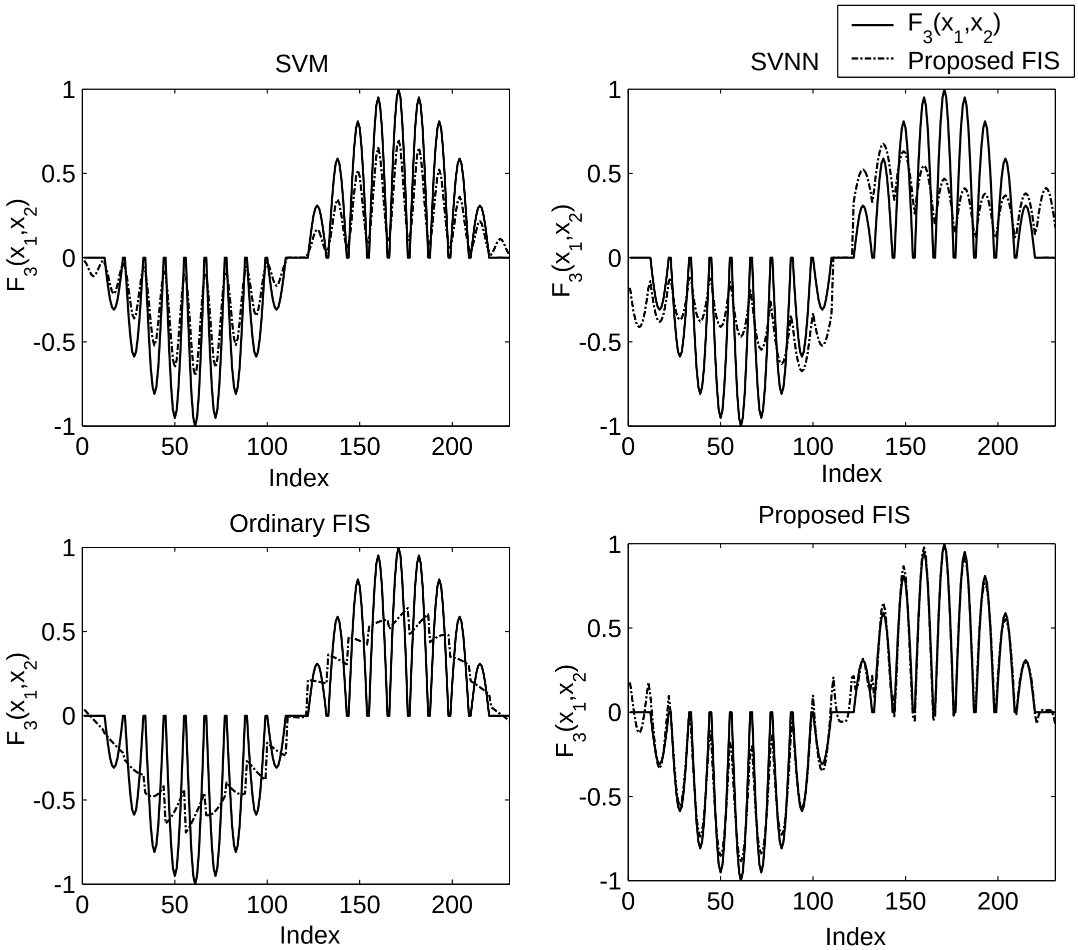


Fig. 3.17 Performance results using four algorithms for $F_3(x_1, x_2)$ modeling

Table 3.7: The compared results of modeling of nonlinear function $F_3(x_1, x_2)$

Type	Rules(or SVs)	RMSE
SVM [21]	8	0.1770
SVNN [28]	8	0.2556
Ordinary FIS	6	0.2423
Proposed FIS	6	0.0676

3.5 Discussion and Conclusions

In this chapter, the FIS based on the Takagi-Sugeno fuzzy model for modeling of nonlinear systems was presented using the extended SVM.

Our main concern is to determine the best structure of the TS fuzzy model from the given input-output data of the particular system. Conventional neuro-fuzzy modeling methods from input-output data are based on sequential design methods of structure identification and parameter identification or clustering methods with either the number of clusters or candidates for cluster centers in advance. By contrast, the proposed FIS automatically decides not only the number of simplified fuzzy inference system rules but also the parameter values. The FIS can linearly analyze a given complex data by performing nonlinear mapping which projects input space into high dimensional feature space and has good generalization by considering both model complexity and approximation error. The structure of the proposed FIS is obtained by minimizing a constrained quadratic programming problem for a given error bound and the number of FIS rules can be reduced by adjusting the parameter values of membership function using the gradient descent method. After the structure is selected, the parameter values in the consequent part of TS fuzzy model are determined by the least square estimation method or the recursive least square estimation algorithm.

We applied the proposed method to several nonlinear functions. The proposed FIS showed the better performance to model nonlinear systems than other methods. However, future work should include the method of choosing the proper error bound from given data as well as the choice problem of the best kernel function and the speed problem consumed for solving the quadratic programming problem.

CHAPTER 4

Fuzzy Inference System Using an Extended FVS

This chapter presents a new approach to fuzzy inference system (FIS) for modeling nonlinear systems based on measured input and output data. The structure of fuzzy model is obtained using an extended Feature Vector Selection (FVS) algorithm based on the kernel method. In the suggested FIS, the number of fuzzy rules and parameter values of membership functions are automatically decided using the extended FVS. The extended FVS method individually performs linear transformation and kernel mapping. Linear transformation projects input space into linearly transformed input space. Kernel mapping projects linearly transformed input space into high dimensional feature space. Especially, the process of linear transformation is needed in order to solve difficulty determining the type of kernel function which presents the nonlinear mapping corresponding to nonlinear system. The structure of the proposed fuzzy inference system is equal to a Takagi-Sugeno (TS) fuzzy model whose input variables are weighted linear combinations of input variables. In addition, the number of fuzzy rules can be reduced by adjusting linear transformation matrix and parameter values of kernel functions using the gradient descent method. Once a structure is selected, coefficients in consequent part are determined by the least square estimation method.

4.1 Introduction

The Fuzzy inference system (FIS) has been shown powerful capability for the modeling of nonlinear systems [10] [16]. FIS can be directly obtained either from human experts using knowledge experiments or learning machine methods using numeric data. For complex and uncertain systems, FIS based only on human experts may not lead to sufficient accuracy. Because of this reason, neuro-fuzzy modeling which acquires knowledge from a set of input-output data has been actively investigated [1]. The important concerns of neuro-fuzzy modeling for the real system are how to determine the proper number of fuzzy rules and parameter values of membership functions. Many methods have been developed as illustrated in Chapter 1.

Recently, kernel-based methods have been popularly developed in classification and regression. Kernel techniques offer an alternative solution by mapping the data into high dimensional feature space to increase the computational power. Particularly, Support Vector Machine (SVM)[21] has been used in order to automatically find the number of network nodes or fuzzy rules based on given error bound [7] [28] [75] . The Support Vector Neural Network (SVNN) is proposed to select the best structure of radial based function network for the given precision [28]. The SVM is suggested to improve the simplified fuzzy inference system for the fuzzy neural network [75]. The Support Vector Fuzzy Inference System (SVFIS) is proposed to find the reduced number of rules using gradient descent method updating kernel parameters [7]. However, because the general support vector learning methodology is used in above all, they have computational complexity for solving the quadratic problem in optimization process and problem for determining the type of kernel function corresponding with nonlinear system.

In this chapter, we propose a new approach to fuzzy modeling using an extended Feature Vector Selection (FVS). The linear transformation of input variables is used to solve problem determining the exact type of the kernel function. Therefore input variables of the proposed FIS become input variables of the Takagi-Sugeno (TS) fuzzy model which are the weighted linear combinations of the input variables. The structure of fuzzy model is obtained using FVS algorithm based on the kernel method. Unlikely the SVM having computational complexity, the FVS performs a simple computation optimizing a given criterion into the feature space. The FVS algorithm is to select a basis of the data subspace in feature space. A basis of the data subspace is called a *feature vector* (FV). Ultimately, this feature vector becomes the center of the membership function. Kernel functions mapping the linearly transformed data into feature space become membership functions. In addition, the number of fuzzy rules can be reduced under the condition of optimizing a given criterion by adjusting the linear transformation matrix and parameter values of kernel functions using the gradient descent method. Once a structure is selected, coefficients in consequent part of the modified TS fuzzy model are determined by the least square estimation method. So we can automatically determine the fuzzy model using the iterative procedure which involve linear transformation, kernel mapping and FVS method under optimizing a given criterion.

4.2 Feature Vector Selection (FVS)

The FVS [26] is based on kernel method. The FVS technique is to select feature vector being a basis of data subspace and capturing the structure of the entire data into feature space F.

The FVS for estimating the mapping $\hat{\phi}_i$ of any vector x_i is as follows:

$$\hat{\phi}_i = \Phi_S \cdot \mathbf{a}_i, \tag{4.1}$$

where the mapping of each vector x_i is noted $\phi(x_i) = \phi_i$ for $1 \leq i \leq M$, the selected vectors x_{s_j} into feature space F is noted $\phi(x_{s_j}) = \phi_{S_j}$ for $1 \leq j \leq L$, $\Phi_S = \{\phi_{S_i}, ..., \phi_{S_L}\}$ is the matrix of the selected vectors $S = \{x_{s_1}, ..., x_{s_L}\}$ into F and $\mathbf{a}_i = [a_i^1, ..., a_i^L]^T$ is the associated weight vector.

The feature vector (FV) is obtained from process finding the weights vector $\mathbf{a}_i$. The weights vector is given by minimizing the following normalized Euclidean distance in feature space.

$$\delta_i = \frac{\|\phi_i - \hat{\phi}_i\|^2}{\|\phi_i\|^2}. \tag{4.2}$$

The minimum of (4.2) for a given S can be expressed over all vector as follows:

$$\min_S \sum_{x_i \in X} \Big(1 - \frac{K_{si}^t K_{ss}^{-1} K_{si}}{K_{ii}}\Big), \tag{4.3}$$

where $K_{ss} =< \Phi_S \cdot \Phi_S >$ is a kernel matrix which is the dot product of the selected vectors, $K_{si} =< \Phi_S \cdot \phi_i >$ is a kernel matrix which is the dot product of between x_i and the selected vectors and $K_{ii} =< \phi_i \cdot \phi_i >$ is a kernel matrix which is the dot product of x_i.

The fitness function is defined as follows:

$$J_S = \frac{1}{M} \sum_{x_i \in X} \Big(\frac{K_{si}^t K_{ss}^{-1} K_{si}}{K_{ii}} \Big). \tag{4.4}$$

Thus (4.3) can be rewritten by

$$\max_S \; J_S, \tag{4.5}$$

where $\max_S J_S$ is a value between 0 and 1 for $x_i \in S$.

The FVS algorithm is an iterative process which performs sequential forward selection until the fitness reaches a given value. In this iterative process, when the calculated fitness reaches the max fitness, the vector from training data is called *feature vector* (FV).

Once the FV is selected, the output of FVS is calculated using a kernel function approximation algorithm. Figure 4.1 shows the architecture of kernel function approximation procedure.

Let us suppose that we have given input and output data

$$(x_1, y_1),\ (x_2, y_2),\ ...,\ (x_M, y_M). \tag{4.6}$$

The transformation of input data x_i is given by the inner product projection as follows:

$$\begin{aligned} z_i &= \Phi_S \cdot \phi_i, \\ &= K_{si}, \end{aligned} \tag{4.7}$$

where kernel matrix K_{si} is the dot product of nonlinear mapping between input data x_i and the selected FV.

The output of kernel function approximation is obtained using the Moore-Penrose pseudo-inverse method as follows:

$$\hat{y}_i = z_i^T A + \beta^T, \tag{4.8}$$

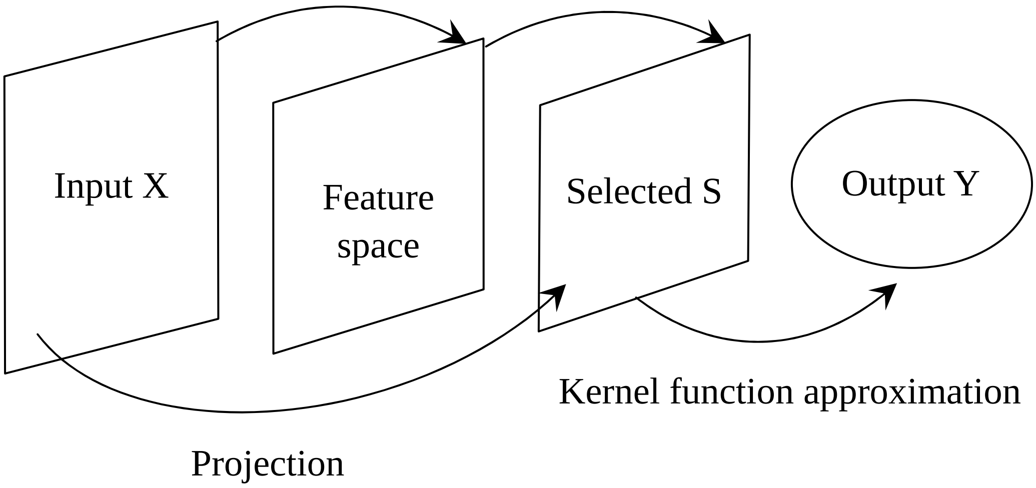


Fig. 4.1 The architecture of kernel function approximation procedure

where $A = (Z^T Z)^{-1} Z^T Y$, $z_i = K_{si}$, $Y = y_i$ and β is a vector that can be included in the estimation of A by adding a constant component in each vector z_i.

The brief summary of the iterative procedure in the FVS is described as follows:

1. Select the type of kernel function and initialize kernel parameter σ^2.

2. Compute fitness.

$$\max_S J_S = \max_S \frac{1}{M} \sum_{x_i \in X} \Big(\frac{K_{si}^t K_{ss}^{-1} K_{si}}{K_{ii}} \Big). \tag{4.9}$$

3. Go to step 2 until the fitness or the number of feature vector are satisfied with given conditions.

4. Find FVs and complete the structure of the FVS.

The main motivation of approach to FVS is that the structure of the FVS is automatically found based on optimizing the normalized Euclidean distance in feature space. The found structure of the FVS has close relation to that of fuzzy rule-base.

4.3 New Fuzzy Inference System Using an Extended FVS

This section describes the structure and learning algorithm of a new fuzzy inference system using an extended FVS.

4.3.1 The structure of the FIS using an extended FVS

The kernel method using an extend FVS is that linear transformation is added to kernel mapping in order to solve the problem selecting the type of kernel function corresponding to nonlinear system. Thus, input variables of the proposed FIS become input variables of the TS fuzzy model which are weighted linear combinations of original input variables.

Suppose we have given input and output data

$$(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2),\ ...,\ (\mathbf{x}_l, \mathbf{y}_l) \tag{4.10}$$

where $\mathbf{x}_i$=$[x_1^i, x_2^i, ..., x_D^i]^T (i = 1, 2, ..., l)$ is original input variable and Y=$[\mathbf{y}_1, ..., \mathbf{y}_l]^T$ is output data. The proposed TS fuzzy model with fuzzy If-Then rules can be represented as follows:

$$\begin{aligned}
R_1 \quad &: \quad \text{If } \bar{x}_1 \text{ is } K(\bar{x}_1, \bar{x}_{11}^*) \text{ and ... } \bar{x}_D \text{ is } K(\bar{x}_D, \bar{x}_{1D}^*),\\
&\quad\ \text{Then } f_1 = a_{10} + a_{11}\bar{x}_1 + ... + a_{1D}\bar{x}_D\\
R_2 \quad &: \quad \text{If } \bar{x}_1 \text{ is } K(\bar{x}_1, \bar{x}_{21}^*) \text{ and ... } \bar{x}_D \text{ is } K(\bar{x}_D, \bar{x}_{2D}^*),\\
&\quad\ \text{Then } f_2 = a_{20} + a_{21}\bar{x}_1 + ... + a_{2D}\bar{x}_D\\
&\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots\\
R_n \quad &: \quad \text{If } \bar{x}_1 \text{ is } K(\bar{x}_1, \bar{x}_{n1}^*) \text{ and ... } \bar{x}_D \text{ is } K(\bar{x}_D, \bar{x}_{nD}^*),\\
&\quad\ \text{Then } f_n = a_{n0} + a_{n1}\bar{x}_1 + ... + a_{nD}\bar{x}_D,
\end{aligned} \tag{4.11}$$

where n is the number of fuzzy rules, D is the dimension of input variables, $\bar{x}_j (j = 1, 2, ..., D)$ is a linearly transformed input variable, f_i is a local output variable, $K(\bar{x}_j, \bar{x}_{ij}^*)$ $(i = 1, 2, ..., n, j = 1, 2, ..., D)$ is a fuzzy set and $a_{ij} (i = 1, 2, ..., n, j = 0, 1, ..., D)$ is a consequent parameter.

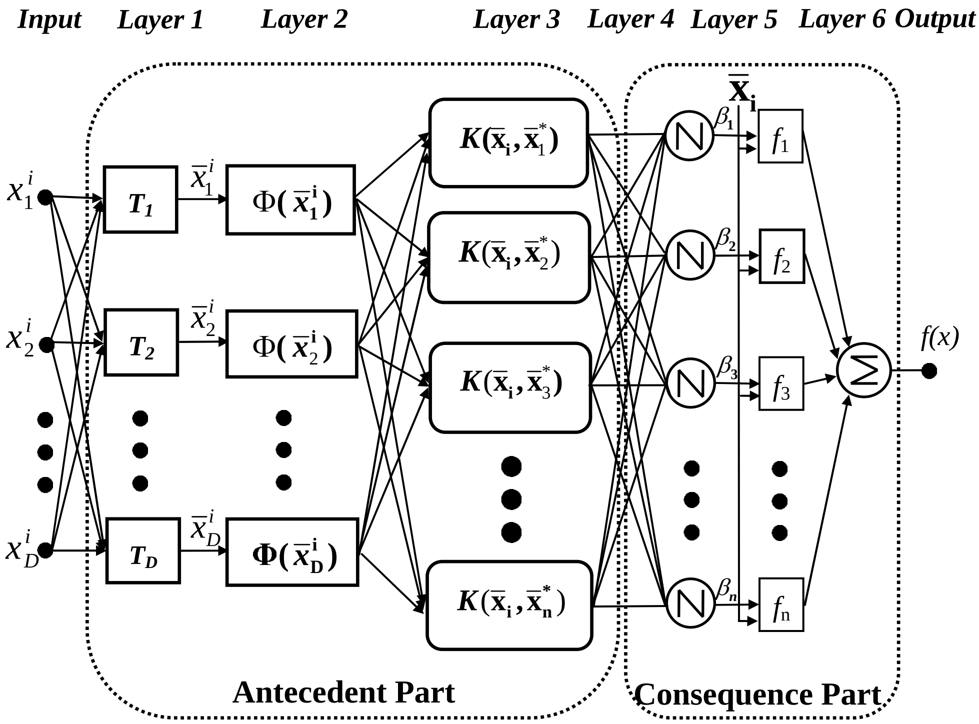


Fig. 4.2 The structure of the proposed FIS using an extended FVS

Linearly transformed input variables are defined as follows:

$$\begin{bmatrix} \bar{x}_1^i \\ \bar{x}_2^i \\ \vdots \\ \bar{x}_D^i \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1D} \\ t_{21} & t_{22} & \dots & t_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ t_{D1} & t_{D2} & \dots & t_{DD} \end{bmatrix} \begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_D^i \end{bmatrix} \tag{4.12}$$

where $\bar{\mathbf{x}}_i$=$[\bar{x}_1^i, \bar{x}_2^i, ..., \bar{x}_D^i]^T(i$=$1, 2, ..., l)$ is a linearly transformed input variable, and $\mathbf{T}_i$=$[t_{i1}, t_{i2}, ..., t_{iD}](i = 1, 2, ..., D)$ is the ith transformed direction unit vector of the original input space. Now, we describe the structure of FIS using an extended kernel method. It consists of six layers as shown in Fig. 4.2.

The four layers involved in the proposed FIS are as follows:

Layer 1: Input space is projected into a linearly transformed input space by a linearly transformation matrix.

$$\bar{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i, \quad i = 1, 2, ..., l, \tag{4.13}$$

where T=$[T_1, T_2, ..., T_D]^T$ is a linear transformation matrix.

Layer 2: Linearly transformed input space is nonlinearly mapped into feature space by a map Φ.

$$\begin{aligned} \overline{\mathbf{x}}_i &= (\bar{x}_1^i, ..., \bar{x}_D^i) \mapsto \\ \Phi(\overline{\mathbf{x}}_i) &= (\Phi_1(\overline{\mathbf{x}}_i), ..., \Phi_D(\overline{\mathbf{x}}_i)),\ i = 1, 2, ..., l. \end{aligned} \tag{4.14}$$

Layer 3: Feature Vector (FV) is determined from a FVS algorithm a using kernel method. Kernel method is a dot product which is computed with the nonlinear mapped input $\Phi(\overline{\mathbf{x}}) = (\Phi(\overline{\mathbf{x}}_1), ..., \Phi(\overline{\mathbf{x}}_l))$ and feature vector $\Phi(\overline{\mathbf{x}}_i^*) = (\Phi_1(\overline{\mathbf{x}}_i^*), ..., \Phi_D(\overline{\mathbf{x}}_i^*))(i = 1, ..., n)$, where $\overline{\mathbf{x}}_i^* = [\bar{x}_{i1}^*, \bar{x}_{i1}^*, ..., \bar{x}_{iD}^*]^T$ is the subset of the input $\overline{\mathbf{x}}$. Dot product $\Phi(\overline{\mathbf{x}}) \cdot \Phi(\overline{\mathbf{x}}_i^*)$ corresponds to evaluating kernel function $K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*)$. The Gaussian kernel function with each variance σ_i is used as follows:

$$K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*) = \exp\Big(- \frac{(\overline{\mathbf{x}} - \overline{\mathbf{x}}_i^*)^2}{2\sigma_i^2} \Big), i = 1, 2, ..., n \tag{4.15}$$

where $\overline{\mathbf{x}}_i^*$ is a FV, σ_i is called a kernel parameter and n is the number of FVs. This kernel function becomes a Gaussian membership function in the proposed FIS. $\overline{\mathbf{x}}_i^*$ and σ_i are the center and the variance of the i-th Gaussian membership function, respectively. FVS algorithm is a fuzzy inference engine determining the number of fuzzy rules.

The Layer 1 to 3 are related to the antecedent part of the FIS.

Layer 4: The fuzzy intersection of Gaussian kernel functions is calculated. The following algebraic product operator as T-norm operator for each Layer4 node is used,

$$K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*) = \prod_j^D K(\bar{x}_j, \bar{x}_{ij}^*), \tag{4.16}$$

where, $\overline{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, ..., \bar{x}_D]$ is the i-th input variable vector, $\overline{\mathbf{x}}_i^* = [\bar{x}_{i1}^*, \bar{x}_{i2}^*, ..., \bar{x}_{iD}^*]$ is the FV of the i-th input variable.

The normalized weight β_i for each fuzzy rule (node) is computed as follows:

$$\beta_i = \frac{K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*)}{\sum_{j=1}^n K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_j^*)}, \tag{4.17}$$

where

$$K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*) \geq 0,\ \sum_{j=1}^n K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_j^*) > 0,\ i = 1, ..., n. \tag{4.18}$$

Layer 5: The normalized weight β_i of each node is multiplied by i-th local output variable f_i. Each node output $\beta_i f_i$ as shown in Fig. 4.2 is described as follows:

$$\beta_i f_i = \frac{K(\bar{\mathbf{x}}, \bar{\mathbf{x}}_i^*)(a_{i0} + a_{i1}\bar{x}_1 + \cdots + a_{iD}\bar{x}_D)}{\sum_{j=1}^{n} K(\bar{\mathbf{x}}, \bar{\mathbf{x}}_j^*)}, \tag{4.19}$$

where, $f_i = a_{i0} + a_{i1}\bar{x}_1 + \cdots + a_{iD}\bar{x}_D$ is the i-th local output variable of TS fuzzy model.

Layer 6: For the overall output of the fuzzy model constructed, defuzzification using the Center Of Gravity (COG) method is performed. Each node corresponds to one output variable $f(\mathbf{x})$,

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{n} \beta_i f_i, \\ &= \frac{\sum_{i=1}^{n} K(\bar{\mathbf{x}}, \bar{\mathbf{x}}_i^*)(a_{i0} + a_{i1}\bar{x}_1 + \cdots + a_{iD}\bar{x}_D)}{\sum_{j=1}^{n} K(\bar{\mathbf{x}}, \bar{\mathbf{x}}_j^*)}. \end{aligned} \tag{4.20}$$

The Layer 4, 5 and 6 connect with the consequent part of the proposed FIS.

4.3.2 The learning algorithm of the FIS using an extended FVS

The learning algorithm of the FIS using an extended FVS is shown in Fig. 4.3. It can be achieved by the following iterative procedure.

Step 1: Assign the desired fitness and initialize the linear transformation matrix T and the kernel parameter σ_i.

Step 2: Perform linear transformation in (4.13) in order to project input space into linearly transformed input space.

Step 3: Using the following FVS algorithm based on kernel mapping, find FVs $\bar{\mathbf{x}}_i^*$ that are the centers $\mathbf{c}_i$ of Gaussian membership functions.

$$\max_S J_S = \max_S \frac{1}{M} \sum_{\bar{\mathbf{x}}_i \in F} \left(\frac{K_{si}^t K_{ss}^{-1} K_{si}}{K_{ii}} \right) \tag{4.21}$$

Step 4: Using the following Least Square Estimation (LSE) method [10], estimate the parameter a_{ij} of the linear equation f_i.

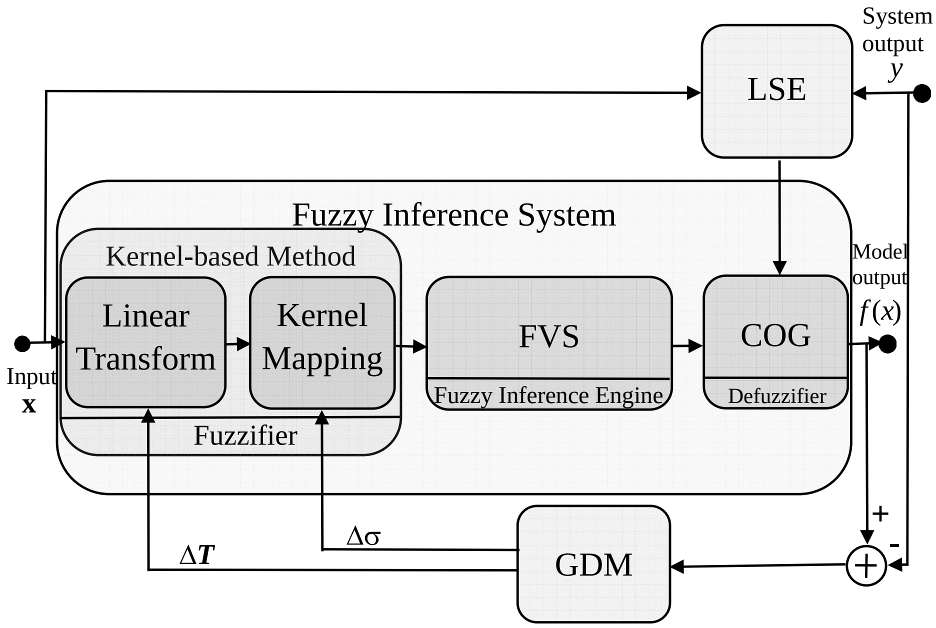


Fig. 4.3 The learning algorithm of the proposed FIS

Let

$$
\begin{aligned}
A &= \begin{bmatrix} a_{10} & a_{11} & \cdots & a_{1D} & \cdots\cdots & a_{n0} & a_{n1} & \cdots & a_{nD} \end{bmatrix}^T, \\
W &= \begin{bmatrix} \beta_1^1 & \beta_1^1\overline{x}_1^1 & \cdots & \beta_1^1\overline{x}_D^1 & \cdots\cdots & \beta_n^1 & \beta_n^1\overline{x}_1^1 & \cdots & \beta_n^1\overline{x}_D^1 \\ \beta_1^2 & \beta_1^2\overline{x}_1^2 & \cdots & \beta_1^2\overline{x}_D^2 & \cdots\cdots & \beta_n^2 & \beta_n^2\overline{x}_1^2 & \cdots & \beta_n^2\overline{x}_D^2 \\ \vdots & \vdots & \cdots & \vdots & \cdots\cdots & \vdots & \vdots & \cdots & \vdots \\ \beta_1^l & \beta_1^l\overline{x}_1^l & \cdots & \beta_1^l\overline{x}_D^l & \cdots\cdots & \beta_n^l & \beta_n^l\overline{x}_1^l & \cdots & \beta_n^l\overline{x}_D^l \end{bmatrix}.
\end{aligned} \tag{4.22}
$$

where $\beta_i^j = \frac{K(\overline{\mathbf{x}}_j, \overline{\mathbf{x}}_i^*)}{\sum_{k=1}^n K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_k^*)}$.

Thus fuzzy model output is $f(\mathbf{x}) = WA$. If (W^TW) is nonsingular, the parameter vector A is calculated by

$$A = (W^TW)^{-1}W^TY. \tag{4.23}$$

Step 5: Using a Gradient Descent Method (GDM) [17], update the kernel parameter σ_i such that error is minimized. From the definition of the GDM,

$$
\begin{aligned}
\Delta\sigma_i &= -\eta_\sigma \nabla_{\sigma_i} E, \\
&= -2\eta_\sigma \sigma_i^{-3} \sum_{j=1}^{l} e_j \beta_i^j (f_i - y_j) \|\overline{\mathbf{x}}_j - \overline{\mathbf{x}}_i^*\|^2,
\end{aligned} \tag{4.24}
$$

where η_σ is the learning rate of σ_i, $e_j = f(\mathbf{x}_j) - y_j$ and $E = \sum_{j=1}^{l} e_j^2$.

Step 6: Also using the following GDM, update the linear transformation matrix T and go to step 2 until error and FVs are satisfied with given conditions.

$$\begin{aligned} \Delta T &= -\eta_T \nabla_T E, \\ &= -2\eta_T \sum_{j=1}^{l} e_j \bar{\mathbf{x}}_j \sum_{i=1}^{n} \beta_i^j \left[A_i + \|\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i^*\| \sigma_i^{-2} (y_j - f_i) \right], \end{aligned} \tag{4.25}$$

where η_T is the learning rate of T and $A_i = [a_{i1}, ..., a_{iD}]^T$.

4.3.3 The input space partition of the FIS using an extended FVS

In this section, the input space partitioning technique of the FIS using an extended FVS is presented. The input space partition approach of the proposed FIS is cluster-based fuzzy rule generation method. The extended FVS consists of the linear transformation part of input variables and the kernel mapping part. The linear transformation of input variables is proposed to solve problem selecting the best shape of the Gaussian kernel function which presents the nonlinear mapping.

Now, we introduce the linear transformation of input variables and input space partitioning technique of the proposed FIS.

Linear transformation of input variables

Consider the following two-dimensional linear transformation,

$$\bar{\mathbf{x}} = \mathbf{T}\mathbf{x}, \tag{4.26}$$

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{4.27}$$

where x_1 and x_2 are original input variables, $\bar{x}_1$ and $\bar{x}_2$ are transformed input variables and t_{11}, t_{12}, t_{21} and t_{22} are linear transformation parameters in linear matrix $\mathbf{T}$.

Figure 4.4 shows the linear transformation of two-dimensional input variables. In Fig. 4.4 (a), any input variables are illustrated. In Fig. 4.4 (b), the transformed input variables are described. From Fig. 4.4 (a) to Fig. 4.4 (b), linear transformation matrix is presented as follows:

$$\mathbf{T} = \begin{bmatrix} 1.0417 & -0.2083 \\ -0.2083 & 1.0417 \end{bmatrix}. \tag{4.28}$$

In Fig. 4.4 (a), the ellipsoids of four groups are illustrated. Figure 4.4 (b) shows the results that the ellipsoids of four groups are transformed to circles of four groups. This result shows that appropriate linear transformation can help the effective input space partition of the extended FVS with Gaussian kernel functions.

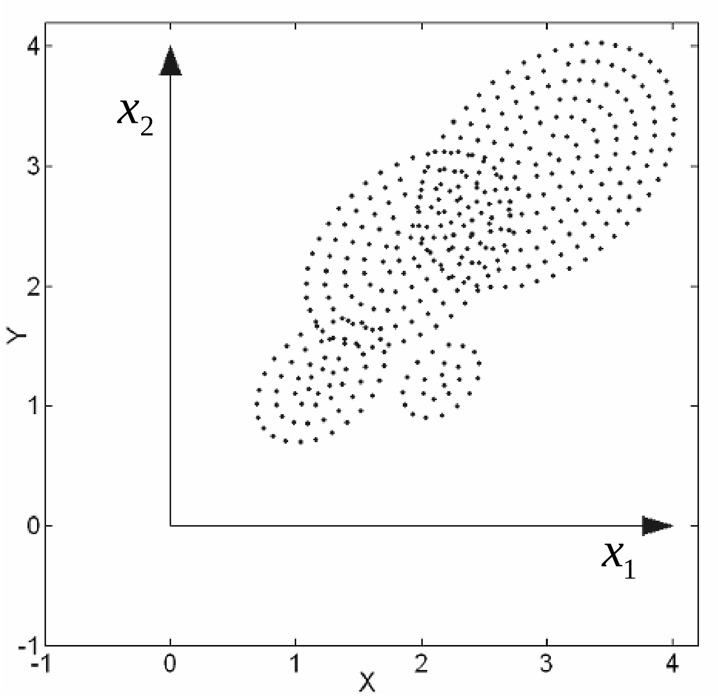
Input space partition using an extended FVS

Ordinary FVS with Gaussian kernel function has the same variance of Gaussian functions. On the contrary, the proposed extended FVS has the linear transformed input variables and the different variances of Gaussian kernel functions.

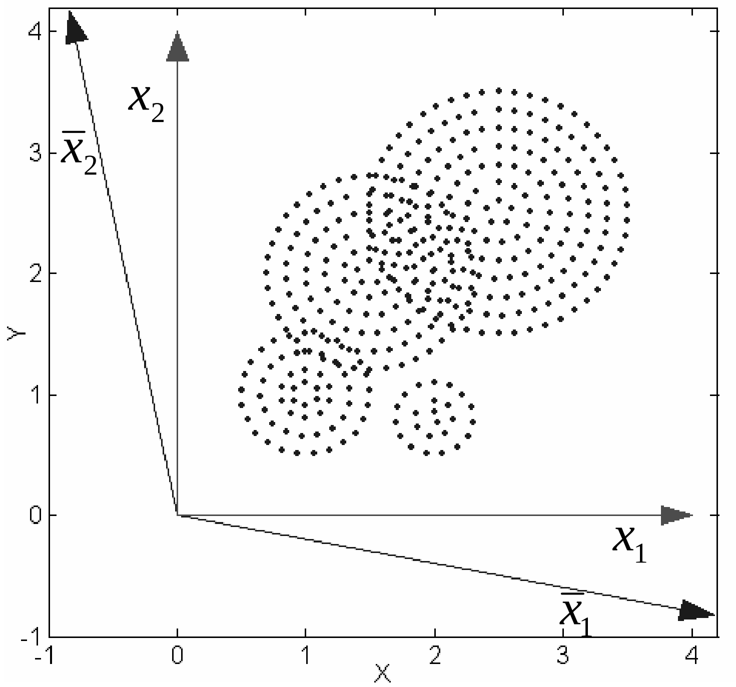
By the above linear transformation of the input variables with appropriate transformation matrix, input data can be relevantly represented. Moreover, the flexible variances of Gaussian kernel functions can help effective input space partitioning. These properties imply that the appropriate linear transformation and the flexible variances of Gaussian kernel functions can reduce the number of fuzzy rules and modeling error.

Figure 4.5 shows the input space partitioning methods of two-dimensional input space using the ordinary FVS and the extended FVS. Figure 4.5 (a) describes the input space partitioning of FVS with the same variances of Gaussian functions. Original input space is partitioned by 6 subspaces using the FVS with same variances. Figure 4.5 (b) illustrates the input space partitioning of the extended FVS with linear transformation and the different variances of Gaussian functions. Linear transformed input space is partitioned by 4 subspaces using the extended FVS. In Fig. 4.5 (b), minimum and maximum variances of Gaussian kernel functions are 0.5 and 1. In Fig. 4.5 (a), the same variance of Gaussian kernel functions is 0.75 as the mean of variance in Fig. 4.5 (b).

From the results of input space partition, Figures 4.5 (a) and (b) generate the six and four fuzzy rules, respectively. Figure 4.5 (b) with four rules shows that the number of fuzzy rules can be reduced as determining the appropriate linear transformation matrix and Gaussian variances using the GDM.

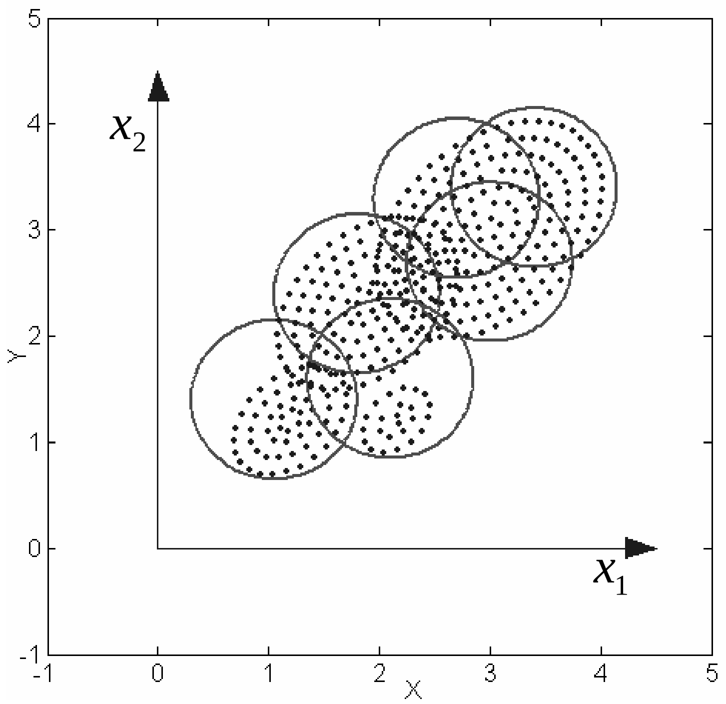


(a) Input variables

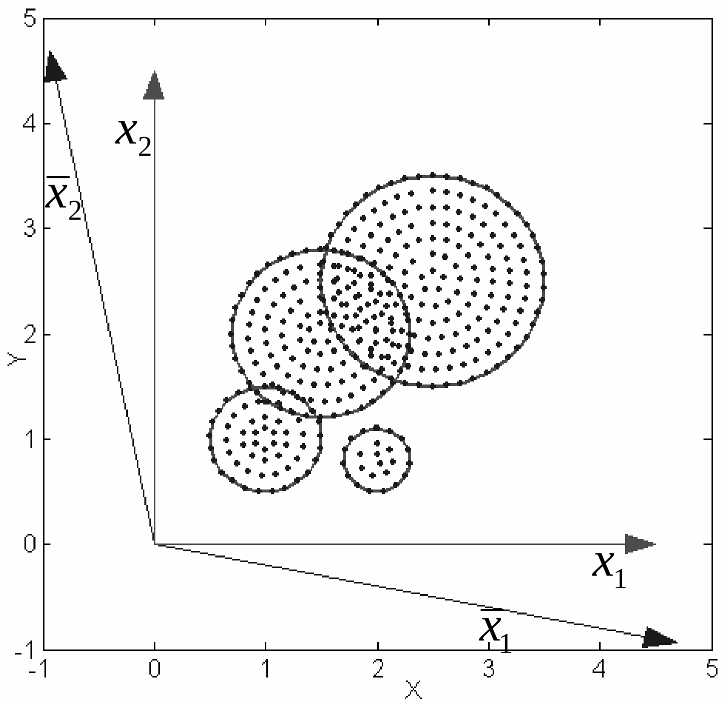


(b) The transformed input variables

Fig. 4.4 The linear transformation of 2-D input variables



(a) The input space partitioning of the FVS method



(b) The input space partitioning of the proposed FIS

Fig. 4.5 The input space partitions of the FVS and the proposed FIS

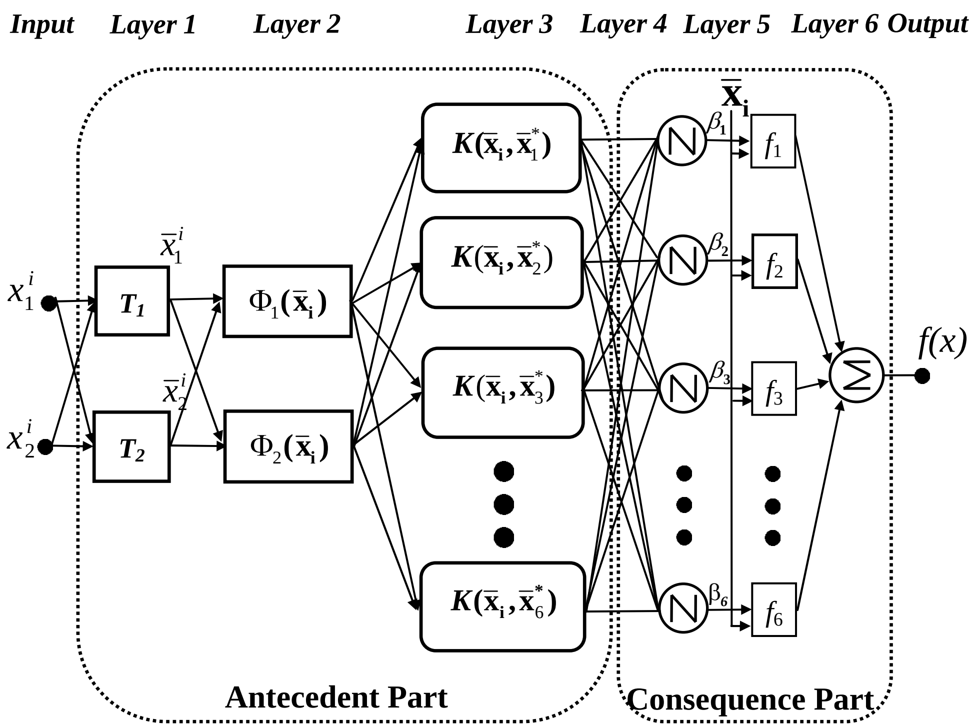


Fig. 4.6 The structure of the FIS for the modeling of $F_1(x_1, x_2)$

4.4 Examples

In this section, we show two simulation results of the proposed FIS for the modeling of typical nonlinear systems.

4.4.1 Example 1 : modeling of 2-input nonlinear function 1

The first example was taken from Wong's works [81]. The nonlinear function is presented as follows:

$$F_1(x_1, x_2) = \sin(\pi x_1)\sin(\pi x_2). \tag{4.29}$$

From the distributed gird points of input range $[-1, 1] \times [0, 1]$ within input space of nonlinear function $F_1(x_1, x_2)$, training data pairs of the $21 \times 11 = 231$ were obtained.

The proposed FIS generates the 6 FVs, so that it has 6 fuzzy rules as follows,

$$\begin{aligned} R_i \quad : \quad & \text{If } \bar{x}_1 \text{ is } K(\bar{x}_1, \bar{x}^*_{i1}) \text{ and } \bar{x}_2 \text{ is } K(\bar{x}_2, \bar{x}^*_{i2}), \\ & \text{Then} f_i = a_{i0} + a_{i1}\bar{x}_1 + a_{i2}\bar{x}_2,\ i = 1, ..., 6. \end{aligned} \tag{4.30}$$

Table 4.1 The parameter values of the FIS for modeling of $F_1(x_1, x_2)$

Rule	Antecedent part		Consequent part
	c_i	σ_i	($a_{i0},\ a_{i1},\ a_{i2}$)
1	(-0.1012, 0.6037)	0.7642	-61, 282, 6
2	(0.9884, 0.0011)	0.9283	1392, -398, -104
3	(-0.9884,-0.0011)	0.8723	-417, -84, 114
4	(0.9846, 1.0074)	0.7399	-73, 71, 15
5	(-0.9923, 1.0052)	0.6658	88, 71, -18
6	(0.0988, 0.0001)	0.9531	-525, -1059, 40

The structure of the FIS with 6 rules is shown in Fig. 4.6. For given the fitness of $\max_S J_S = 0.92$ and the initial condition of $\sigma_i = 0.75$, the linear transformation matrix T, the center c_i and the variance σ_i of the Gaussian membership function in antecedent part and coefficients a_{ij} in consequent part were obtained through learning procedure. The linear transformation matrix was computed as follows:

$$T = \begin{bmatrix} 0.9997 & -0.0001 \\ 0 & 1.0002 \end{bmatrix}. \tag{4.31}$$

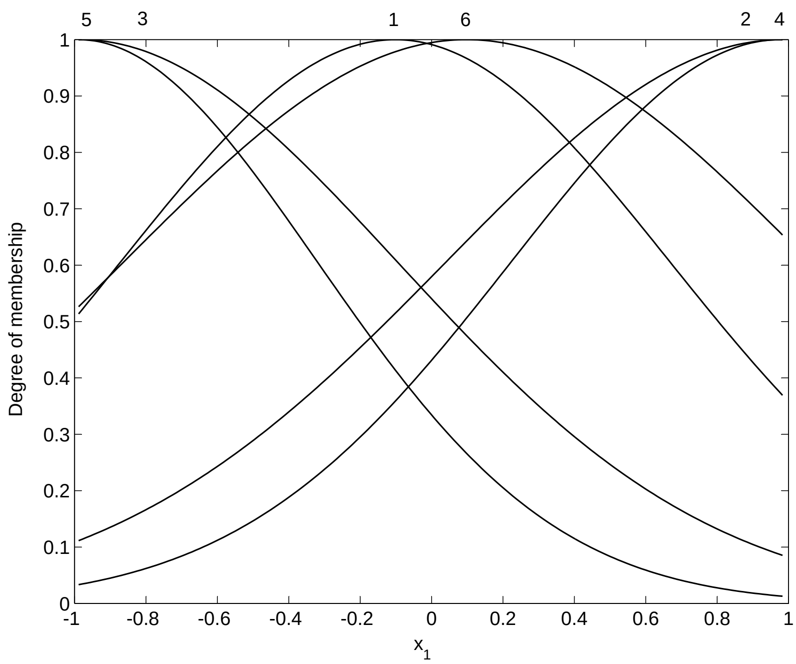
The parameter values of antecedent and consequent parts are listed in Table 4.1. Figure 4.7 shows the membership functions of proposed FIS with 6 rules for modeling of $F_1(x_1, x_2)$. Figure 4.8 shows the modeling result of $F_1(x_1, x_2)$ using an extended FVS.

To analyze the performance of the proposed FIS, the modeling error is defined by as following Root Mean Square Error (RMSE)

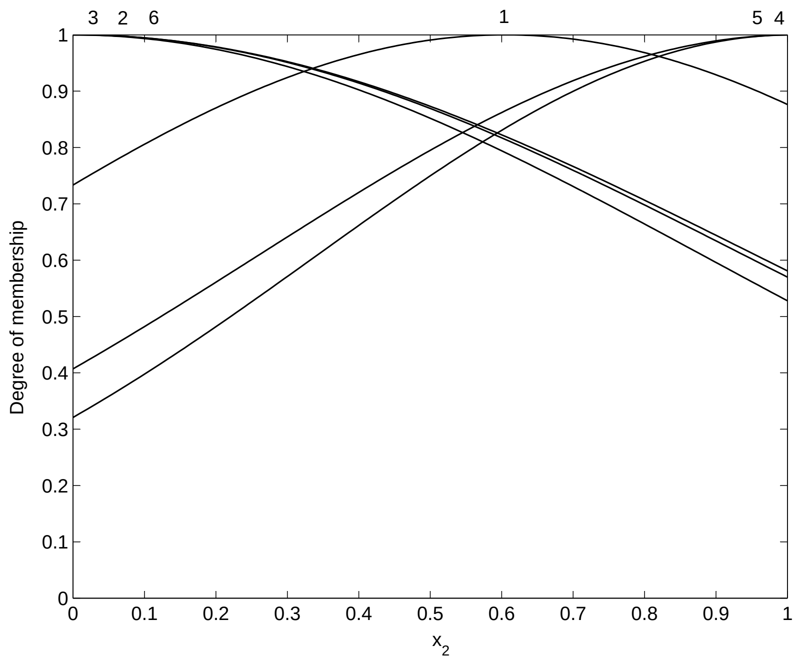
$$E = \sqrt{\frac{\sum_{k=1}^{N}(y_k - f(x_k))^2}{N}}, \tag{4.32}$$

where N is the number of data, y_k and $f(x_k)$ are the system and the model output, respectively.

The method in the literature applied to the same function $F_1(x_1, x_2)$, and the results are listed on the Table 4.2. Compared with the number of rules and modeling error of others, the proposed method gives the smallest modeling error .



(a) The membership functions of x_1



(b) The membership functions of x_2

Fig. 4.7 The membership functions of the proposed FIS for modeling of $F_1(x_1, x_2)$

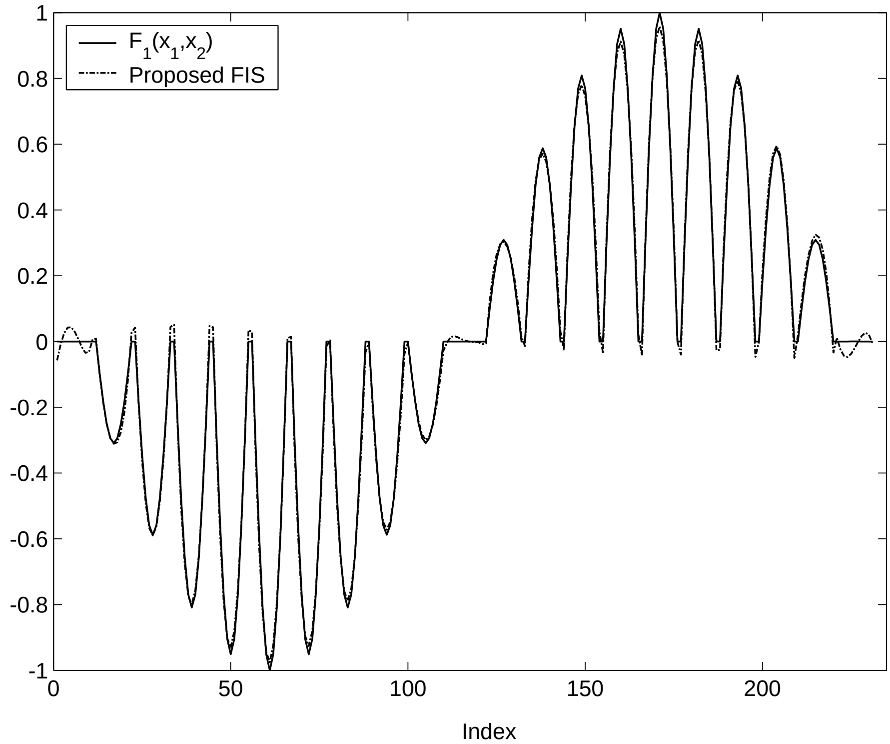


Fig. 4.8 The modeling result of $F_1(x_1, x_2)$

Table 4.2 The compared results of nonlinear function $F_1(x_1, x_2)$

Type	Rules(or FVs)	RMSE
Chan et al. [28]	8	0.2556
Baudat et al. [26]	6	0.3339
Kim et al. [7]	6	0.0676
Proposed FIS	6	0.0228

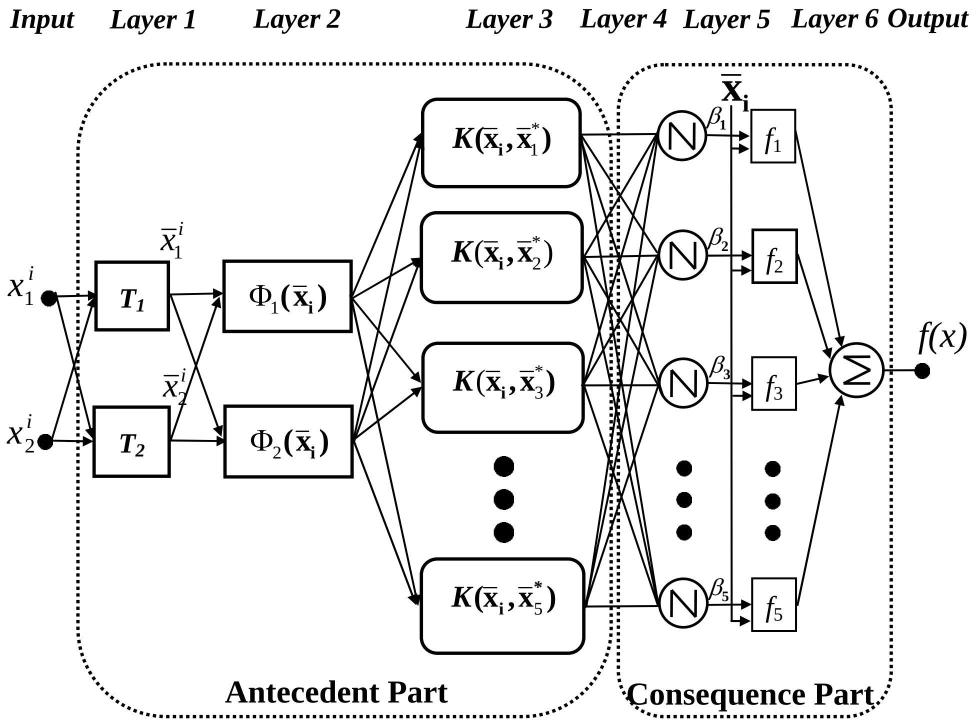


Fig. 4.9 The structure of the FIS for the modeling of $F_2(x_1, x_2)$

4.4.2 Example 2 : modeling of 2-input nonlinear function 2

Consider the nonlinear function [16]

$$F_2(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1.5})^2. \tag{4.33}$$

From input ranges $[1, 5] \times [1, 5]$ of (4.33), 50 training data pairs were obtained. The proposed FIS extracts the 5 FVs, so that it has 5 fuzzy rules as follows:

$$\begin{aligned} R_i \quad : \quad & \text{If } \bar{x}_1 \text{ is } K(\bar{x}_1, \bar{x}^*_{i1}) \text{ and } \bar{x}_2 \text{ is } K(\bar{x}_2, \bar{x}^*_{i2}), \\ & \text{Then} f_i = a_{i0} + a_{i1}\bar{x}_1 + a_{i2}\bar{x}_2, \ i = 1, ..., 5. \end{aligned} \tag{4.34}$$

The structure of the FIS is shown in Fig. 4.9.

For given the fitness of $\max_S J_S = 0.992$ and an initial condition of $\sigma_i = 3.2$, from learning algorithm, the T, c_i and σ_i in antecedent part and a_{ij} in consequent part were obtained. The linear transformation matrix was calculated as follows:

$$T = \begin{bmatrix} 0.9998 & 0.0001 \\ 0.0001 & 1.0002 \end{bmatrix}. \tag{4.35}$$

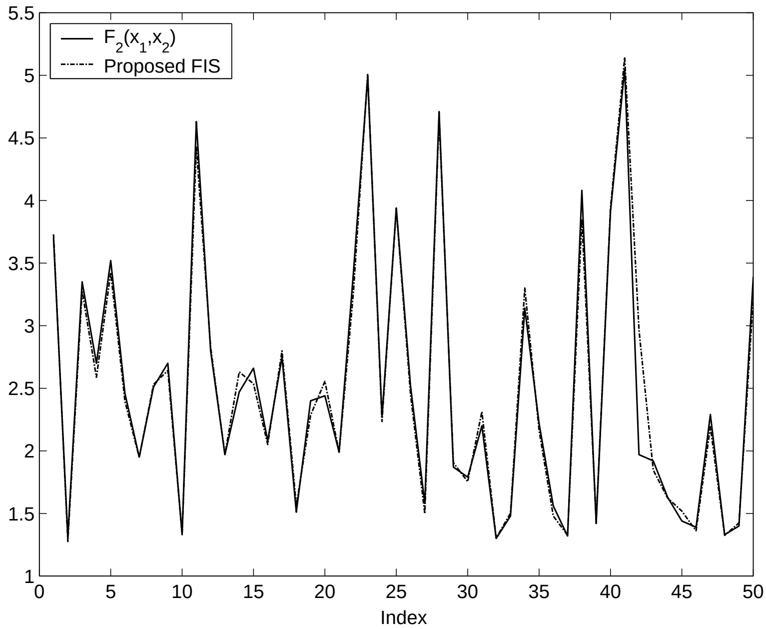


Fig. 4.10 The modeling result of $F_2(x_1, x_2)$

The parameter values of antecedent and consequent parts are listed in Table 4.3. Figure 4.10 shows the modeling result of $F_2(x_1, x_2)$ using the proposed FIS.

The method in the literature applied to the same function $F_2(x_1, x_2)$, and the results are listed on the Table 4.4. It shows that the proposed method gives the smallest modeling error with the smaller number of rules than others.

Table 4.3 The parameter values of the FIS for modeling of $F_2(x_1, x_2)$

Rule	Antecedent part		Consequent part
	c_i	σ_i	(a_{i0}, a_{i1}, a_{i2})
1	(2.4151, 2.4151)	3.3854	270946, -18283, -10464
2	(4.7757, 5.0076)	3.1297	9966, -55, -373
3	(1.2561, 4.5498)	3.0863	-8620, 623, 484
4	(4.3525, 1.5288)	3.1524	-1059, 1072, 1133
5	(1.2275, 1.5110)	3.3572	-195313, -7695, -5205

Table 4.4 The compared results of nonlinear function $F_2(x_1, x_2)$

Type	Rules(or FVs)	RMSE
Sugeno and Yasukawa [16]	6	0.281
Gomez-Skarmeta et al. [79]	5	0.266
Chan et al. [28]	6	0.324
Baudat et al. [26]	6	0.333
Kim et al. [7]	5	0.171
Proposed FIS	5	0.164

4.5 Discussion and Conclusions

In this chapter, we have presented a new approach to fuzzy modeling using an extended FVS. Our main concern is to determine the best structure of the TS fuzzy model for modeling nonlinear systems with measured input and output data. The number of rules and the parameter values of membership functions in the proposed FIS can be decided using an extended FVS based on kernel method. The kernel method involves the linear transform of input variables and kernel mapping. The linear transformation of input variables was proposed to solve problem selecting the best shape of the Gaussian kernel function corresponding to the nonlinear mapping. The linear transformation matrix and parameter values of kernel functions were adjusted using the gradient descent method. The coefficients of the TS fuzzy model in consequent part were determined by the least square estimation method. Examples showed the effectiveness of the proposed FIS for the modeling of nonlinear systems.

CHAPTER 5

Fuzzy Inference System Using an Extended RVM

This chapter presents a new fuzzy inference system for modeling of nonlinear dynamic systems based on input and output data with measurement noise. The proposed fuzzy system has a number of fuzzy rules and parameter values of membership functions which are automatically generated using the extended relevance vector machine (RVM). The RVM has a probabilistic Bayesian learning framework and has good generalization capability. The RVM consists of the sum of product of weight and kernel function which projects input space into high dimensional feature space. The structure of proposed fuzzy system is same as that of the Takagi-Sugeno fuzzy model. However, in the proposed method, the number of fuzzy rules can be reduced under the process of optimizing a marginal likelihood by adjusting parameter values of kernel functions using the gradient ascent method. After a fuzzy system is determined, coefficients in consequent part are found by the least square method.

5.1 Introduction

The Fuzzy Inference System (FIS) is very effective for modeling of nonlinear systems [10] [16]. However, the FIS based on only human expertise may not lead to sufficient accuracy for complex and uncertain systems. Therefore, neuro-fuzzy modeling which acquires knowledge from a set of input-output data has been actively investigated [1] [4] [5]. If training data set for modeling has measurement noise and (or) available data size is too small in the real system modeling, neural network can bring out over-fitting problem which is a factor of poor generalization. It is an important problem to select the appropriate structure of neuro-fuzzy model that can perform good generalization. Currently, some researchers have dealt with this problem. Branco *et al.* [82] investigated how and why fuzzy modeling systems are affected when learning data is corrupted by noise. Holmstrom *et al.* [83] made an effort to improve the generalization capability of a neural network by introducing additive noise to the training samples. Karystinos *et al.* [84] addressed K-mean clustering algorithm which results from the least entropic Gaussian mixture upon equal-likelihood cross-validated shaping for improving miltilayer perceptrons (MLP) generalization ability. Lee *et al.* [85] described a general regression neural network with fuzzy ART clustering (GRNNFA), as hybrid neural network model, based on the fusion of fuzzy adaptive resonance theory (Fuzzy ART) and the general regression neural network (GRNN) for data regression. However, many researches have usually dealt system optimization [84] [85] and generalization problem [83] independently.

Recently, statistical approach methods have been popularly developed in nonlinear system modeling based on input and output data with measurement noise [28] [86] [87] [88]. Statistical techniques generally deal with trade-off between fitting the training data and simplifying model capacity. In statistical method, kernel function offers an alternative solution by mapping the data into high dimensional feature space to increase the computational power [24] [8]. Particularly, the state-of-the-art Support Vector Machine (SVM)[21] has been used in order to find the number of network nodes or fuzzy rules based on given error bound [28] [29] [30] [89]. The Support Vector Neural Network (SVNN) is proposed to select the best structure of radial based function network for the given precision [28]. Support vector learning mechanism for fuzzy rule-based inference system is presented in [29] [30].

The SVM has delivered good performance in various application. However, the SVM has a number of the significant and practical limitations [27]. In the SVM, predictions are not probabilistic and the kernel function $K(\mathrm{x}, \mathrm{x}_i)$ must satisfy Mercer's condition. That is, it must be a positive definite continuous symmetric function. It is also necessary to estimate the error/margin trade-off parameter *C*. The number of the found support vector is sensitive to given error bound ϵ. Tipping [27] proposed the Relevance Vector Machine (RVM) based on a kernel-based Bayesian estimation method which does not suffer from above disadvantages. Above all, the RVM has shown a comparable generalization performance with fewer kernel function than the SVM in [27].

In this chapter, we propose a new fuzzy inference system, which performs system optimization and generalization simultaneously using relevance vector learning mechanism, for modeling nonlinear dynamic system based on input and output data with measurement noise. In the suggested fuzzy system, the number of fuzzy rules and parameter values of membership functions are automatically found using a relevance vector learning methodology. The structure of proposed fuzzy system is same as that of the Takagi-Sugeno (TS) fuzzy model. However, in the proposed method, the number of fuzzy rules can be reduced under the process of optimizing a marginal likelihood by adjusting parameter values of kernel functions using the gradient ascent method. After a fuzzy system is determined, coefficients in consequent part are found by the least square method.

5.2 Relevance Vector Machine (RVM)

The RVM has an exploited probabilistic Bayesian learning framework [90] [91]. It acquires relevance vectors and weights by maximizing a marginal likelihood. The structure of the RVM is described by the sum of product of weights and kernel functions. A kernel function means a set of basis function projecting the input data into a high dimensional feature space.

Given a data set of input-target pairs $\{\mathbf{x}_n, t_n\}_{n=1}^N$, and assuming that the targets are independent and contaminated with mean-zero Gaussian noise ϵ_n with variance σ^2:

$$t_n = y(\mathbf{x}_n; \mathbf{w}) + \epsilon_n. \tag{5.1}$$

The RVM without a bias term can be represented as follows [27] [92]:

$$\begin{aligned} y(\mathbf{x};\mathbf{w}) &= \sum_{i=1}^{N} w_i K(\mathbf{x},\mathbf{x}_i), && (5.2)\\ &= \Phi\mathbf{w}, && (5.3) \end{aligned}$$

where N is the length of the data, weight vector $\mathbf{w} = [w_1, ..., w_N]^T$ and $(N \times N)$ design matrix $\Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}), ..., \phi(\mathbf{x}_N)]^T$, wherein $\phi(\mathbf{x}_n) = [K(\mathbf{x}_n,\mathbf{x}_1), K(\mathbf{x}_n,\mathbf{x}_2), ..., K(\mathbf{x}_n,\mathbf{x}_N)]^T$ and $K(\mathbf{x},\mathbf{x}_i)$ is a kernel function.

The likelihood of the measured training data set is written as:

$$p(\mathbf{t}|\mathbf{w},\sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{t}-\Phi\mathbf{w}\|^2\right\}, \quad (5.4)$$

where target vector $\mathbf{t} = [t_1, ..., t_N]^T$. Maximizing likelihood estimation of $\mathbf{w}$ and σ^2 from (5.4) leads to over-fitting. To avoid this over-fitting, a zero-mean Gaussian prior distribution over $\mathbf{w}$ with variance $\boldsymbol{\alpha}^{-1}$ is added as:

$$\begin{aligned} p(\mathbf{w}|\boldsymbol{\alpha}) &= \prod_{i=0}^{N} \mathcal{N}(w_i|0,\alpha_i^{-1}),\\ &= \prod_{i=0}^{N} \sqrt{\frac{\alpha_i}{2\pi}}\ \exp\left(-\frac{\alpha_i}{2}\ w_i^2\right), && (5.5) \end{aligned}$$

where hyperparameter $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_N]^T$. An individual hyperparameter associates independently with every weight.

The posterior distribution over the weight from Bayes rule is thus given by:

$$\begin{aligned} p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha},\sigma^2) &= \frac{\textit{Likelihood} \times \textit{Prior}}{\textit{Normalizing factor}},\\ &= \frac{p(\mathbf{t}|\mathbf{w},\sigma^2)\ p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2)},\\ &= (2\pi)^{-(N+1)/2}|\boldsymbol{\Sigma}|^{-1/2}\ .\\ &\quad \exp\left\{-\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{w}-\boldsymbol{\mu})\right\}, && (5.6) \end{aligned}$$

where the posterior mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ are as follows:

$$\begin{aligned} \boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Sigma}\Phi^{\mathrm{T}}\mathbf{t}, && (5.7)\\ \boldsymbol{\Sigma} &= (\sigma^{-2}\Phi^{\mathrm{T}}\Phi +\ \mathbf{A})^{-1}, && (5.8) \end{aligned}$$

with $\mathbf{A} = \text{diag}(\alpha_1, \alpha_2, ..., \alpha_N)$.

The likelihood distribution over the training targets (5.4) can be marginalized with respect to the weights to obtain the *marginal likelihood*, which is also a Gaussian distribution

$$\begin{aligned} p(\mathbf{t}|\boldsymbol{\alpha}, \sigma^2) &= \int p(\mathbf{t}|\mathbf{w}, \sigma^2) p(\mathbf{w}|\boldsymbol{\alpha}) d\mathbf{w}, \\ &= (2\pi)^{-N/2} |\mathbf{C}|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{t}^T \mathbf{C}^{-1} \mathbf{t}\right\} \end{aligned} \tag{5.9}$$

with covariance $\mathbf{C} = \sigma^2 \mathbf{I} + \Phi\, \mathbf{A}^{-1} \Phi^{\mathrm{T}}$.

Values of $\boldsymbol{\alpha}$ and σ^2 that maximize the *marginal likelihood* can not be obtained in closed form, and an iterative re-estimation method is required [27]. The following approach of MacKay [93] gives:

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2}, \tag{5.10}$$

$$(\sigma^2)^{new} = \frac{\|\mathbf{t} - \boldsymbol{\Sigma}\boldsymbol{\mu}\|^2}{N - \sum_i \gamma_i}, \tag{5.11}$$

where μ_i is the i-th posterior mean weight (5.7) and the quantities $\gamma_i \equiv 1 - \alpha_i \sum_{ii}$ with the i-th diagonal element $\sum_{ii}$ of the posterior weight covariance (5.8).

In practice, since many of the hyperparameter α_i tend to infinity during the iterative re-estimation, the posterior distribution (5.6) of the corresponding weight w_i becomes highly peak at zero [27]. In this optimization process, the vector from the training set that associates with the remaining nonzero weights w_i is called the *relevance vector* (RV). The brief summary of inference procedure of the RVM is described as follows:

1. Initialize α_i and σ^2.
2. Compute $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and posterior distribution (5.6).
3. Re-estimate α_i and σ^2 using (5.7) and (5.8).
4. Go to step 2 until the maximum of α_i and variation of α_i are satisfied with given condition.
5. Find RVs and complete the structure of the RVM.

The main motivation of approach to fuzzy inference system is that the structure of the RVM is automatically found based on optimizing the marginal likelihood. The found structure of the RVM has close relation to that of fuzzy rule-base.

5.3 New Fuzzy Inference System Using an Extended RVM

This section describes the structure of the new fuzzy inference system based on the TS fuzzy model, input space partition method and the learning algorithm.

5.3.1 The structure of the FIS using an extended RVM

Let us suppose that we have given input and target data

$$(\mathbf{x}_1, t_1),\ (\mathbf{x}_2, t_2),\ ...,\ (\mathbf{x}_N, t_N) \tag{5.12}$$

where $\mathbf{x}_i$=$[x_1^i, x_2^i, ..., x_D^i](i = 1, 2, ..., N)$ is a input variable and **t**=$[t_1, ..., t_N]$ is a target variable. The proposed TS fuzzy model with fuzzy if-then rules can be represented as follows:

$$\begin{aligned}
R_1 \quad &: \quad \text{If } x_1 \text{ is } K(x_1, x_{11}^*) \text{ and } \cdots \text{ and } x_D \text{ is } K(x_D, x_{1D}^*), \\
&\quad\ \ \text{Then } \ f_1 = a_{10} + a_{11}x_1 + \cdots + a_{1D}x_D \\
R_2 \quad &: \quad \text{If } x_1 \text{ is } K(x_1, x_{21}^*) \text{ and } \cdots \text{ and} x_D \text{ is } K(x_D, x_{2D}^*), \\
&\quad\ \ \text{Then } \ f_2 = a_{20} + a_{21}x_1 + \cdots + a_{2D}x_D \\
&\qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
R_n \quad &: \quad \text{If } x_1 \text{ is } K(x_1, x_{n1}^*) \text{ and } \cdots \text{ and } x_D \text{ is } K(x_D, x_{nD}^*), \\
&\quad\ \ \text{Then } \ f_n = a_{n0} + a_{n1}x_1 + \cdots + a_{nD}x_D,
\end{aligned} \tag{5.13}$$

where n is the number of fuzzy rules, D is the dimension of input variables, $x_j(j = 1, 2, ..., D)$ is an input variable, f_i is the i-th local output variable, $K(x_j, x_{ij}^*)(i = 1, 2, ..., n, j = 1, 2, ..., D)$ is a fuzzy set and $a_{ij}(i = 1, 2, ..., n, j = 0, 1, ..., D)$ is a consequent parameter.

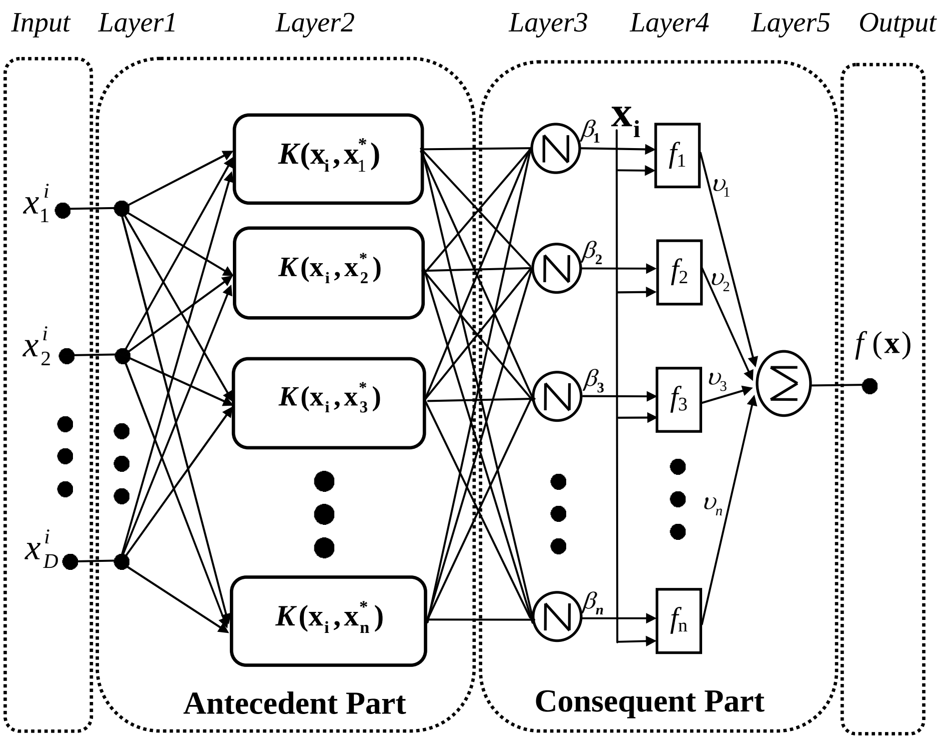


Fig. 5.1 The structure of the proposed fuzzy inference system

Now, we describe the structure of FIS using the extended RVM. It consists of five layers as shown in Fig. 5.1. The five layers involved in the proposed FIS are presented as follows:

Layer 1: Each input variable transmits one node. Input variables are distributed to next layer.

Layer 2: The distributed input space is nonlinearly projected into feature space using kernel functions. Each kernel function corresponds to one fuzzy linguistic label, that is, fuzzy set (example, young, middle, old, etc). Since kernel function is not necessary to satisfy Mercer's condition, various types of it can be used, such as polynomial, Gaussian, Fourier series, triangular, bell, trapezoidal ones etc.. Because Gaussian kernel function allows the exact computation of the center and variance of predictive distribution and variance can be easy learned, in this paper, it is employed as follows:

$$K(x_j, x^*_{ij}) = \exp\Big(- \frac{(x_j - x^*_{ij})^2}{2\theta^2_{ij}} \Big), \tag{5.14}$$

where x^*_{ij} is the RV, θ_{ij} is called a kernel parameter and n is the number of RVs and i=1,...,n, and j=1,...,D. After all, this kernel function becomes a Gaussian membership function in the proposed FIS. $K(x_j, x^*_{ij})$ is the grade of membership of x_j. x^*_{ij} and θ_{ij} are respectively the center and variance of the Gaussian membership function of j-th dimension term of i-th input variable x_i. The Relevance vector learning algorithm plays a role as a fuzzy inference engine finding the number of fuzzy rules in FIS. The Layer 1 and 2 are related to the antecedent part of the FIS.

Layer 3: The fuzzy intersection of Gaussian kernel functions is calculated. Here, the following algebraic product operator as T-norm operator for each Layer 3 node is used,

$$K(\mathbf{x}, \mathbf{x}^*_i) = \prod_{j}^{D} K(x_j, x^*_{ij}), \tag{5.15}$$

where $\mathbf{x} = [x_1, x_2, ..., x_D]$ is the i-th input variable vector, $\mathbf{x}^*_i = [x^*_{i1}, x^*_{i2}, ..., x^*_{iD}]$ is the RV of the i-th input variable.

The normalized weight β_i for each fuzzy rule (node) is computed as follows,

$$\beta_i = \frac{K(\mathbf{x}, \mathbf{x}^*_i)}{\sum_{j=1}^{n} K(\mathbf{x}, \mathbf{x}^*_j)}, \tag{5.16}$$

where

$$K(\mathbf{x}, \mathbf{x}^*_i) \geq 0,\ \sum_{j=1}^{n} K(\mathbf{x}, \mathbf{x}^*_j) > 0,\ i = 1, ..., n. \tag{5.17}$$

Layer 4: The normalized weight β_i of each node is multiplied by i-th local output variable f_i. Each node output υ_i as shown in Fig. 5.1 is presented as follows:

$$\upsilon_i = \beta_i f_i, \tag{5.18}$$

$$= \frac{K(\mathbf{x}, \mathbf{x}^*_i)(a_{i0} + a_{i1}x_1 + \cdots + a_{iD}x_D)}{\sum_{j=1}^{n} K(\mathbf{x}, \mathbf{x}^*_j)}, \tag{5.19}$$

where $f_i = a_{i0} + a_{i1}x_1 + \cdots + a_{iD}x_D$ is the i-th local output variable of TS fuzzy model.

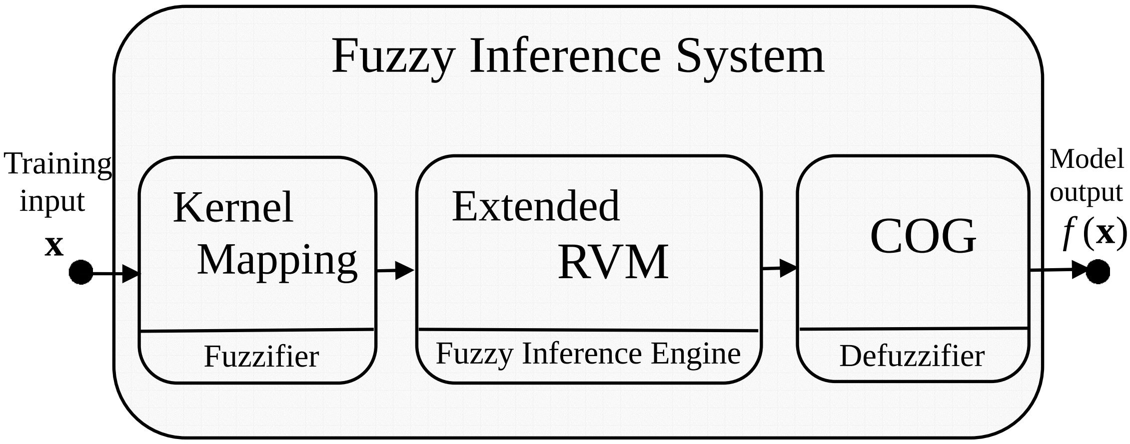


Fig. 5.2 The learning algorithm of the proposed FIS

Layer 5: For the overall output of the fuzzy model constructed, defuzzification using the Center Of Gravity (COG) method is performed. Each node corresponds to one output variable $f(\mathbf{x})$,

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{n} \beta_i f_i, \\ &= \frac{\sum_{i=1}^{n} K(\mathbf{x}, \mathbf{x}_i^*)(a_{i0} + a_{i1}x_1 + \cdots + a_{iD}x_D)}{\sum_{j=1}^{n} K(\mathbf{x}, \mathbf{x}_j^*)}. \end{aligned} \tag{5.20}$$

The Layer 3, 4 and 5 connect with the consequent part of the proposed FIS.

5.3.2 The learning algorithm of the FIS using an extended RVM

The learning algorithm of the FIS using the RVM is shown in Fig. 5.2. It can be summarized by the following learning procedure.

Step 1: Assign the initial hyperparameter α, kernel parameter θ_{ij} and the learning rate η_θ.

Step 2: Using the following extended RVM algorithm based on kernel mapping [27], find RVs $\mathbf{x}_i^*$ being the centers $\mathbf{c}_i$ of Gaussian membership function and weight $\mathbf{w}$. Particularly, using the Gradient Ascent Method (GAM), kernel parameter θ_{ij} is adjusted in order to select the appropriate type of kernel function related to the nonlinear dynamic system. Assume that the log of the marginal likelihood (5.9) is the objective function L,

$$L = -\frac{1}{2}\left[\log|\sigma^2\mathbf{I} + \Phi\,\mathbf{A}^{-1}\Phi^{\mathrm{T}}| \; + \mathbf{t}^{\mathrm{T}}(\sigma^2\mathbf{I} + \Phi\,\mathbf{A}^{-1}\Phi^{\mathrm{T}})^{-1}\mathbf{t}\right]. \tag{5.21}$$

From the GAM, the kernel parameter θ_{ij} is updated such that the objective function L is maximized as:

$$\begin{aligned}
\Delta\theta_{ij} &= \eta_\theta \nabla_{\theta_{ij}} L, \\
&= \eta_\theta \frac{\partial L}{\partial \theta_{ij}}, \\
&= \eta_\theta \frac{\partial L}{\partial \phi_{mi}} \frac{\partial \phi_{mi}}{\partial \theta_{ij}}, \\
&= \eta_\theta \theta_{ij}^{-3}\left[\sum_{m=1}^{N}\sum_{i=1}^{n} F_{mi}\Phi_{mi}(x_{mj} - x_{ij})^2\right]
\end{aligned} \tag{5.22}$$

where $F_{mi} = \partial L/\partial\phi_{mi}$ wherein matrix $F = \sigma^{-2}[(\mathbf{t}-\mathbf{y})\boldsymbol{\mu}^{\mathrm{T}} - \Phi\boldsymbol{\Sigma}]$, a set of Gaussian kernel function $\phi_{mi} = \exp\{-\sum_{j=1}^{D}(x_{mi} - x_{ij})^2/2\theta_{ij}^2\}$ and η_θ is the learning rate of θ_{ij}.

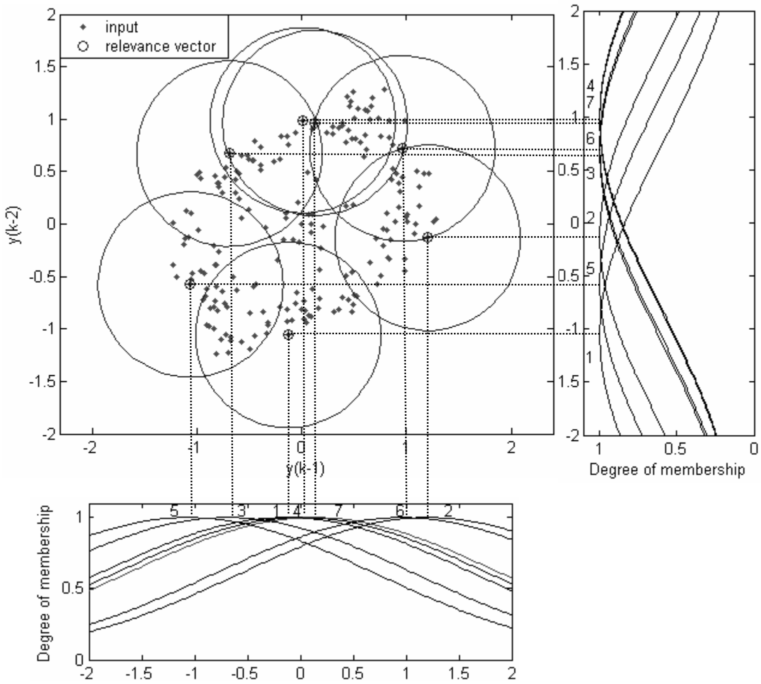
This learning Step 2 is inserted into the inference procedure Step 3 of the RVM in Section 5.2 Therefore, the extended RVM re-estimates θ_{ij} together with α_i and σ^2 in inference procedure Step 3 of the RVM.

Step 3: Using the following Least Square Estimation (LSE) method, estimate the parameter a_{ij} of the linear equation f_i in (5.20). Let

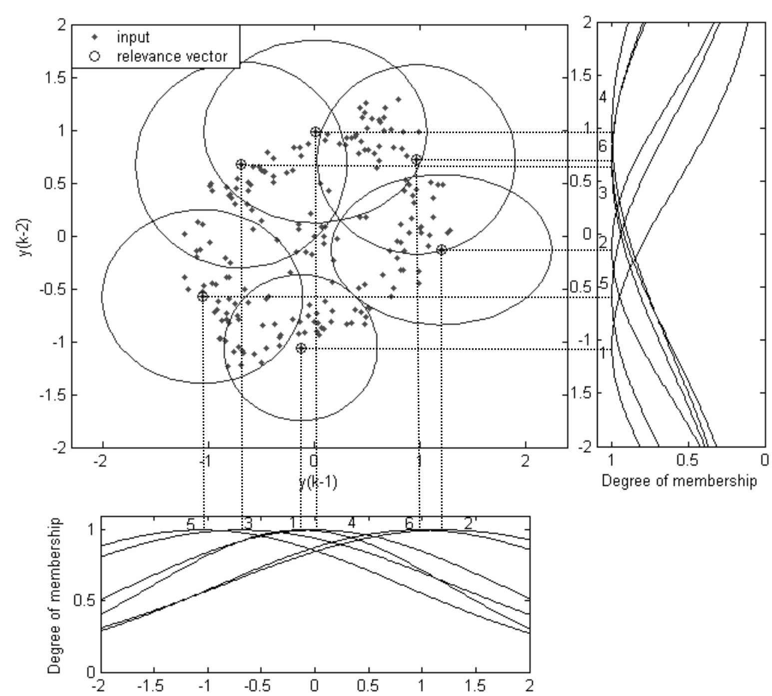
$$\begin{aligned}
A &= \begin{bmatrix} a_{10} & a_{11} & \cdots & a_{1D} & \cdots\cdots & a_{n0} & a_{n1} & \cdots & a_{nD} \end{bmatrix}^T, \\
W &= \begin{bmatrix}
\beta_1^1 & \beta_1^1 x_1^1 & \cdots & \beta_1^1 x_D^1 & \cdots\cdots & \beta_n^1 & \beta_n^1 x_1^1 & \cdots & \beta_n^1 x_D^1 \\
\beta_1^2 & \beta_1^2 x_1^2 & \cdots & \beta_1^2 x_D^2 & \cdots\cdots & \beta_n^2 & \beta_n^2 x_1^2 & \cdots & \beta_n^2 x_D^2 \\
\vdots & \vdots & \cdots & \vdots & \cdots\cdots & \vdots & \vdots & \cdots & \vdots \\
\beta_1^l & \beta_1^l x_1^l & \cdots & \beta_1^l x_D^l & \cdots\cdots & \beta_n^l & \beta_n^l x_1^l & \cdots & \beta_n^l x_D^l
\end{bmatrix},
\end{aligned} \tag{5.23}$$

where $\beta_i^j = \frac{K(\mathbf{x}_j,\mathbf{x}_i^*)}{\sum_{k=1}^{n} K(\mathbf{x},\mathbf{x}_k^*)}$. Thus fuzzy model output is $f(\mathbf{x}) = WA$.
The parameter vector A is calculated using the following pseudo inverse,

$$A = (W^{\mathrm{T}}W)^{-1}W^{\mathrm{T}}\mathbf{y}. \tag{5.24}$$



(a) RVM



(b) The extended RVM

Fig. 5.3 The input space partition of the proposed FIS using the RVM(a) and the extended RVM(b)

5.3.3 The input space partition of the FIS using an extended RVM

The structure of fuzzy modeling is closely related to the partitioning of input space for fuzzy rule generation. The input space partition approach of the proposed FIS is a clustering-based method. Figure 5.3 shows the input space partition method of two-dimensional input space. Figure 5.3 (a) and (b) show input space partitioning using the RVM and the extended RVM, respectively. Because each cluster leads to a fuzzy rule, seven and six rules are respectively created in Fig. 5.3 (a) and (b). The RV as center of Gaussian kernel function becomes the center of Gaussian membership function.

Although the RV is sparse because the posterior distributions of many of the weights are sharply peaked around zero in RVM. Figure 5.3 illustrates how the method using the extended RVM can reduce the number of rules and membership function. The 4-th and 7-th rules which are generated using the RVM with the fixed Gaussian variance in Fig. 5.3 (a) can be merged into the 4-th rule using the extended RVM with a different Gaussian variance θ_{ij} in Fig. 5.3 (b).

The proposed FIS through the generalization strategy of the RVM estimates the noise of system and determines fuzzy rules and parameters of membership functions automatically.

5.4 Examples

In this section, two simulation results of the proposed FIS for the modeling of the nonlinear dynamic systems are described.

5.4.1 Example 1 : modeling of 2-input nonlinear dynamic system

Consider the nonlinear dynamic system [28],

$$\begin{aligned} y(k) &= (0.8 - 0.5\ \exp(-y^2(k-1)))y(k-1) \\ &\quad - (0.3 + 0.9\ \exp(-y^2(k-1)))y(k-2) \\ &\quad + 0.1\sin(\pi y(k-1)) + e(k) \qquad (5.25) \end{aligned}$$

where $e(k)$ is a white noise, $e(k) \sim N(0, 0.1^2)$. The training input of the model is $X(k) = [y(k-1), y(k-2)]$. For $e(k) \equiv 0$, this nonlinear dynamic system is unstable

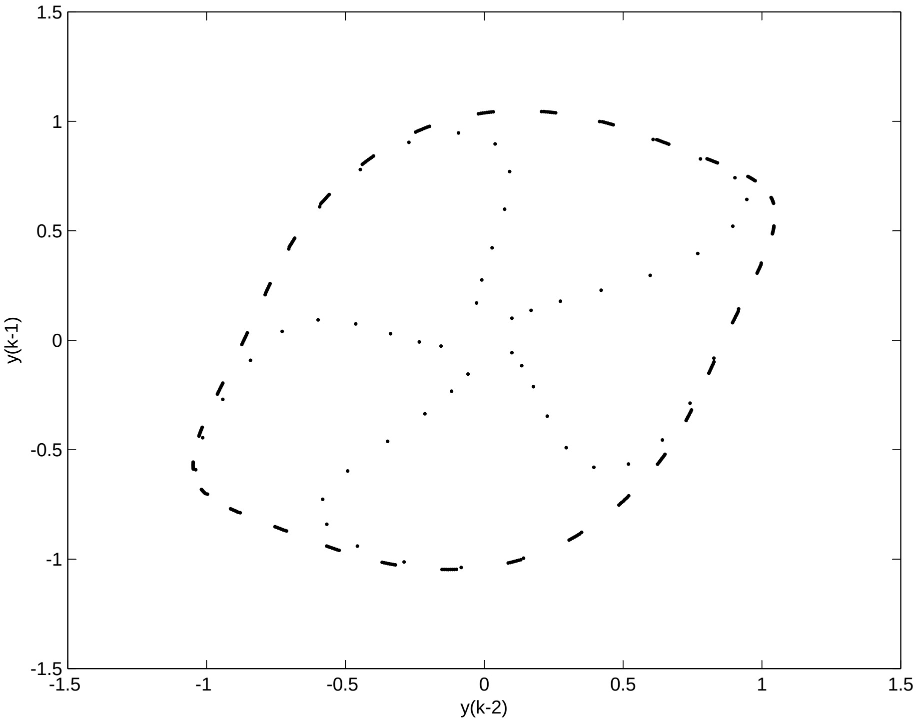
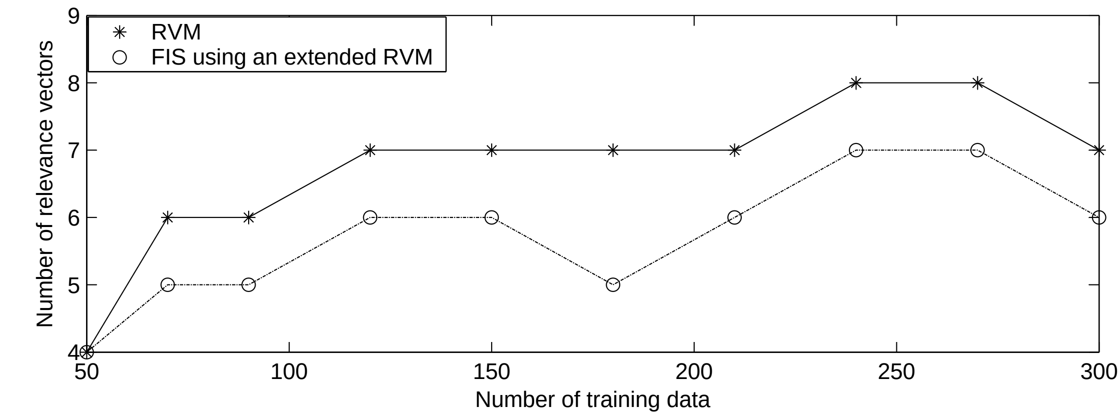


Fig. 5.4 The output data of dynamic system for $e(k) \equiv 0$

at the origin. Output data of dynamic system with 300 data points for $e(k) \equiv 0$ is shown in Fig. 5.4. This data points are generated from an initial condition of $X(1) = [0.1, 0.1]$. But the training input data of 300 point pairs are generated from initial condition of $X(1) = [0, 0]$. The proposed FIS using the extended RVM has the following fuzzy If-Then rules.

$$\begin{aligned} R_i : \text{If } & y(k-1) \text{ is } K(y(k-1), y^*_{i1}(k-1)) \\ & \text{and } y(k-2) \text{ is } K(y(k-2), y^*_{i2}(k-2)), \\ \text{Then } & f_i = a_{i0} + a_{i1}y(k-1) + a_{i2}y(k-2), i = 1, ..., n. \end{aligned} \tag{5.26}$$

When training data sizes are generally large from $k = 1$, the number of RVs and the prediction test error of both algorithms, the RVM and the FIS using the extended RVM, are shown in Fig. 5.5. When training data size increases under the same initial $\theta = 1.2782$, prediction test error decreases. The number of RV in FIS using the extended RVM is smaller than that of the RVM for a similar error.



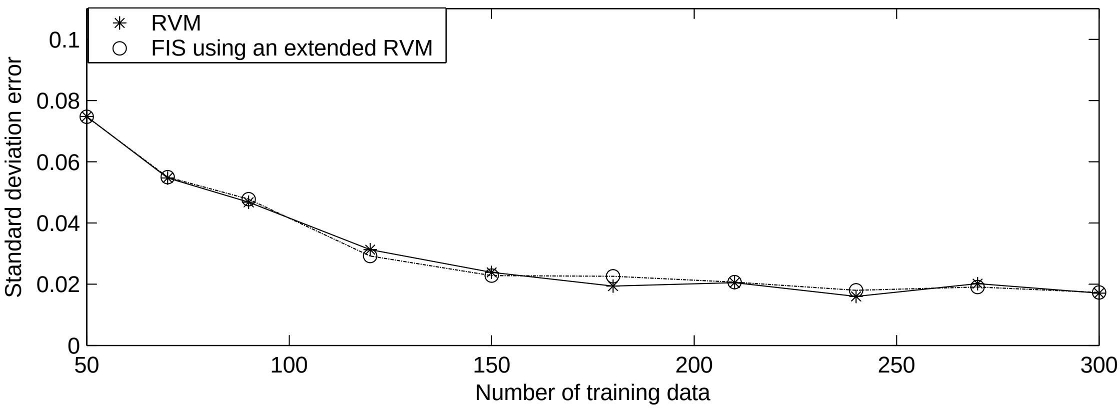


Fig. 5.5 The number of RVs and prediction test error of the RVM and the FIS using the extended RVM when data sizes are generally large

Table 5.1 The parameter values of the FIS for modeling of $X(k) = [y(k-1), y(k-2)]$

Rule	Antecedent part		Consequent part
	c_{ij}	θ_{ij}	$(a_{i0},\ a_{i1},\ a_{i2})$
1	(0.7821 -0.2076)	1.1791 1.1696	-107.50 -27.08 -31.64
2	(1.1036 0.5064)	1.2799 1.1488	362.62 -69.23 -39.57
3	(0.0464 -0.9427)	1.2344 1.2385	50.95 -16.81 12.20
4	(-1.0619 -0.5757)	1.2759 1.2319	-56.45 -16.93 0.35
5	(-0.5010 0.6232)	1.2491 1.2397	389.53 45.00 11.61
6	(-0.0553 0.9856)	1.2762 1.3008	-619.62 11.81 120.18

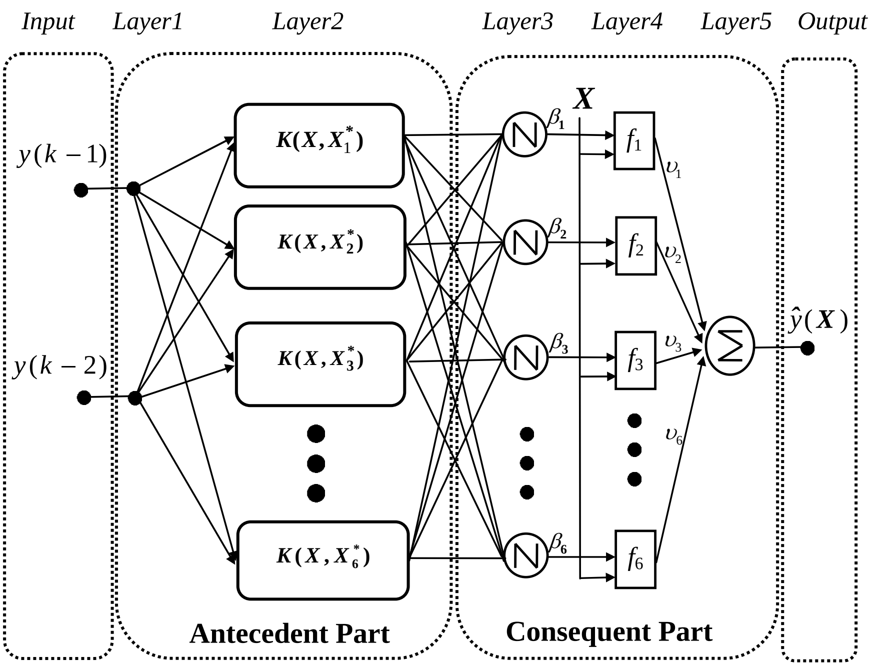
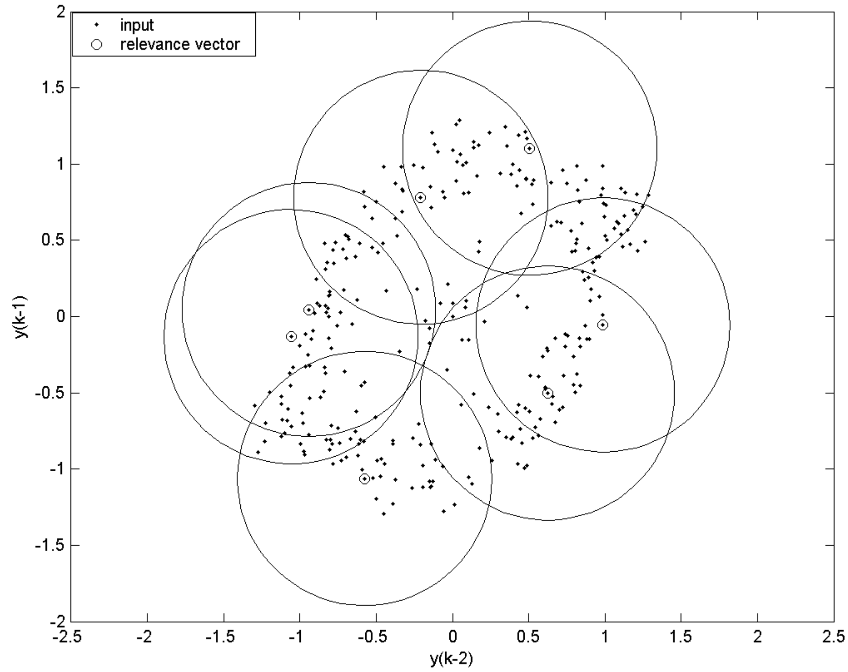
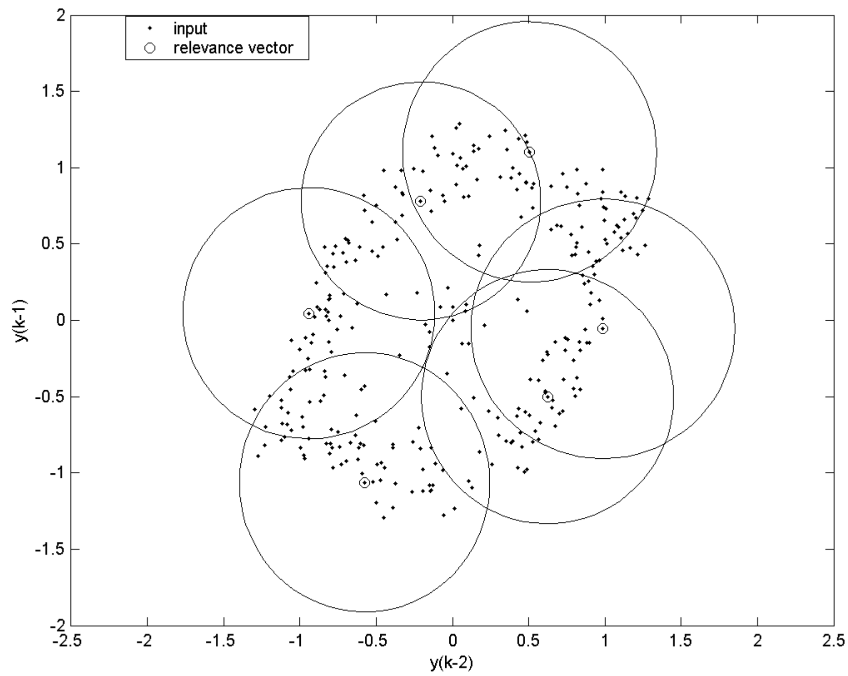


Fig. 5.6 The structure of the proposed FIS for modeling of $y(k)$

After the simulation from the training input data of 300 point pairs, the proposed FIS using the extended RVM generates 6 RVs ($\mathbf{x}_i^*$), so that it has 6 rules as follows. The structure of the proposed FIS using the extended RVM for modeling of $y(k)$ is shown in Fig. 5.6. Figure 5.7 shows input space partitioning using the RVM(a) and the extended RVM(b) in training data of dynamic system with noises and found RVs. The parameter values of antecedent and consequent parts are listed in Table 5.1. The c_{ij} and θ_{ij} are the center and variance of Gaussian membership function, respectively. Parameters $(a_{i0},\ a_{i1},\ a_{i2})$ are the consequent those of TS fuzzy model. Membership functions of FIS are shown in Fig. 5.8. Figure 5.9 shows the modeling result of estimated dynamic system output of $X(k) = [y(k-1), y(k-2)]$. Modeling output of estimated dynamic system as shown in Fig. 5.9 is similar to output of original system with no error as shown in Fig. 5.4. The method in the literature applied to the same dynamic system, and the results listed on the Table 5.2. The extended RVM is used as fuzzy inference engine in proposed FIS. The initial condition of simulation such as initial hyperparameter α and kernel parameter θ_i is

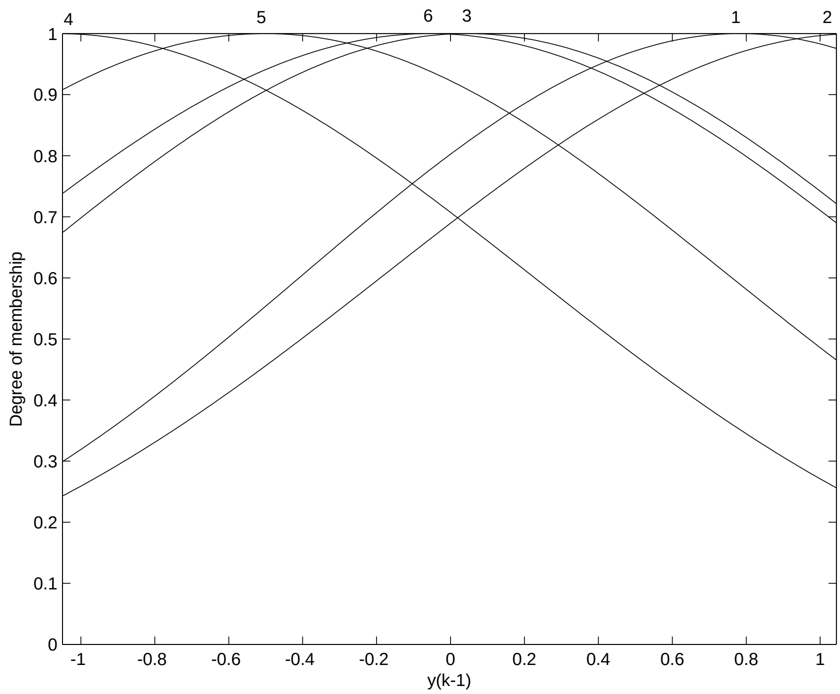


(a) RVM

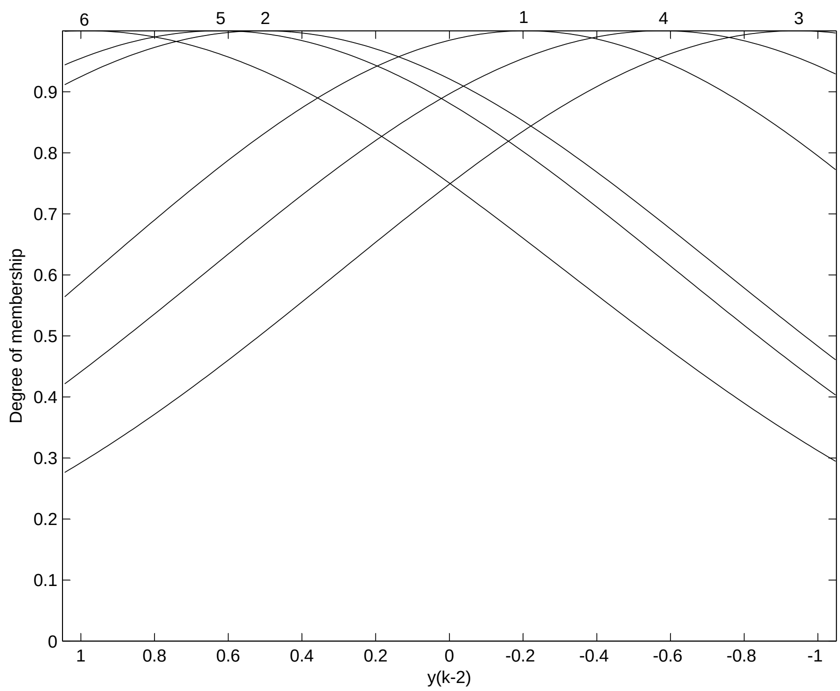


(b) The extended RVM

Fig. 5.7 The comparison of input space partitions using the RVM(a) and the extended RVM(b) in training data of dynamic system with noises and found RVs(∘)



(a) The membership function of $y(k-1)$



(b) The membership function of $y(k-2)$

Fig. 5.8 The membership functions $y(k-1)$ and $y(k-2)$ of the proposed FIS

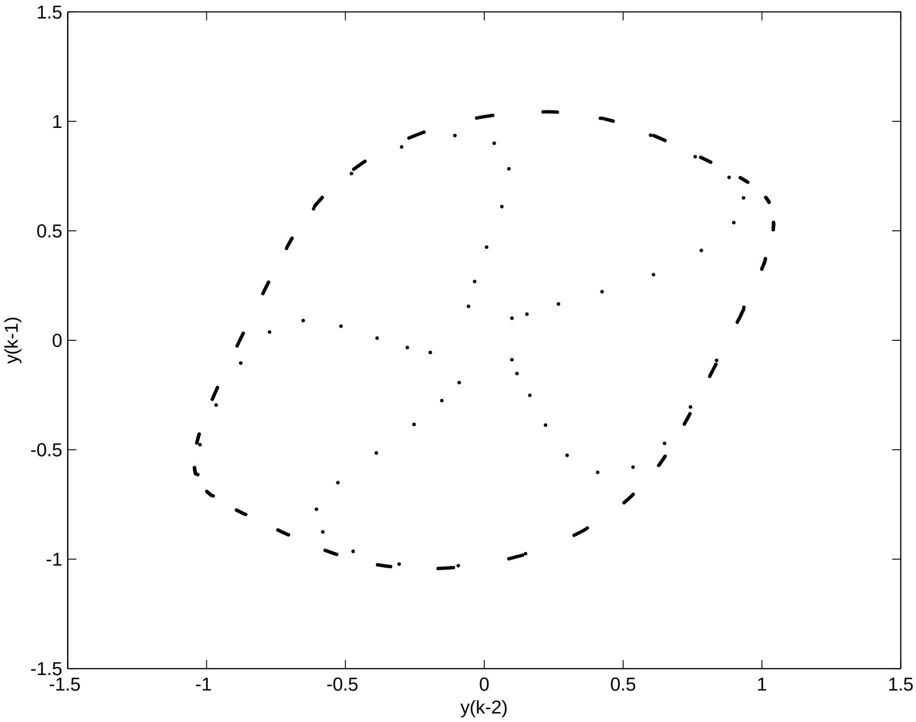


Fig. 5.9 The estimated dynamic system output of $X(k) = [y(k-1), y(k-2)]$

equal. The modeling error is the standard deviation of test errors. Compared with the number of rules and modeling error, the proposed method using the extended RVM gives the smaller number of rules and modeling error than the Chan's approach shown in Table 5.2. Especially, the FIS gives the smaller number of rules for the same error.

Table 5.2 The compared results of nonlinear dynamic function

Type	Rules(or SVs/RVs)	Model error
Chan *et al.* [28]	10	0.099
RVM	7	0.017
Proposed FIS using the extended RVM	6	0.017

5.4.2 Example 2 : modeling of robot arm data

The training robot arm data are obtained from the relationship between input variables (x_1, x_2) of joint angles and target variables (y_1, y_2) of positions,

$$y_1 = 2.0\cos x_1 + 1.3\cos(x_1 + x_2) + \delta, \quad (5.27)$$

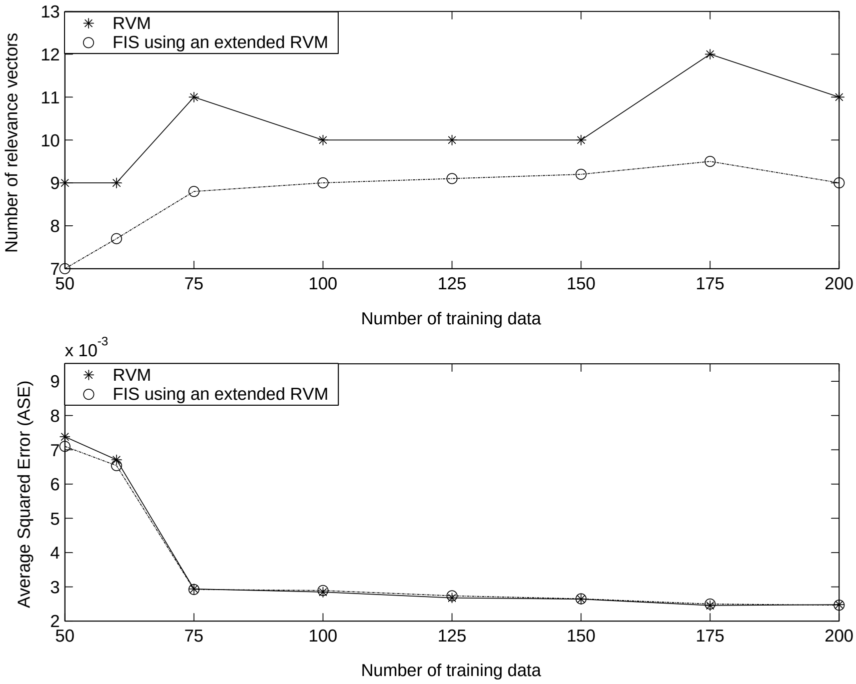
$$y_2 = 2.0\sin x_1 + 1.3\sin(x_1 + x_2) + \delta, \quad (5.28)$$

where δ is a Gaussian noise, $\delta \sim N(0, 0.05^2)$. The 400 input-target pairs of robot arm which was used by MacKay [94] and Chu *et al.* [89] are used. In this data set, the first 200 data and the second 200 data are used as training and test data set, respectively.

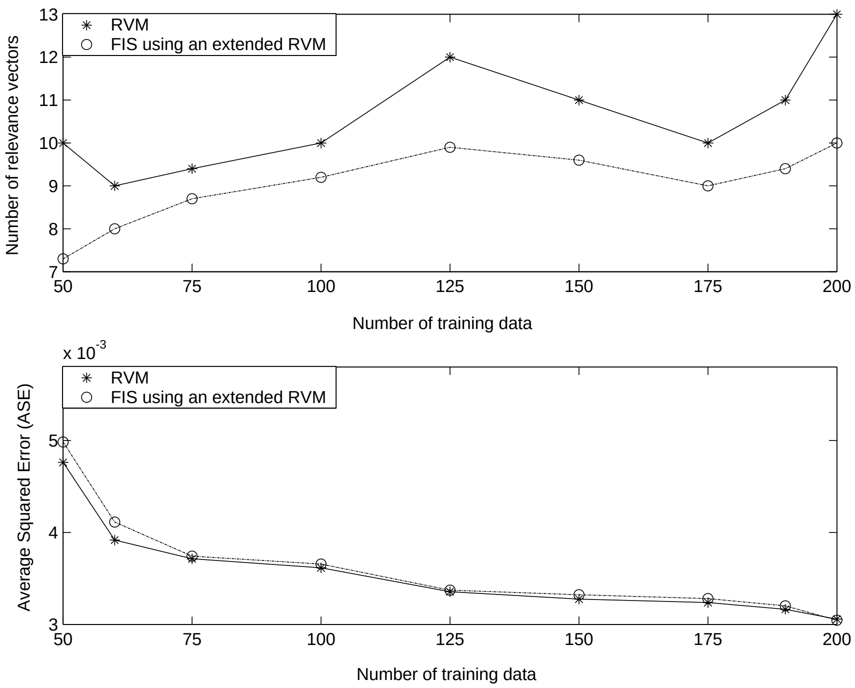
The proposed FIS using the extended RVM has the following fuzzy If-Then rules.

$$R_i \;:\; \text{If } x_1 \text{ is } K(x_1, x_{i1}^*) \text{ and } x_2 \text{ is } K(x_2, x_{i2}^*), \quad \text{Then } f_i = a_{i0} + a_{i1}x_1 + a_{i2}x_2.\ i = 1, ..., n. \quad (5.29)$$

When training data sizes are generally large in target variables (y_1, y_2), the number of RVs and prediction test error of the RVM and the FIS using the extended RVM are shown in Fig. 5.10. Average results for 10 repetitions were quoted, where 50, 60, 75, 100, 125, 150, 175 and 200 randomly generated training samples from training data. When training data sizes increase under the same initial condition $\theta = 1.7677$, prediction test error decreases. The number of RV of FIS using the extended RVM is smaller than that of the RVM for a similar error. After the simulation from the training input data of 200 point pairs, the proposed FIS using the extended RVM respectively generates 9 and 10 RVs $(\mathbf{x}_i^*)$ for y_1 and y_2, so that it has 9 and 10 rules. Figures 5.11 and 5.12 show input space partitioning of y_1 and y_2 using the RVM(a) and the extended RVM(b), respectively. Under the same initial condition such as hyperparameter α and kernel parameter θ_i, 11 RVs were merged into 9 RVs using the extend RVM in Fig. 5.11 and 13 RVs were merged into 10 RVs in Fig. 5.12. The parameter values of antecedent and consequent parts of proposed FIS are listed in Table 5.3 and 5.4. The membership functions of y_1 and y_2 are shown in Figs. 5.13 and 5.14. Figure 5.15 shows comparison of test robot arm data of y_1 and y_2 and outputs of the proposed FIS using the extended RVM.

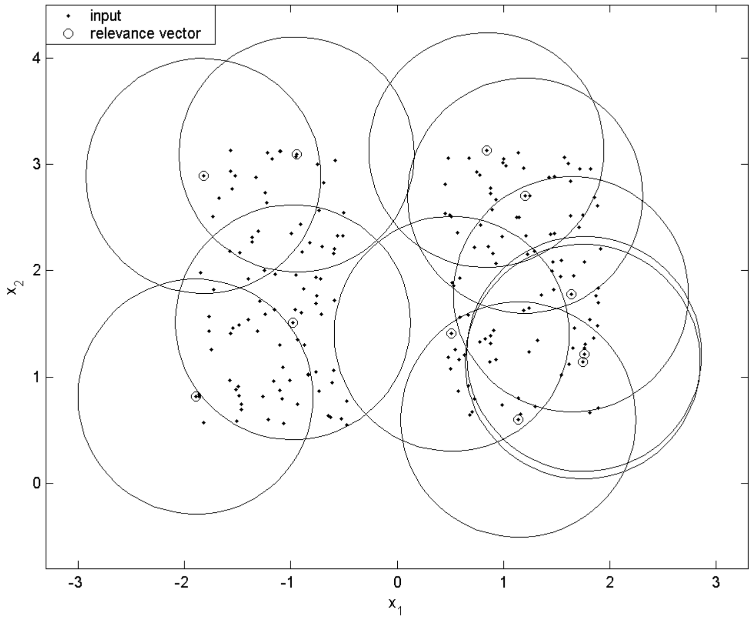


(a) When training data of y_1 is generally large

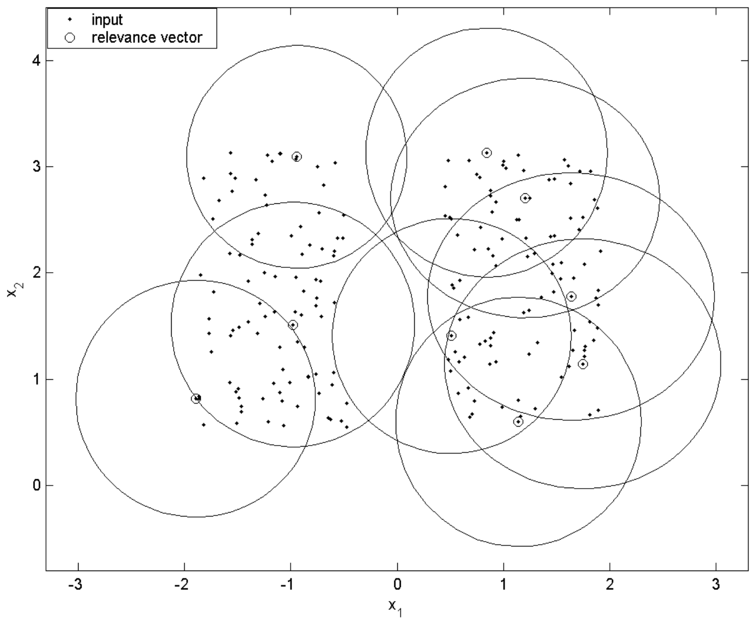


(b) When training data of y_2 is generally large

Fig. 5.10 The number of RVs and prediction test error of the RVM and the FIS using the extended RVM when training data of y_1 (a) and y_2 (b) are generally large

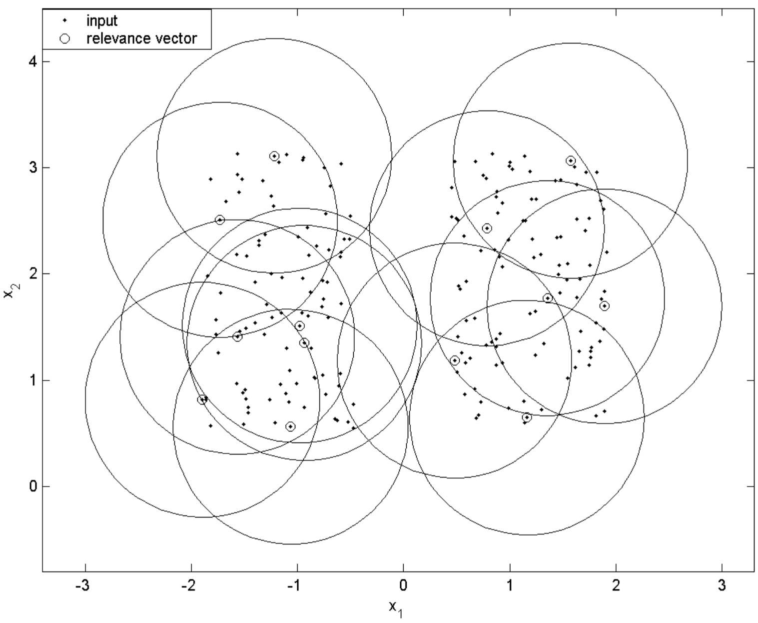


(a) RVM

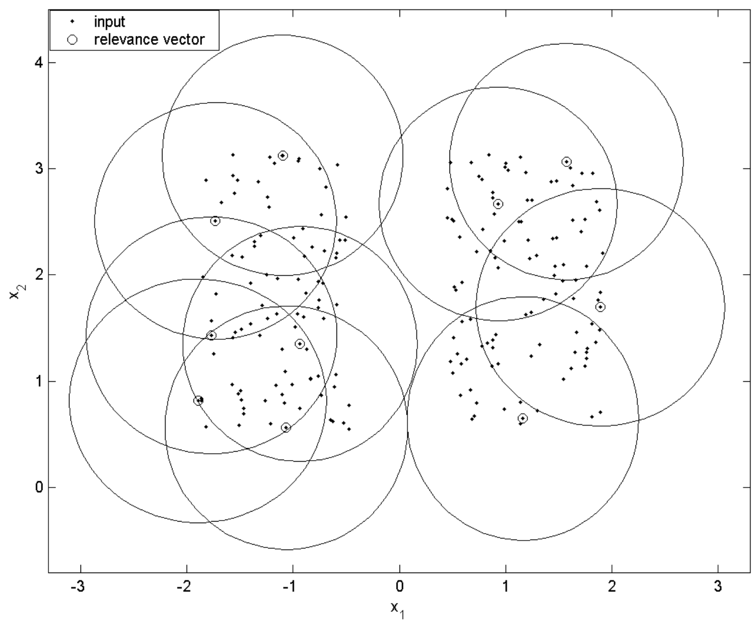


(b) The extended RVM

Fig. 5.11 The comparison of input space partitions using the RVM(a) and the extended RVM(b) in training data of y_1 with noises and found RVs($\circ$)



(a) RVM



(b) The extended RVM

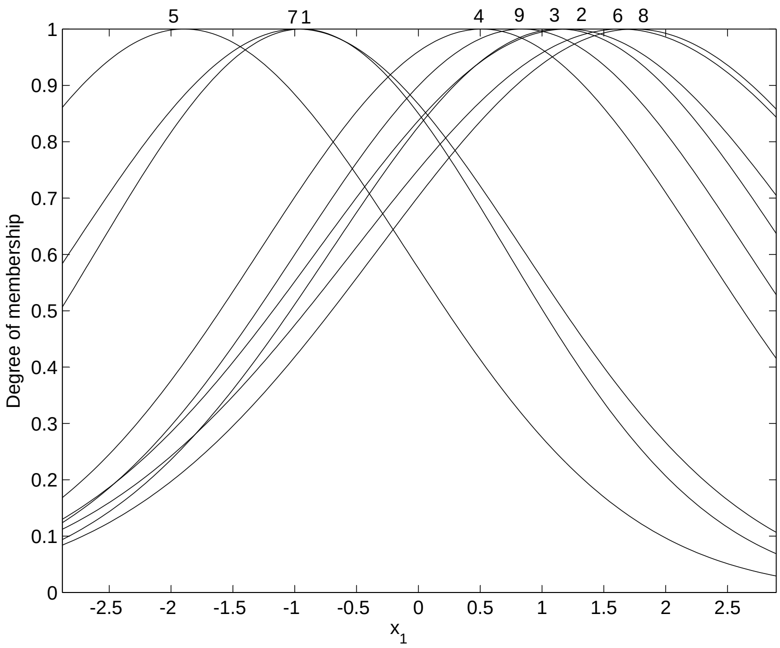
Fig. 5.12 The comparison of input space partitions using the RVM(a) and the extended RVM(b) in training data of y_2 with noises and found RVs(○)

Table 5.3 The parameter values of the FIS for modeling of y_1

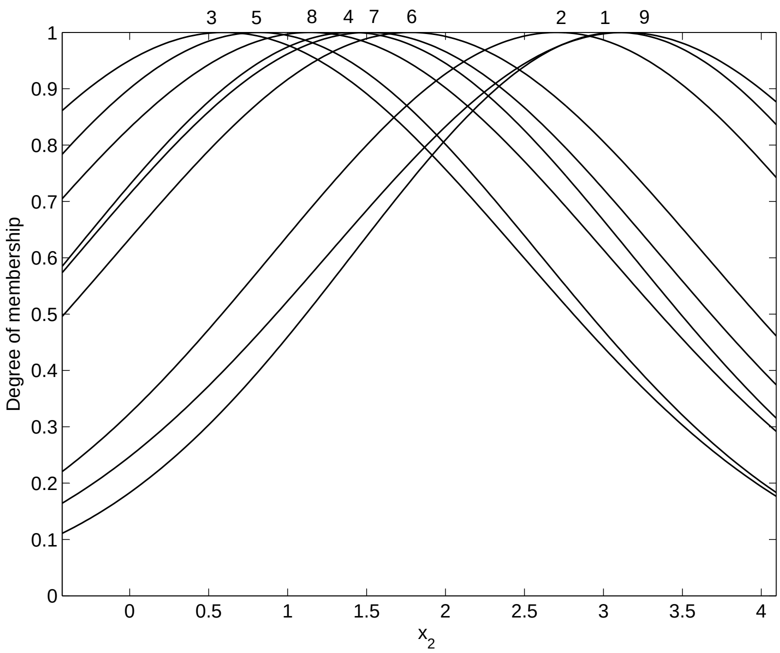
Rule	Antecedent part		Consequent part
	c_{ij}	θ_{ij}	$(a_{i0},\ a_{i1},\ a_{i2})(\times 10^4)$
1	(-0.9451 3.0913)	1.6588 1.6775	-0.0129 0.0009 0.0103
2	(1.2031 2.7042)	2.0207 1.8009	-1.2575 0.2623 0.0406
3	(1.1397 0.5988)	1.8473 1.8774	-0.0960 -0.0161 0.0596
4	(0.5122 1.4056)	1.7967 1.7697	-0.9088 -0.0897 0.0128
5	(-1.8941 0.8151)	1.8013 1.7803	0.3044 0.0316 -0.0024
6	(1.6345 1.7778)	2.1581 1.8618	7.0833 -0.1508 -0.5640
7	(-0.9796 1.5137)	1.8310 1.8410	-0.7585 0.0134 0.1401
8	(1.7438 1.1445)	2.0773 1.8796	-3.8342 -0.1712 -0.5538
9	(0.8388 3.1328)	1.8192 1.8736	1.0656 -0.0248 -0.0978

Table 5.4 The parameter values of the FIS for modeling of y_2

Rule	Antecedent part		Consequent part
	c_{ij}	θ_{ij}	$(a_{i0},\ a_{i1},\ a_{i2})(\times 10^4)$
1	(-0.9362 1.3506)	1.7684 1.7694	0.2482 0.0293 0.0044
2	(0.9283 2.6690)	1.7911 1.7589	0.1233 0.0649 0.0329
3	(1.1615 0.6477)	1.7439 1.8359	0.0179 -0.0015 0.0047
4	(-1.0987 3.1254)	1.8132 1.8088	0.2060 -0.0208 -0.0094
5	(1.8904 1.6948)	1.8714 1.7888	-0.0176 0.0027 -0.0017
6	(-1.8941 0.8151)	1.9331 1.8334	1.2448 0.0753 0.1497
7	(-1.7287 2.5087)	1.8173 1.7800	0.2522 -0.0354 -0.0871
8	(-1.0644 0.5615)	1.8271 1.8348	-0.5942 -0.0095 -0.0888
9	(1.5707 3.0659)	1.7596 1.7809	-0.4063 0.0153 0.0208
10	(-1.7653 1.4311)	1.8854 1.7824	-0.9783 0.0563 -0.0278

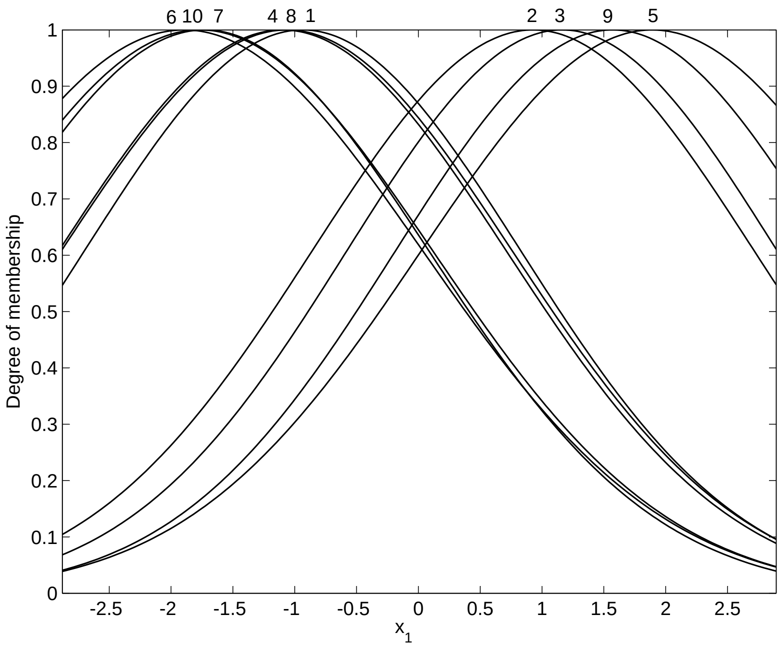


(a) The membership functions of x_1

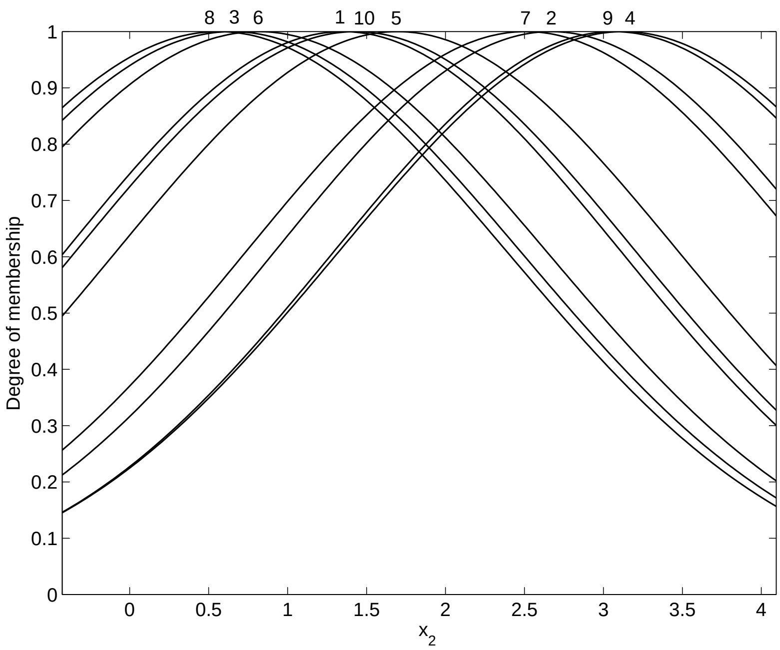


(b) The membership functions of x_2

Fig. 5.13 The membership functions of the proposed FIS for modeling of y_1

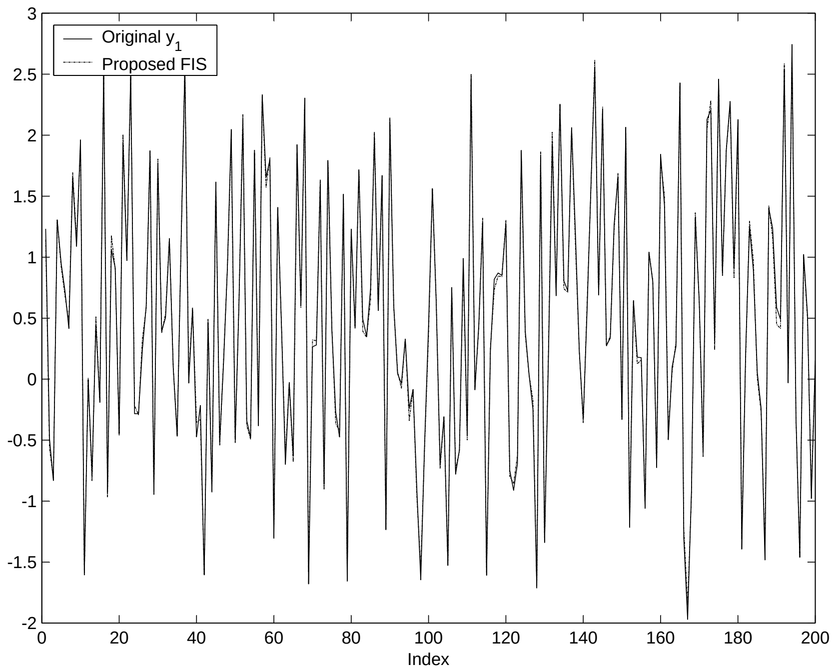


(a) The membership functions of x_1

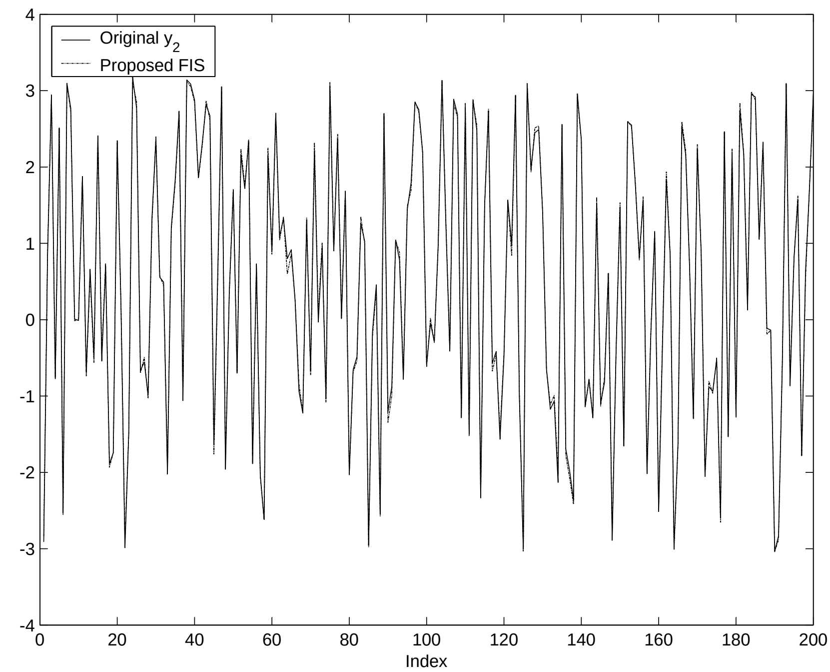


(b) The membership functions of x_2

Fig. 5.14 The membership functions of the proposed FIS for modeling of y_2



(a) robot arm data of y_1



(b) robot arm data of y_2

Fig. 5.15 The comparison of test robot arm data of y_1 (a) and y_2 (b) and outputs of the proposed FIS

Table 5.5 The compared results of the modeling robot arm data y_1 and y_2

Type		Rules(or SVs/RVs)	ASE ($\times 10^{-3}$)
Chu *et al.* [89]	y_1	21	2.491
	y_2	42	3.184
RVM	y_1	11	2.475
	y_2	13	3.057
Proposed FIS using the extended RVM	y_1	9	2.465
	y_2	10	3.046

To analyze the performance of the proposed FIS, the modeling error is defined as following Average Square Error (ASE)

$$\text{ASE} = \frac{\sum_{k=1}^{N}(y_k - f(x_k))^2}{N}, \tag{5.30}$$

where N is the number of data, y_k and $f(x_k)$ are the original system and fuzzy modeling output, respectively. The method in the literature applied to the same system and the results listed on the Table 5.5. A comparison in terms of the number of rules and modeling error shows that the proposed method using the extended RVM gives the smaller number of rules for a similar modeling error than approaches shown in Table 5.5.

5.5 Discussion and Conclusions

In this chapter, a new approach to fuzzy modeling using the relevance vector learning mechanism based on a kernel-based Bayesian estimation was proposed. Our main concern is to find the best structure of the TS fuzzy model for modeling nonlinear dynamic systems with measurement error. The number of fuzzy rules and the parameter values of membership functions can be found as optimizing the marginal likelihood of the RVM in the proposed FIS. Because the RVM is not necessary to satisfy Mercer's condition, kernel function is beyond the limit of the positive definite continuous symmetric function of SVM. The relaxed condition of kernel function can satisfy the various types of membership functions in fuzzy model.

We applied the proposed method to two nonlinear dynamic functions. The RVM compared with support vector learning mechanism in examples had the small model capacity and described good generalization. Simulated results showed the effectiveness of the proposed FIS for modeling of nonlinear dynamic systems with noise.

The RVM showed a good generalization property in examples of reference [27]. In the extended RVM, marginal likelihood (5.9) with respect to Gaussian kernel parameter θ is maximized using the gradient ascent method. The choice of learning parameter η_θ influences the convergence of the extended RVM. In this thesis, the η_θ was experimentally selected. Nevertheless, the FIS using the extended RVM has good generalization property in Examples.

In RVM [27], the posterior weight covariance matrix Σ of (5.8), which requires an inverse operation of order $O(M^3)$ complexity and $O(M^2)$ memory storage, with M the number of basis functions is computed in order to re-estimate hyperparameters α and σ. In addition, the gradient ascent method is added to update Gaussian kernel parameter θ. We need to improve computing time for big data size. The iteration of this algorithm depends on inference procedure of the RVM. When the maximum of α_i and variation α_i are satisfied with given condition, this algorithm is stopped.

CHAPTER 6

Conclusions

In this thesis, we present new approaches to fuzzy inference system for system modeling using kernel machines. Our main concern is to determine the best structure of the TS fuzzy model for modeling nonlinear system based on input and output data. The number of fuzzy rules and the parameter values of membership functions which are automatically generated using the extended Support Vector Machine (SVM), the extended Feature Vector Selection (FVS) and the extended Relevance Vector Machine (RVM) as a kernel machine.

In FIS using an extended SVM, the structure of the proposed FIS is obtained by minimizing a constrained quadratic programming problem for a given error bound in SVM. The number of fuzzy rules can be reduced by adjusting the parameter values of Gaussian kernel function using the gradient descent method.

In FIS using an extended FVS, the structure of the proposed FIS is obtained using an extended Kernel method. The learning algorithm of the extended FVS is faster than the extended SVM. The extended kernel method consists of linear transformation of input variables and kernel mapping of the extended FVS. The linear transformation of input variables is used to solve problem selecting the best shape of the Gaussian kernel function. The number of fuzzy rules can be reduced by adjusting the linear transformation matrix and parameter values of kernel functions using the gradient descent method.

In FIS using an extended RVM, the structure of the proposed FIS is obtained using the relevance vector learning mechanism based on a kernel-based Bayesian estimation. The RVM consists of the sum of product of weight and kernel function which projects input space into high dimensional feature space. The extended RVM generates the smaller number of fuzzy rules than the extended SVM. The extended RVM does not need the linear transformation of input variables because the basis function of the extended RVM is not restricted within the limitation of the kernel function. The number of fuzzy rules can be reduced by adjusting the parameter values of kernel functions using a gradient ascent method. After a fuzzy model is determined, coefficients in consequent part are determined using the least square estimation method.

In the experiment presented in each chapter, the performance and result of the proposed FIS were evaluated and discussed. The results of all simulations showed the effectiveness of the proposed FIS for modeling nonlinear systems.

As future work, we need to select the proper kernel function corresponding to nonlinear system and improve the computation capacity in learning process. In addition to, online learning mechanisms of the SVM, FVS and RVM are necessary for more effective modeling of the nonlinear system.

Bibliography

[1] J. R. Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference Systems", *IEEE Trans. Syst. Man. Cybern.*, vol.23, pp.665-685, 1993.

[2] C. T. Lin, "Neural Fuzzy Control Systems with Structuer and Parameter Learing", New York: World Scientific, 1994.

[3] C. T. Lin and C. S. G. Lee, "Neural Fuzzy Systems : A Neural-Fuzzy Synergism to Intelligent Systems", Englewood Cliffs, Prentice-Hall, 1995.

[4] C. F. Juang and C. T. Lin, "An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications", *IEEE Trans. Fuzzy Syst.*, vol.6, no.1, pp.12-32, 1999.

[5] J. Ryu and S. Won, "Partitioning of Linearly Transformed Input space in Adaptive Network Based Fuzzy Inference System", *IEICE Trans. Inf. and Syst.*, vol.E84-D, no.1, pp.213-216, 2001.

[6] S. Mitra and Y. Hayashi, "Neuro-Fuzzy Rule Generation: Survey in Soft Computing Framework", *IEEE Trans. on Neural Network*, vol.11, no.3, pp.748-768, 2000.

[7] J. Kim and S. Won, "New Fuzzy Inference System using a Support Vector Machine," *Proc. 41th IEEE conf. of decision and control,* Las Vegas, USA, pp1349-1354, 2002.

[8] J. Kim, T. Kim and Y. Suga, "A New Approach to Fuzzy Modeling Using an Extended Kernel Method", *IEICE Trans. Fundamentals*, vol.E86-A, no.9, pp.2262-2269, 2003.

[9] J. Kim Y. Suga, and S. Won, "A New Approach to Fuzzy Modeling of Nonlinear Dynamic Systems With NOise: Relevance Vector Learning Mechanism", *IEEE Trans. Fuzzy Syst.,* vol.14, no.2, pp.222-231, 2006.

[10] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications for modeling and control", *IEEE Trans. Syst., Man. Cybern.*, vol.15, pp.116-132, 1985.

[11] H. Ishibuchi, k. Nozaki, H. Tanaka, Y. Hosaka, and M. Matsuda, "Empirical Study on Learning in Fuzzy Systems by Rice Test Analysis", *Fuzzy Sets and System*, vol.64, pp.120-144, 1994.

[12] J. Ryu and S. Won, "Partitioning of Linearly Transformed Input space in Adaptive Network Based Fuzzy Inference System", *IEICE Trans. INF. and SYST.*, vol. E84-D, no.1,pp. 213-216, 2001.

[13] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees", Wadsworth, Inc., Belmont, Califonia, 1984.

[14] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples", *IEEE Trans. Syst., Man. Cybern.*, vol.22, pp.1414-1427, 1992.

[15] C. T. Sun, and J. S. Jang, "A neuro-fuzzy classifier and its applications", *Proc. of IEEE Int. Conf. Fuzzy Syst.*, San Francisco, I, pp.94-98, 1993.

[16] M. Sugeno and T. Yasukawa, "A Fuzzy-logic-based Approach to Qualitative Modeling", *IEEE Trans. on Fuzzy Sys.*, vol.1, no.1, pp.7-31, 1993.

[17] R. R. Yager and D.P. Filev, "Approximate Clustering via the Mountain method", *IEEE Trans. Syst., Man. Cybern.*, vol.24, no.8, pp.1279-1284, 1994.

[18] Y. H. Joo, H. S. Hwang and K. B. Kim, "Fuzzy System Modeling by Fuzzy Partition and GA Hybrid Schemes", *Fuzzy Sets and System*, vol.86, pp.279-288, 1997.

[19] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algrithm", New York : Plenum, 1981.

[20] C. F. Juang and C. T. Lin "An On-Line Selt-Constuctin Neural Fuzzy Inference Network and Its Applications", *IEEE Trans. Fuzzy Syst.*, vol.6, pp.12-32, 1998.

[21] V. N. Vapnic, "The Nature of Statistical Learning Theory", *Spring-Verlag*, 1995.

[22] B, Scholkopf and A. J. Smola, "Learning with Kernels", *MIT Press*, 2002.

[23] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and other kernel-based learning methods", *Cambridge University Press*, 2000.

[24] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An Introduction to Kernel-Based Learning Algorithms", *IEEE Trans. Neural Network*, vol.12. no.2, pp.181-201, 2001.

[25] J. Shawe-Taylor and N. Cristianini, "Kernel Methods for Pattern Analysis", *Cambridge University Press*, 2004.

[26] Baudat G and Anouar F, "Kernel-based Methods and Function Approximation", *Proc. of Int. Joint Conf. on Neural Networks*, pp.1244-1249, 2001.

[27] M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine", *Journal of Machine Learning Research*, vol.1, pp.211-244, 2001.

[28] W. C. Chan, W. C. Chan, K. C. Cheung and C. J. Hariss, "On the Modeling of Nonlinear Dynamic Systems Using Support Vector Neural Networks", *Engineering Application of Artifical Intelligence*, vol.14, pp.105-113, 2001.

[29] J. H. Chiang and P. Y. Hao, "Support Vector Learning Mechanism for Fuzzy Rule-Based Modeling: A New Approach", *IEEE Trans. Fuzzy Syst.,* vol.12, no.1, pp.1-12, 2004.

[30] Y. Chen and J. Z. Wang, "Support Vector Learning for Fuzzy Rule-Based Classification Systems", *IEEE Trans. Fuzzy Syst.,* vol.11, no.6, pp.716-728, 2003.

[31] Y. H. Liu and Y. T. Chen, "Face Recognition Using Total Margin-Based Adaptive Fuzzy Support Vector Machines", *IEEE Trans. Neural Networks*, vol.99, no.1, 2006. Accepted for future publication.

[32] C. A. Waring and L. Xiuwen, "Face Detection Using Spectral Histograms and SVMs", *IEEE Trans. Systems, Man and Cybernetics*, Part B, vol.35, no.3, pp.467 - 476. 2005.

[33] P. F. Yeung, H. S. Wong and B. Ma, "Relevance Vector Machine for Content-based Retrieval of 3D Head Models", *Proc. Ninth Intern. Conf. of Information Visualisation*, pp.425-429, 2005.

[34] Q. Liu, H. Lu and S. Ma, "Improving Kernel Fisher Discriminant Analysis for Face Recognition", *IEEE Trans. Circuits and Systems for Video Technology*, vol.14. no.1, pp.42-49, 2004.

[35] S. Mitra and Y. Hayashi, "Bioinformatics with Soft computing", *IEEE Trans. Systems, Man and Cybernetics, Part C*, vol.36, no.5, pp.616-635, 2006.

[36] K. B. Duan, J. C. Rajapakse, H. Wang and F. Azuaje, "Multiple SVM-RFE for gene selection in cancer classification with expression data", *IEEE Trans. Nano Bioscience*, vol.4, no.3, pp.228-234, 2005.

[37] S. Li and T. C. Eng, "Dimension Reduction-based Penalized Logistic Regression for Cancer Classification Using Microarray Data", *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol.2, no.2, pp.166-175. 2005.

[38] A. Basu and C. Walters, M. Shepherd, "Support Vector Machines for Text Ctegorization", *Proc. of 36th Annual Hawaii Intern. Conf. of System Sciences*, no.7. 2003.

[39] F. Sun and M. Sun, "A New Transductive Support Vector Machine Approach to Text Categorization", *Proc. IEEE Intern. Conf. of Natural Language Processing and Knowledge Engineering*, pp.631-635, 2005.

[40] L.J. Cao and F. E. H. Tay, "Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting", *IEEE Trans. Neural Networks*, vol.14, no.6, pp.1506-1518, 2003.

[41] C. H. Wu, J. M. Ho and D. T. Lee, "Travel-time Prediction with Support Vector Regression" *IEEE Trans. Intelligent Transportation Systems*, vol.5, no.4, pp.276-281, 2004.

[42] J. Q. Candela and L. K. Hansen, "Time Series Prediction Based on the Relevance Vector Machine with Adaptive Kernels", *Proc. of IEEE Intern. Conf. Acoustics, Speech, and Signal Processing*, vol.1, pp.985-988, 2002.

[43] N. Nikolaev and P. Tino, "Sequential Relevance Vector Machine Learning from Time Series", *Proc. of IEEE Intern. Joint Conf. of Neural Networks*, vol.2, pp.1308-1313, 2005.

[44] S. Avidan, "Support Vector Tracking", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.26, no.8, pp.1064-1072, 2004.

[45] A. Agarwal and B. Triggs, "3D Human Pose from Silhouettes by Relevance Vector Regression", *Proc. of IEEE Conf. of Computer Vision and Pattern Recognition*, vol.2, pp.882-888, 2004.

[46] S. Chen, S. R. Gunn and C. J. Harris, "The Relevance Vector Machine Technique for Channel Equalization Application", *IEEE Trans. Neural Networks*, vol.12, no.6, pp.1529-1532, 2001.

[47] M. Martinez-Ramon, J. L. Rojo-Alvarez, G. Camps-Valls, J. Munoz-Mari, A. Navia-Vazquez, E. Soria-Olivas, and A. R. Figueiras-Vidal, "Support Vector Machines for Nonlinear Kernel ARMA System Identification", *IEEE Trans. Neural Networks*, vol.17, no.6, pp.1617- 1622, 2006.

[48] X. G. Wang and W. H. Li, "Nonlinear System Identification using Least Squares Support Vector Machines", *Proc. of Intern. Conf. of Neural Networks*, pp.414-418, 2005.

[49] V. N. Vapnic, "Statistical Learning Theory", *Spring-Verlag*, 1999.

[50] A. J. Smola and B. Scholkopf, "A Tutoral on Support Vector Regression", *Statistics and Computing*, vol.14, no.3, pp.199-222, 1998.

[51] H. O. Wang, K. Tanaka, and M. F. Griffin, "Parallel Distributed Compensation of Nonlinear Systems by Takagi-Sugeno Fuzzy Model", *Proc. of FUZZ-IEEE/IFES* , pp.531-538, 1995.

[52] K. Tanaka and M. Sugeno, "Stability Analysis and Design of Fuzzy Control Systems," *Fuzzy Sets Syst.*, vol.45, pp.135-156, 1992.

[53] R. R. Yager and L. A. Zadeh, "Fuzzy Sets, Neural Networks, and Soft Computing", Thomson Learning, 1994.

[54] H. T. Nguyen and M. Sugeno, "Fuzzy Systems: Modeling and Control", Kluwer, Boston, MA, 1998.

[55] C. A. P. Reyes, "Coevolutionary Fuzzy Modeling", *Lecture Notes in Computer Science*, vol.3204, Springer, 1998.

[56] R. R. Yager and A.V. Kreinvich, "Universal Approximation Theorem for Uninorm-based Fuzzy Systems modeling", *Fuzzy Sets and Systems*, vol.140, no.22, pp.331-339, 2003.

[57] M. Sugeno and G. T. Kange, "Structure Identification of Fuzzy Model", *Fuzzy Sets andSystmes*, vol.28, pp.15-33, 1988.

[58] E. H. Mamdani, "Applications of Fuzzy Algorithms for Control of a Simple Dynamic Plant", *Proc. of IEEE*, vol.121, no.12, pp.1585-1588, 1974.

[59] K. Tanaka et al., "Self-organizing Fuzzy Identification of a Municipal Refuse Incinerator", *Proc. of Int. Fuzzy Syst. Intell. Contr.*, pp.13-22, 1993.

[60] K. Tanaka, M. Sano and H. Watanabe, "Modeling and Control of Carbon Monoxide Concentration Using Aneuro-fuzzy Technique", *IEEE Trans. Fuzzy Systems*, vol.3, no.3, pp.271-279, 1995.

[61] L. A. Zadeh, "Fuzzy Set", *Information and Control*, vol.8, pp.338-353, 1996.

[62] J. R. Jang, C. Sun and E. Mizutani, “Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence”, Prentice Hall, 1996.

[63] J. M. Mendel, “Fuzzy Logic Systems for Engineering: A tutorial”, *Proc. of IEEE*, vol.83, no.3, pp.345-377, 1995.

[64] L. A. Zadeh, “The Concept of a Linguistic Variable and Its Applications to Approximate Ressoning”, *Information Science*, Part I, vol.8, pp.199-249; Part II, vol.8, pp.301-357; Part III, vol.9, pp.43-80, 1975.

[65] S. R. Gunn, “Support Vector Machines for Classification and Regression”, Image Speech and Intelligent Systems Research Group, University of Southampton, 1997.

[66] T. Evgeniou, M. Pontil and T. Poggio, “Regularization Networks and Support Vector Machines”, *Advances in Comutional Mathematics*, 1999.

[67] V. N. Vapnik and A. Y. Chervonenkis, “On the uniform convergence of relative frequences of events to their probabilities”, *Theoretical Probability and Its Applications*, vol.17, no.2, pp.264-280, 1971.

[68] Mercer, “Functions of positive and negative type and their connection with the theory of integral equations”, *Philos. Trans. Roy. Soc.* A209: pp.415-446, 1909.

[69] J. S. R. Jang and C. T. Sun, “Neuro-fuzzy Modeling and Control”, *Proc. of IEEE*, vol.83, pp.378-406, 1995.

[70] P. M. L. Drezet and R. F Harris, “Support Vector Machines for System Identification”, *Proc. of UKACC Int. Conf. on Control*, pp.688-692, 1998.

[71] A. Gretton, A. Doucet, R. Herbrich, P. J. W. Rayner and B. Scholkopf, “Support Vector Regression for Blackbox System Identification”, *Proc. of IEEE signal Processing Workshop on Statistical Signal Processing*, pp.341-344, 2001.

[72] J. A. K. Suykens, "Support Vector Machines: a Nonlinear Modelling and Control Perspective", *European Journal of Control, Special Issue on Fundamental Issues in Control*, vol.7, no.2-3, pp.311-327, 2001.

[73] J. L. Rojo Alvarez, M. Martinez-Ramon, M. DePrado-Cumplido, A. Artes-Rodriguez and A. R. Figueiras-Vidal, "Support Vector Method for ARMA System Identification", *IEEE Trans. on Signal Processing*, vol.52, no.1, pp.155-164, 2004.

[74] J. L. Rojo-Alvarez, G. Camps-Valls, M. Martinez-Ramon, E. Soria-Olivas, A. Navia-Vazquez and A. R. Figueiras-Vidal, "Support Vector Machines Framework for Linear Signal Processing", *IEEE Trans. on Signal Processing*, vol.85, no.12, pp.2316-2326, 2005.

[75] J. T. Jeng and T. T. Lee, "Support Vector Machines for the Fuzzy Neural Networks", *Proc. of IEEE conf. SMC*, vol.6, pp.115-120, 1999.

[76] B. E. Boser, I. M. Guyon and V. N. Vapnik, "A training algorithm for optimal margin classifiers", *Proc. of 5th Annual ACM on COLT*, pp.144-152, 1992.

[77] I. Guyon, B. Boser and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers", *Advances in Neural Information Processing Systems*, vol.5, pp.147-155, 1993.

[78] Z. Uykan, C. Guzelis and M. E. Celebi, "Analysis of Input-output Clustering for Determining Center of RBFN", *IEEE Trans. Neural Network*, vol.11, no.4, pp.851-858. 2000.

[79] A. F. Gomez-Skarmeta, M. Delgado, M. A. Vila, "About the Use of Fuzzy Clustering Techniques for Fuzzy Model Identification", *Fuzzy Sets and Systems*, vol.106, pp.179-188, 1999.

[80] S. Kim, "A Study on the Development of Skew Compensation Models inthe Plate Mill", M0208335, POSTECH, 2000.

[81] C. C. Wong and C. C. Chen, “A Hybrid Clustering and Gridient Descent Approach for Fuzzy Modeling”, *IEEE Trans. on Sys. Man and Cybernetics Part B: Cybernetic.*, vol.29, no.6, pp.686-693, 1999.

[82] P. J. C. Branco and J. A. Dente, “Noise Effects in Fuzzy Modelling Systems: Three Case Studies”, *Computational Intelligence and Applications*, World Scientific and Engineering Society Press, Danvers, USA, pp.103-108, 1999.

[83] L. Holmstrom and P. Koistine, “Using Additive Noise in Back-Propagation Training”, *IEEE Trans. Neural Networks,* vol.3, no.1, pp.24-38, 1992.

[84] G. N. Karystinos and D. A. Pados, “On Overfitting, Generalization, and Randomly Expanded Training Sets,” *IEEE Trans. Neural Networks,* vol.13, no.5, pp.1050-1057, 2000.

[85] E. W. M. Lee, C. P. Lim, R. K. K. Yuen, and S. M. Lo, “A Hybrid Neural Network Model for Noisy Data Regression”, *IEEE Trans. Syst. Man. Cybern. Part B:Cybern.,* vol.34, no.2, pp.951-960, 2004.

[86] J. Yen and L. Wang, “Application of Statistical Information Criteria for Optimal Fuzzy Model Construction”, *IEEE Trans. Fuzzy Syst.,* vol.6, no.3, pp.362-372, 1998.

[87] T. Matsumoto, Y. Nakajima, J. Sugi, and Hl. Hamagishi, “Reconstructions and Predictions of Nonlinear Dynamical Systems: A Hierarchical Bayesian Approach”, *IEEE Trans. Signal Processing*, vol.49, no.9, pp.2138-2155, 2001.

[88] I. Rivals and L. Personnaz, “Neural-Network Construction and Selection in Nonlinear Modeling”, *IEEE Trans. Neural Networks*, vol.14, no.4, pp.804-819, 2003.

[89] W. Chu, S. S. Keerthi and C. J. Ong, “Bayesian Support Vector Regression Using a Unified Loss Function”, *IEEE Trans. Neural Networks*, vol.15, no.1, pp.29-44, 2004.

[90] C. M. Bishop and M. E. Tipping, “Variational Relevance Vector Machines”, *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, pp.46-53, 2000.

[91] M. E. Tipping and A. Faul, “Fast Marginal Likelihood Maximisation for Sparse Bayesian Models”, *Proc. of the 9th Int. Workshop on Artifical Intelligence and Statistics*, 2003.

[92] L. Wei, Y. Yang, R. M. Nishikawa and Y. Jiang, “A Study on Serveral Machine-Learning Methods for Classification of Malignant and Benign Clustered Microcalcifications”, *IEEE Trans. Medical Imaging*, vol.24, no.3, pp.371-380, 2005.

[93] D. J. C. MacKay, “Bayesian interpolation”, *Neural Computation*, vol.4, no.3, pp.415-445, 1992.

[94] D. J. C. MacKay, “A Practical Bayesian Framework for Backpropagation Network”, *Neural Computation*, vol.4, no.3, pp.448-472, 1992.

要 約

A Study on the Fuzzy Modeling of Nonlinear Systems Using Kernel Machines

近年，入出力データからファジィモデルの最適な構造およびパラメータを選定し，これを自動生成する手法の開発がファジィ推論の研究における重要な課題の一つとされている．一方，入出力データに基づく非線形システムのモデリングに対してカーネルマシンを用いた手法が注目されている．そこで，本研究では、最新のカーネルマシンを適用してファジィ推論システムを自動的に構築するための新しい手法について検討を加えた．すなわち，カーネルマシンとして拡張Support Vector Machine (SVM)，拡張Feature Vector Selection (FVS)および拡張Relevance Vector Machine (RVM)の３種類を提案し，これらを用いてファジィ推論エンジンにおける最適なファジィルールの個数およびメンバーシップ関数のパラメータを自動的に生成するシステムの構築を図った．なお，ここでは基本的なファジィ推論システムとしてTakagi-Sugeno (TS) ファジィモデルを用いた．

第１章では，本研究の背景，目的，本論文の構成などについて述べた．

第２章では，ファジィシステム，統計学的な学習理論およびカーネル特徴空間など，本研究に関連する基本的事項について述べた．ファジィシステムでは，ファジィ集合とロジック，ファジィ推論システム，TSファジィモデルについて述べた．統計学的な学習理論では，汎化エラー，empirical risk minimization，structure risk minimizationの原理について述べた．また，カーネル特徴空間では特徴空間における学習方法およびカーネル関数について基礎的事項を解説した．

第３章では，拡張SVMを用いたファジィ推論システムについて述べた．すなわち，拡張SVMを用いたファジィ推論システムを新たに提案し，その構造および学習アルゴリズムについて詳細に述べた．また，幾つかの例題に対して本システムを適用し，提案手法の有効性を確認した．

第４章では，拡張FVSを用いたファジィ推論システムについい述べた．拡張FVSは，主に入力変数の線形変換とカーネル関数によるカーネルマッピング等から構成されてい

る．これは，第３章で述べた拡張SVMより学習速度が速く，また入力変数を線形変換することによってより適切なカーネル関数の選択が可能となる等の特徴を有している．最後に，例題を用いて本手法の有効性を確認した．

第5章では，拡張RVMを用いたファジィ推論システムについて述べた．これは，第３章で述べた拡張SVMを用いたシステムに比べてファジィルールの個数を減らすことができ，また第4章で述べた拡張FVSのように入力変数に線形変換を施すことなくより適したカーネル関数を選択できる等の特長を有している．ここではその構造および学習アルゴリズムについて詳述し，また，本ファジィ推論システムを非線形ダイナミックシステムおよびロボットアームデータに関する例題に適用し，その有効性を検証した．

第６章は結論であり，各章で得られた内容をまとめ，本研究の成果を総括した．

요 약 문

A Study on the Fuzzy Modeling of Nonlinear Systems Using Kernel Machines

본 논문에서는 입출력 데이터를 가진 비선형 시스템의 모델링을 하기 위한 kernel machines을 이용한 퍼지 추론 시스템의 새로운 접근 방법들을 제안한다. 입출력 데이터로부터 퍼지 모델의 최적의 구조와 매개변수를 선택하는 것은 뉴로퍼지 시스템의 모델링에서 아주 중요한 이슈이다. 본 논문에서는 이러한 문제를 해결하기 위하여 퍼지 추론 엔진으로 최신의 kernel machines을 제안한다. 제안된 퍼지추론 시스템에서는 확장된 Support Vector Machine (SVM), Feature Vector Selection (FVS) 그리고 Relevance Vector Machine (RVM)과 같은 kernel machine을 이용하여 퍼지 룰의 개수와 멤버쉽 함수의 매개변수를 자동적으로 결정하는 장점이 있다.

Kernel machine은 기계학습과 kernel 함수와 같은 2개의 모듈로 구성되어 있다. 기계학습은 일종의 학습이론이고, kernel 함수는 입력 데이타를 높은 차원의 특징공간으로 투영한다. Kernel machine은 입력공간을 특징공간으로 비선형 투영함으로써 입력 데이타를 선형적으로 해석할 수 있고, 모델의 크기와 에러를 동시에 고려하여 일반화 문제를 다룰 수 있다.

제안된 퍼지추론 시스템은 Tagaki-Sugeno 퍼지 모델의 구조를 가진다. 특히 퍼지 룰의 개수는 확장된 SVM, FVS과 RVM들의 학습과정에서 사면하강법을 이용하여 퍼지 멤버쉽 함수의 매개변수를 조정함으로써 감소될 수 있다. 일단 퍼지모델의 룰과 멤버쉽 함수의 매개변수가 결정되면 최소자승법을 이용하여 TS 퍼지모델의 후건부의 매개변수를 결정한다. 이때 비퍼지화 방법은 무게중심법을 이용한다.

본 논문에서는 확장된 SVM, FVS과 RVM과 같은 3종류의 kernel machine 각각을 이용한 퍼지추론시스템들에 대해서 비선형 함수의 모델링에 적용하여 기존에 제안된 퍼지

모델링 방법들의 성능과 비교하여 그 타당성과 유효성을 검정하였다.

그리고, 비록 제안된 방법들이 기존의 연구들과 비교하여 퍼지모델의 구조와 매개변수를 자동적으로 결정하고 overfiting의 문제를 해결하기 위한 일반화 방향으로 접근했다고 할지라도, 비선형 투영에 사용된 kernel 함수의 적당한 종류 선택의 문제와 기계학습 시에 소요되는 긴 계산 시간 문제는 향후 해결되어야 할 과제들이다. 또 제안된 방법은 보다 많은 실제 비선형 시스템의 모델링에 적용될 필요가 있다.

謝 辞

本研究は，著者が慶應義塾大学大学院理工学研究科在学中，菅泰雄教授のもとで行われたものであります．先生より賜ったご指導とご鞭撻に対し厚く御礼申し上げます．また，本論文を完成するにあたり，ご多忙中の貴重な時間を割かれてご査読され，有益なご助言を賜りました慶應義塾大学理工学部の三井公之教授，村上俊之教授，萩原将文教授に厚く御礼申し上げます．また，貴重なご意見，ご指摘を頂いた慶應義塾大学理工学部の山田邦博教授，澤田達男教授，三木則尚専任講師に厚く御礼申し上げます．同塾の理工学部学部長である稲崎一郎教授には，研究面および留学生活面で多大な支えを頂きました．心より御礼申し上げます．

筑波大学の鬼沢武久教授，愛媛大学工学部谷脇滋宗助教授，慶應義塾大学理工学部の高橋正樹助手，韓国の昌原国立大学制御計測工学科の教授の方々，全州大学の金建會教授，浦項工科大学(POSTECH)の元相喆教授及びPOSTECHの制御研究室の皆様から研究に関する様々なご助言を賜りました．深く感謝します．

本研究を遂行するにあたって苦楽を共にした菅研究室の卒業生である高橋雄介氏，荒谷真一氏，江木畠淳之氏，倉橋大樹氏，また，多大なご協力を頂きました村松正浩先輩，Ario Sunar Baskoro氏，谷澤拓海氏，足利嘉彦氏，金泰元先輩および菅研究室の皆様に深く感謝します．

大学院また留学生活のなかでは，多くの先輩に恵まれ，多大なご助言およびご協力を頂きました．特に朴敏圭先輩，朱行植先輩，鄭忠考先輩，愼祥揆氏，韓潤燮氏，宋哲源氏および慶應義塾大学韓国留学生会の皆様に深く感謝します．

何時も励まして下さった新東京バプテスト協会の金東源牧師および教会の家族皆様に心より感謝致します．留学生として来日して以来，丸紅(株)等々力寮の事務室の皆様に長い間心温かい支えを頂きました．深く感謝します．

後期博士課程の在学中には，大韓民国政府の韓国科学財団の国費奨学金を賜りました．また，慶應義塾大学の推薦で川嶋記念スカラーシップ基金の奨学金及び本庄国際奨学財団の奨学金も賜りました．ご協力頂いた慶應義塾大学の皆様と川嶋記念スカラーシップ基金，韓国科学財団及び本庄国際奨学財団に深く感謝します．

最後に，長い留学生活において色々な面から支えてくれた母親，妻の両親，兄，姉の家族，そして親戚の皆様厚く御礼申し上げます．特に，研究への取り組みを暖かく支えてくれた妻(徐貞華)に心より感謝します．

Curriculum Vitae

Personal

Name:	Jongcheol Kim
Date of Birth:	July 25, 1972
Birth Place:	Gosung
Email:	jaejaemail@hanmail.net
Home Address:	Whasan-ri 144-1, Maam-myun, Gosung-gun, Korea
Office Address:	School of Integrated Design Engineering Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, Japan.

Education

2003.4 – 2007.3: Ph.D. in School of Integrated Design Engineering,
Keio University, Yokohama, Japan.
Major: Robotics and Machatronics
Thesis Topic: A Study on the Fuzzy Modeling of Nonlinear Systems Using Kernel Machines
Advisor: Yasuo Suga

2000.3 – 2002.2: M.S. in Electrical and Electronic Engineering, POSTECH, Pohang, Korea.
Major: Control and Instrumentation
Thesis Topic: A Study on the Modeling of Nonlinear Systems Using Support Vector Fuzzy Inference System
Advisor: Sangchul Won

1993.3 – 2000.2: B.S. in Control and Instrumentation Engineering, Changwon National University, Changwon, Korea.
Major: Control and Instrumentation Engineering
Bachelor Topic: Phase Controlled Rectifier (PCR) By Using PC Interface
Advisor: Eui-Ho Song

Research Experience

- "Design: Paradigm Shift from Intelligence to Life", (The 21st century COE Program, Keio University, System Design: From Intelligence to Life, Cooperation Robot Group, Japan), (Keio University: 2004.4 - 2006.3).
- "Optical and Electronic Device Technology for Access Network", (The 21st century COE Program, Keio University, Graduated School Information, Electrical Engineering, and Electronics, Japan), (Keio University: 2003.4 - 2004.3).
- Research student in School of Integrated Design Engineering, Faculty of Science and Technology, Keio University, Japan,(Keio University: 2002.4 - 2003.3).
- "Development of Measurement System by Using Wave Nonlinearity", (POSCO: 2000.3 - 2001.7).

Activities

2006.9:	International Program Committee Member in SCIS&ISIS2006,
2003. 4. - 2006. 3:	Research Assistant in Keio University
2005. 2. - 2005. 12:	Oversea science and technology informant in KISTI
2000. 3. - 2002. 2:	Teaching Assistant/Research Assistant in POSTECH

Awards and Honors

2005.3.29:	Hujihara Award in Keio University for Outstanding Research
2004.9.22:	Best Paper Award in SCIS & ISIS2004
2005.4 - 2007.3:	Honzo International Scholarship Foundation, Japan
2003.4 - 2005.3:	Graduate Study Abroad Scholarship supported by Korea Science and Engineering Foundation (KOSEF), Korea

Publications

- **Paper submitted to International Journal**

1. Jongcheol Kim and Suga Yasuo, "An Omnidirectional Vision Based Moving Obstacle Detection Algorithm in Mobile Robot", *International Journal Control Automation and System*,(submitted).

- **Accepted Paper for Publication in International Journal**

1. Jongcheol Kim, Masahiro Muramatsu, Yuichiro Murata and Yasuo Suga, "Omnidirectional Vision-Based Ego-Pose Estimation for An Autonomous In-Pipe Mobile Robot," *Advanced Robotics*, Vol. 21. No. 5. (In press, 2007,5).

2. Jongcheol Kim, Sangchul Won and Yasuo Suga, "A New Approach to Fuzzy Modeling of Nonlinear Dynamic Systems with Noise: Relevance Vector Learning Mechanism", *IEEE Trans. Fuzzy on Systems*, Vol.14,

No.2, pp.222-231, April, 2006.

3. Jongcheol Kim, Taewon Kim and Yasuo Suga, “New Approach to Fuzzy Modeling using an Extended Kernel Method”, *IEICE Trans. Fudamentals*, Vol.E86-A, No.9, pp.2262-2269, 2003.
4. Taewon Kim, Takeshi Suto, Jyunya Kobayashi, Jongcheol Kim and Yasuo Suga, ”Automatic Welding System using Speed Controllable Autonomous Mobile Robot”, *JSME International Journal Series A*, Vol.49, No.1, pp.53-57, Jan. 2006.
5. Taewon Kim, Jongcheol Kim, Yu Hasegawa and Yasuo Suga, “Welding of AZ31B Magnesium Alloy by YAG Laser/ TIG Arc Hybrid Welding Process”, *Material Science Forum, Designing, Processing and Properties of Advanced Engineering Materials*, Vol.449-452, Part 1, pp.417-420, March, 2004.

- **Accepted Paper for Publication in Domestic Journal**

1. Taewon Kim, Jongcheol Kim, Yu Hasegawa and Yasuo Suga, “Application of YAG Laser TIG Arc Hybrid Welding to Thin AZ31B Magnesium Alloy Sheet,” *Journal of High Temperature Society*, Vol.30, No.4, pp.220-226, July, 2004.

- **International Conference Paper**

1. Jongcheol Kim, Yasuo Suga and Sangchul Won, “New Approaches to Fuzzy Inference System Using Kernel Machines”, *Proceedings of the Joint 3nd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on Advanced Intelligent Systems*, Tokyo, Japan, Sept., pp.1934-1939, 2006. (Invited Talk)

2. Jongcheol Kim, Takumi Tanizawa, Yasuo Suga and N Miki, "Moving Object Detection in Mobile Robot with Omnidirectional Camera", *SICE Annual Conference* Okayama, Japan, pp.1696-1699, August, 2005.
3. Taewon Kim, Takeshi Suto, Jyunya Kobayashi, Jongcheol Kim and Yasuo Suga, "Automatic Welding System using Speed Controllable Autonomous Mobile Robot", *JSME/ASME International Conference*, AWB-04, Seattle, USA, Jun, 2005.
4. Jongcheol Kim, Taewon Kim and Yasuo Suga, "New Fuzzy Modeling of Nonlinear Dynamic System Using a Relevance Vector Machine," , *Joint 2nd International Conference on Soft Computing and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems*, Yokohama, Japan, WP-8-1, Sept. 2004. (Best Paper Award)
5. Taewon Kim, Jongcheol Kim, Yu Hasegawa and Yasuo Suga, "Weldability of Magnesium Alloy by YAG Laser / TIG Arc Hybrid Welding Process", *International Institute of Welding Annual Meeting*, Osaka, Japan, July, 2004.
6. Jongcheol Kim, Sangchul Won, and Yasuo Suga, "New Fuzzy Inference System Using a Kernel-based Method", *International Conference on Control, Automation and Systems*, Gyeongju, Korea, pp.2393-2398, Oct. 2003.
7. Taewon Kim, Jongcheol Kim, Yu Hasegawa and Yasuo Suga, "Welding of AZ31B Magnesium Alloy by YAG Laser/ TIG Arc Hybrid Welding Process", *3rd International Symposium on Designing, Processing and Properties of Advanced Engineering Materials*, Jeju, Korea, Nov., 2003.
8. Jongcheol Kim and Sangchul Won, "New Fuzzy Inference System using a Support Vector Machine", *Proc. the 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada USA, pp.1349-1354, Dec., 2002.

• Domestic Conference Paper

1. Jongcheol Kim and Sangchul Won, "H^∞ Fuzzy Dynamic Output feedback Controller Design with Pole Placement Constraints", International Conference on Control, Automation and Systems, Cheju, Korea, pp.2568-2571, 2001.
2. Dukbum Shin, Jongcheol Kim and Sangchul Won, "Development of the New Shape Control Algorithm with Strip Thickness Decoupling in Hot Strip Mill", International Conference on Control, Automation and Systems, Cheju, Korea, pp.2545-2548, 2001.
3. Jongcheol Kim and Sangchul Won, "H∞ Fuzzy Dynamic Output Feedback Controller Design With Pole Placement Constraint", 4th Joint Conference of Division of Daegu-Kyoungbuk of ICASE and Technical Society of Steel Process Control, Pohang, Korea,Vol.1, No.1, pp.24-28, 2001.
4. 金 泰元, 金 宗哲, 長谷川, 悠 菅 泰雄, "ステンレス鋼及びマグネシウム合金の薄板レーザ／アークハイブリッド溶接," 第12回機械材料・材料加工技術講演会, 2004.
5. 谷澤 拓海, 金 宗哲, 菅 泰雄, "全方位カメラを利用して移動体検出," 第11回画像センシングシンポジウム (SSII2005), 横浜, pp.499-502, 2005.
6. Jongcheol Kim, Takumi Tanizawa and Yasuo Suga, "Moving Object Detection Using Optical Flow in Mobile Robot with an Omnidirectional Camera," The 23rd Annual Conference of the Robotics Society of Japan, Yokohama, 1B17, 2005.
7. 金 宗哲, 谷澤 拓海, 足利 嘉彦, 菅 泰雄, "全方位カメラによるオプティカルフローを用いた移動体認識," 第12回画像センシングシンポジウム (SSII2006), 横浜, pp.497-500, 2006.
8. 倉橋 大樹, 金 宗哲, 堀田 桐子, 村松 正浩, 菅 泰雄, "Visual Hullを用いた物体の3次元形状モデリング," 第12回画像センシングシンポジウム (SSII2006), 横浜, pp.175-180, 2006.