LEARNING AND IDENTIFICATION OF FUZZY SYSTEMS

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2010

By

Shin-Jye Lee

School of Computer Science

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis concentrates on learning and identification of fuzzy systems, and this thesis is composed about learning fuzzy systems from data for regression and function approximation by constructing complete, compact, and consistent fuzzy systems.

Fuzzy systems are prevalent to solve pattern recognition problems and function approximation problems as a result of the good knowledge representation. With the development of fuzzy systems, a lot of sophisticated methods based on them try to completely solve pattern recognition problems and function approximation problems by constructing a great diversity of mathematical models. However, there exists a conflict between the degree of the interpretability and the accuracy of the approximation in general fuzzy systems. Thus, how to properly make the best compromise between the accuracy of the approximation and the degree of the interpretability in the entire system is a significant study of the subject.

The first work of this research is concerned with the clustering technique on constructing fuzzy models in fuzzy system identification, and this method is a part of clustering based learning of fuzzy systems. As the determination of the proper number of clusters and the appropriate location of clusters is one of primary considerations on constructing an effectively fuzzy model, the task of the clustering technique aims at recognizing the proper number of clusters and the appropriate location as far as possible, which gives a good preparation for the construction of fuzzy models. In order to acquire the mutually exclusive performance by constructing effectively fuzzy models, a modular method to fuzzy system identification based on a hybrid clustering-based technique has been considered. Due to the above reasons, a hybrid clustering algorithm concerning input, output, generalization and specialization has hence been

introduced in this work. Thus, the primary advantage of this work is the proposed clustering technique integrates a variety of clustering properties to positively identify the proper number of clusters and the appropriate location of clusters by carrying out a good performance of recognizing the precise position of each dataset, and this advantage brings fuzzy systems more complete.

The second work of this research is an extended work of the first work, and two ways to improve the original work have been considered in the extended work, including the pruning strategy for simplifying the structure of fuzzy systems and the optimization scheme for parameters optimization. So far as the pruning strategy is concerned, the purpose of which aims at refining rule base by the similarity analysis of fuzzy sets, fuzzy numbers, fuzzy membership functions or fuzzy rules. By other means, through the similarity analysis of which, the complete rules can be kept and the redundant rules can be reduced probably in the rule base of fuzzy systems. Also, the optimization scheme can be regarded as a two-layer parameters optimization in the extended work, because the parameters of the initial fuzzy model have been fine tuning by two phases gradation on layer. Hence, the extended work primarily puts focus on enhancing the performance of the initial fuzzy models toward the positive reliability of the final fuzzy models. Thus, the primary advantage of this work consists of the simplification of fuzzy rule base by the similarity-based pruning strategy, as well as more accuracy of the optimization by the two-layer optimization scheme, and these advantages bring fuzzy systems more compact and precise.

So far as a perfect modular method for fuzzy system identification is concerned, in addition to positively solve pattern recognition problems and function approximation problems, it should primarily comprise the following features, including the well-understanding interpretability, low-degree dimensionality, highly reliability, stable robustness, highly accuracy of the approximation, less computational cost, and maximum performance. However, it is extremely difficult to meet all of these conditions above. Inasmuch as attaining the highly achievement from the features above as far as possible, the research works of this thesis try to present a modular method concerning a variety of requirements to fuzzy systems identification.

DECLARATION

I declare that no portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification at this or any other university or other institute of learning

Shin-Jye Lee

COPYRIGHT STATEMENT

University Library's regulations (see http://www.manchester.ac.uk/ library/aboutus/regulations) and in The University's policy on presentation of Theses

ACKNOWLEDGEMENTS

Firstly, special thanks are undoubtedly due to my supervisor, Dr Xiao-Jun Zeng, for his continuous support and encouragement throughout my PhD. He has been my main source of my inspiration and a reliable source of my knowledge for my PhD work.

Thanks to my parents for their support and encouragement.

Thanks to Ms. Hui-Shin Wang for her support and encouragement.

Thanks to Dr. Michal Wozniak for his support and encouragement.

I would like to thank to my colleagues and friends in the MLO group in the School of Computer Science at the University of Manchester for their advice, help, and friendship. Thanks also to all those past and present members of the MLO group for providing a great working atmosphere to work on for the final year of my PhD.

.

Finally, many thanks to all of my friends for their great support.

LIST OF PUBLICATIONS RELATED TO THE THESIS

Lee, S. J. and Zeng, X. J. (2008). A modular method for estimating null values in relational database systems. *In Proceedings of the Eighth International Conference on Intelligent Systems Design and Applications*, vol. 2. Kaohsiung, Taiwan, Republic of China, pp. 415-419.

Lee, S. J., Zeng, X. J. and Wang, H. S. (2009). Generating automatic fuzzy system from relational database system for estimating null values. *Cybernetics and Systems,* vol. 40, no.6, pp.528-548.

Lee, S. J. and Wang, H. S. (2009). A dynamic modular method for estimating null values in relational database systems. *International Journal of Computer Information Systems and Industrial Management Applications,* vol. 1, pp.249-257.

Lee, S. J. and Zeng, X. J. (2010). A three-part input-output clustering-based approach to fuzzy system identification. *In Proceedings of the Tenth International Conference on Intelligent Systems Design and Applications*, Cairo, Egypt, pp. 55-60.

Chapter 1 INTRODUCTION

Albert Einstein said, "Make everything as simple as possible, but not simpler."

With the development of information technology, computing technology has already become an important tool to help humans dealing with various kinds of tasks, and the relationship between computing and humans has therefore become an increasing issue as a result of the gradual nature of development. Being public, highly reliable, aggressive and efficient, data mining is the hottest subject of the day with a splendid future for helping humans to understand undiscovered knowledge and to further explore it. So far as exploring undiscovered knowledge is concerned, data mining is the process of extracting patterns from data, which plays an important role in discovering knowledge from data by machine learning algorithms. In addition to the fields of engineering and computer science, data mining has also been applied to a variety of fields because of its practical features of extracting useful information from data. However, the more advanced computing technologies develop and are applied, the more complex and larger size the data sets have grown. To relieve this problem interrogation of new data in accordance with Occam's razor is a promising course for researchers of machine learning and optimisation.

1.1 Machine Learning

A computer program is said to ***learn*** *from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. (Mitchell, 1997)*

Machine learning is a discipline of computer science that researches into how the computer positively acquires precise behaviour, as well as new knowledge by the learning ability of learning algorithms. Machine learning has been applied to a lot of fields because of its reliable performance of dataset analysis, such as machine perception, computer vision, natural language processing, bioinformatics, software engineering, and so on. Additionally, machine learning can be regarded as the core of artificial intelligence, playing an important role and it is also the significant component in making a computer "intelligent", making decisions by learning from current data. The learning ability is the primary concept of machine learning, and it is the key feature of so-called intelligent systems. The learning ability of an intelligent system can be identified by finding embedded rules behind datasets that allows a computer to learn from the data, the purpose of which is to make intelligent decisions by recognising complex patterns. Hence, "how to precisely learn to recognise complex patterns from data automatically" is a rudimentary course in machine learning. So far as the core of machine learning algorithms are concerned, the learning algorithms basically emphasise the generalisations learnt from the training examples, and the training examples can be known as the experience of the target system. The behaviour of the target system can be learnt using the learning algorithms on training examples. Moreover, according to the desired outcome of machine learning algorithms, the type of machine learning algorithms can be classified into (Mitchell, 1997):

- **Supervised learning** - Generates a function that maps inputs to desired outputs. In a classification problem, the learner approximates a function mapping a vector into classes by looking at input-output examples of the target function.
- **Unsupervised learning** – Different from supervised learning, unsupervised learning models a set of inputs and seeks to summarize and explain key features of the data, such as clustering.

- **Semi-supervised learning** – Combines both supervised learning (labelled training data) and unsupervised learning (unlabeled training data) to generate an appropriate function or classifier.
- **Reinforcement learning** – Learns to take action through the observation of the environment, and which reinforces learning algorithms by the feedback given by the impact in the environment.

1.2 Pattern Recognition

Pattern recognition is one of scientific topics highly relevant to machine learning and has been prevalently applied to a variety of areas of computer science such as computer vision, speech recognition, handwriting recognition, biometrics, and so on. Briefly, pattern recognition is the process of interpreting, analysing and recognising patterns by computers and mathematics, and pattern recognition aims at discovering the reasonable output for a possible input by labelling the output to a given input.

According to the type of label output, most algorithms for pattern recognition are generally categorised into supervised learning and unsupervised learning. The difference between supervised learning and unsupervised learning is that supervised learning assumes that the training data and output have been labelled, whilst unsupervised learning assumes that the training data has not been labelled. Semi-supervised learning integrates the feature of supervised learning and unsupervised learning, making use of both labelled and unlabelled data when training, and typically combines a small amount of labelled data as well as a large amount of unlabelled data. As a whole, algorithms for solving pattern recognition problems can be summarised as follows:

- Supervised learning algorithms for predicting categorical labels-Classification algorithms, which the interpretation of labels is given.

- ❖ Unsupervised learning algorithms for predicting categorical labels- Clustering algorithms, which they could return artificial (non-interpretable) class labels.

1.3 Fuzzy Theory and Fuzzy System

Fuzzy systems are knowledge-based or rule-based systems. The heart of a fuzzy system is a knowledge base consisting of the so-called fuzzy IF-THEN rules. (Wang, 1997)

Since Zadeh proposed the concept of fuzzy sets (Zadeh, 1965), fuzzy theory has been structured and gradually developed, as illustrated in Fig. 1.1 (Wang, 1997). Moreover, the concept of the fuzzy set is one of the essential components of a fuzzy system, and also plays an important role in fuzzy modelling. Fuzzy sets are sets whose elements have degrees of fuzzy membership, and each fuzzy set is associated with membership values. Basically, a fuzzy set can be defined as a pair (A, m), where A is a set and m is a membership value, $m : A \rightarrow [0, 1]$. To define the vagueness of a data set as far as possible, the main purpose of membership functions is to transform a real-value data set into a fuzzy set, and then place it at the appropriate level of each input variable in the fuzzy system. In principle, fuzzy systems are developed based on fuzzy *IF-THEN* rules with discrete or continuous membership functions or by other means. Fuzzy systems are constructed from a collection of fuzzy *IF-THEN* rules.

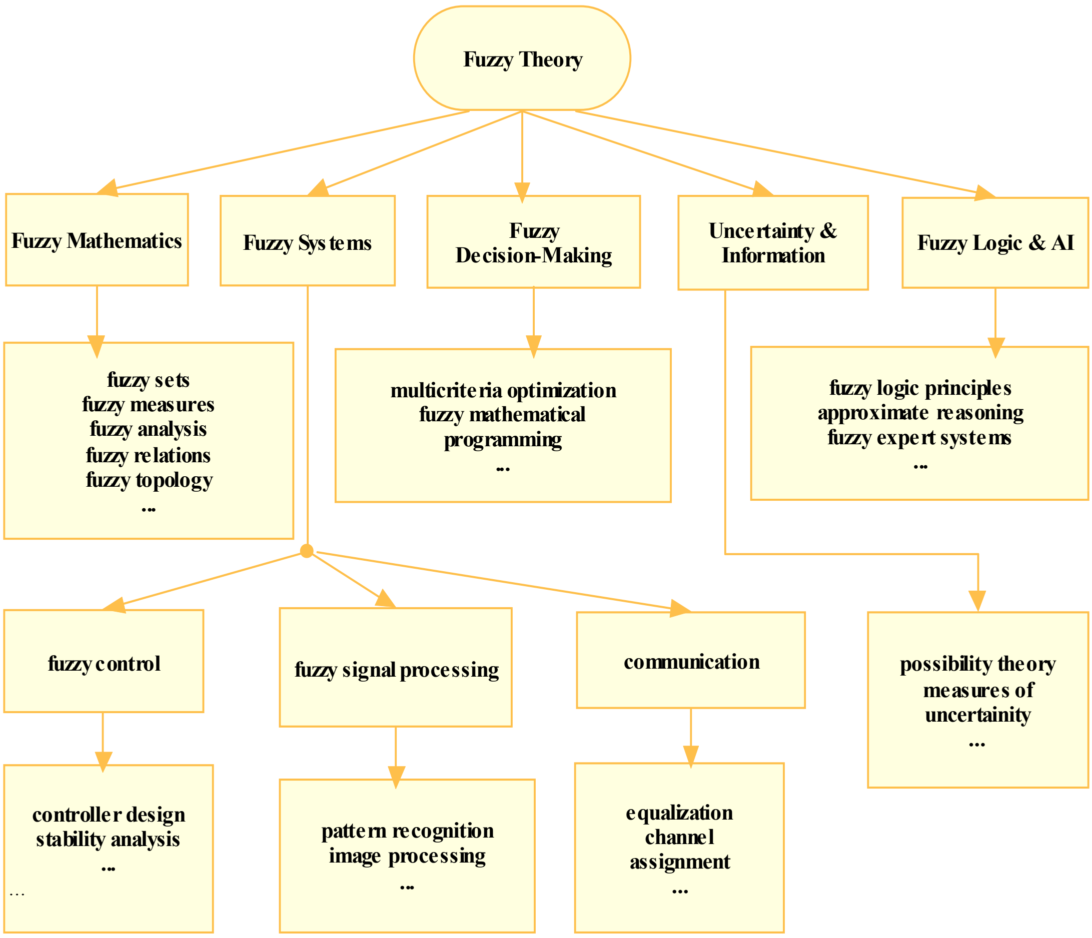

Fig. 1.1 Classification of Fuzzy Theory

1.3.1 The Fuzzy *IF-THEN* Rule

The Fuzzy *IF-THEN* rule is the essential concept of fuzzy systems, as it is structured on which factors (premise part - *If*) lead to which actions (consequent part - *Then*) within the fuzzy system. For example, for the purpose of safety, if the fire sensor detects the density of the smoke is higher than that normally present in the air, then the fire alarm works immediately. Meanwhile, the fuzzy *IF-THEN* rule of this example is described in the following form:

IF the density of the smoke is higher than 20%, THEN the fire alarm works immediately

To carry the fuzzy *IF-THEN* rule out, the mechanism of the fuzzy system emphasises processing a factor into an action. The basic configuration of a fuzzy system can be described as in Fig. 1.2 (Wang, 1997).

1.3.2 Basic Configuration of Fuzzy System

A basic configuration of a fuzzy system includes a fuzzifier, defuzzifier, fuzzy rule base and fuzzy inference engine, and each component plays a role in the overall mechanism of the fuzzy system. As shown in Fig. 1.2, the mechanism of a fuzzy system consists of three stages. In the first stage, a fuzzifier transforms a real-value variable (x in U) into a fuzzy set according to a degree of compatibility of the respective fuzzy sets, and this procedure can be called fuzzification. In the second stage, the fuzzy system processes fuzzy rules using a fuzzy inference engine. In the third stage, the resultant fuzzy values are transformed again into a real-value variable (y in V) by a defuzzifier, this procedure can be called defuzzification. The detailed configuration of a normal fuzzy model will be introduced in chapter 2.

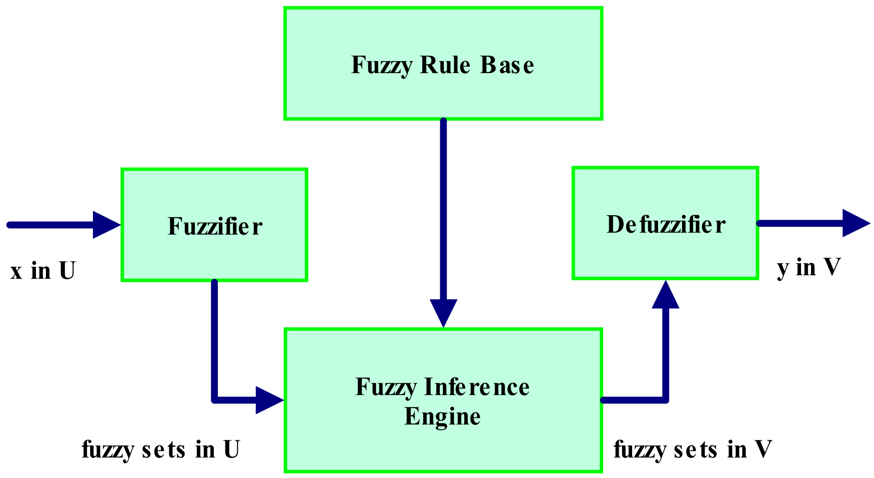

Fig. 1.2 Basic configuration of fuzzy systems

1.3.3 The Advantage of Fuzzy Systems

An important contribution of fuzzy systems theory is that it provides a systematic procedure for transforming a knowledge base into a nonlinear mapping (Wang, 1997).

The main advantage of fuzzy systems is their well-interpretability. This well-interpretability is the unique merit of fuzzy systems and results from the good knowledge representation of fuzzy systems, which provides a reasonable explanation for a particular value appearing as the output from the fuzzy system. Due to the simplicity of the fuzzy *IF-THEN* rules, not only can the definition of the problem be represented by linguistic terms, but it is also not difficult to understand the detailed information represented by fuzzy membership functions within fuzzy systems. The advantages of fuzzy systems were summarised by (Vieira, 2004):

- ❖ Capacity to represent inherent uncertainties of the human knowledge with linguistic variables.
- ❖ Simple interaction of the expert of the domain with the engineer designer of the system.
- ❖ Easy interpretation of the results, because of the natural rules representation.
- ❖ Easy extension of the base of knowledge through the addition of new rules.
- ❖ Robustness in relation of the possible disturbances in the system.

1.3.4 The Disadvantage of Fuzzy Systems

The main disadvantage of fuzzy systems is that the fuzzy rule base has been fixed and this means that the fuzzy system cannot adapt to changing situations flexibly. Another disadvantage of fuzzy systems is that in order to define the fuzzy rules, an expert's knowledge or instructions are required in advance. Also, the process of tuning the parameters of fuzzy systems often requires a relatively long period of time, especially if there are a large number of fuzzy rules within the fuzzy system. Another minor disadvantage of fuzzy systems is their lack of universality, as it is not possible for an end user using a fuzzy system to solve a wide range of different tasks, due to the complexities of the fuzzy systems. The disadvantages of the fuzzy systems were summarised by (Vieira, 2004):

- ❖ Incapable to generalise, or either, it only answers to what is written in its rule base.
- ❖ Not robust in relation the topological changes of the system, such changes would demand alterations in the rule base.
- ❖ Depends on the existence of an expert to determine the inference logical rules.

1.4 Problem Statement

Due to the good knowledge representation of fuzzy systems, they have been used to construct fuzzy models to support the solution of both regression-type problems and function approximation problems. This thesis puts focus on dealing with regression-type problems and function approximation problems by learning and identification of fuzzy systems. Basically, function approximation is the process of learning algorithms acquiring only some *approximation* to the target function, and function approximation problems are required to choose a function among a well-defined class that approximates a target function in a task-specific way (Mitchell, 1997). As the need for function approximation arises in many fields of applied mathematics and computer science, how to deal with function approximation problems optimally becomes an increasingly important issue.

Although many efforts have been made to deal with regression and function approximation problems by fuzzy system identification, there are still several challenges which remain. Two of main research topics are discussed in this thesis and are described as follows.

1.4.1 Interpretability vs. Accuracy

As the complexity of a system increases, our ability to make precise and yet significant statement about its behaviour diminished until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics (Zadeh, 1973).

1.4.1.1 Interpretability

Interpretability is the capability to express the behaviour of the real system in an understanding way, and it has been especially considered in the linguistic fuzzy rule base system. The interpretability can also be recognised as compactness, completeness, consistency, or transparency. As the objective of linguistic fuzzy modelling is to acquire fuzzy systems with a good interpretability, the major purpose of the linguistic fuzzy rule-base system aims at providing good interpretability to explain the behaviour of the system in an understanding way. In addition to providing a clear explanation of the system by a good interpretability, a linguistic fuzzy rule-base system with well-interpretability may not result in a complex system, and this may relieve the minor disadvantage of fuzzy systems as mentioned before; "It is not convenient for the end user using fuzzy systems to solve a wide area of different tasks, as a result of the complexity of fuzzy systems."

The interpretability issues of fuzzy systems are a popular topic and these primarily put the focus on approaches to improving the interpretability of fuzzy systems, such as improving the interpretability with flexible rule structures, complexity reduction in linguistic fuzzy models, complexity reduction in precise fuzzy models, interpretability constraints in TSK fuzzy rule-based systems, and assessments of the interpretability (Casillas *et al.*, 2003).

1.4.1.2 Accuracy

Accuracy is the capability to faithfully represent the real system, and it has been especially considered in precise fuzzy systems. As the objective of precise fuzzy modelling is to acquire fuzzy systems with good accuracy, the major purpose of the precise fuzzy system aims at faithfully representing the modelled system. Without considering the problem of overfitting, the higher the accuracy of the model, the closer the model to the system is. In addition to obtaining high levels of accuracy by precise fuzzy modeling, a precise fuzzy system with good accuracy may effectively perform unsupervised learning, where the problem of overfitting is not applicable.

1.4.1.3 Compromise between Interpretability and Accuracy

Solving pattern recognition problems by constructing a large diversity of modular methods based on fuzzy system identification has also been researched. In particular, one of the unique features of fuzzy systems is their well-understanding interpretability. To get a highly estimated accuracy of the approximation and the well-understanding interpretability of the fuzzy system together, more complex methods try to achieve this by reinforcing the strength of existing methods. However, there still exists a conflict between the interpretability and the accuracy of the approximation in general fuzzy systems. If the degree of the interpretability increases, then the accuracy rate of the approximation decreases. In contrast, if the degree of the interpretability decreases, then the accuracy rate of the approximation increases. The study of the best compromise between the accuracy of the approximation and the degree of the interpretability in the fuzzy system is a significant research subject.

1.4.2 Underfitting vs. Overfitting

Underfitting and overfitting are important concepts in machine learning, and examine the performance of the learner or learning algorithm. In supervised learning, the learner or learning algorithm has been trained using training examples, the reliability of which can then be examined by validating the test examples.

1.4.2.1 Underfitting

So far as underfitting is concerned, if the training accuracy is poor, then underfitting may occur. Underfitting generally occurs when a model is too simple, has insufficient parameters, or an improper mechanism. Underfitting results in poor predictive performance and a bad generalisation ability.

1.4.2.2 Overfitting

Overfitting occurs when the validation errors increase but the training errors steadily decrease. In other words, the situation of overfitting occurs when there is high training accuracy but poor validation accuracy. Overfitting generally occurs when a model is too complex, with too much parameters, or an overlearning mechanism. Overfitting results in a bad generalisation ability, acquiring good accuracy whilst using training examples trains the model but it then fail when testing the examples so invalidating the model, as illustrated in Fig. 1.3.

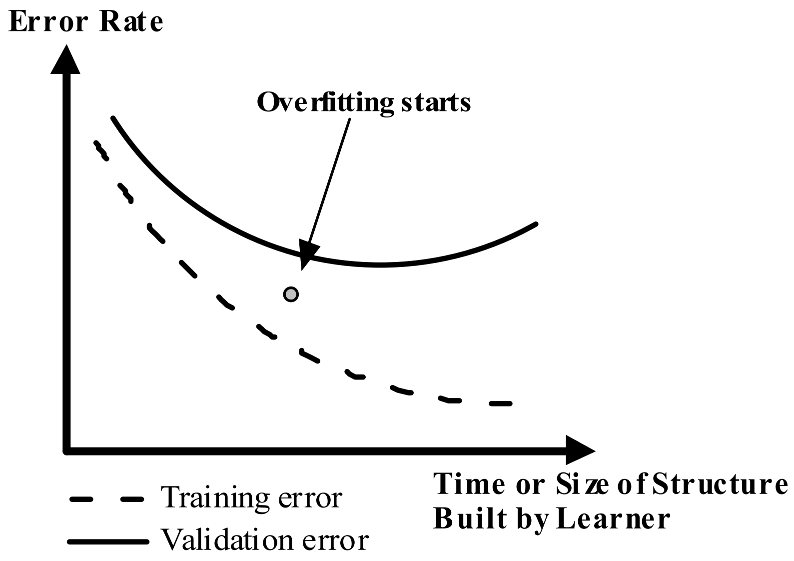

Fig. 1.3 Overfitting relations with training error and validation error

However, avoiding the problem of underfitting is easier than avoiding that of overfitting. It is not difficult to construct a complex model with many parameters based on training examples only. It is a difficult issue to develop a learner or learning algorithm with a good predictive performance as well as a sound generalisation ability. Therefore developing a learner or learning algorithm which has a good predictive performance as well as a sound generalisation ability remains a research challenge.

1.4.3 The Problem of "Curse of Dimensionality"

"Curse of dimensionality" is one of the most serious weaknesses of fuzzy systems, and it is a general problem for fuzzy systems to deal with all high-dimensional approximation problems. It usually occurs in designing precise fuzzy systems with a required accuracy, because the number of fuzzy rules has to increase exponentially with the number of input variables to fuzzy systems. Assume that there are *n* input variables and *m* fuzzy rules are defined for each input variable, then the number of fuzzy rules in fuzzy systems is m^n (Wang, 1997). Clustering algorithms are one of good approaches to overcome this problem, but a grid-based clustering method may result in the problem of "curse of dimensionality" as well.

1.4.4 The Cluster Analysis Issue

The predetermined clustering technique before constructing an initial fuzzy model is another key issue between the interpretability and accuracy of the overall system. Though clustering algorithms have been always been oriented to solve classification and pattern recognition problems, some methods solve function approximation problems by constructing initial models for function approximation (Gonzalez, 2002). A simple clustering technique may lead to good interpretability but with low accuracy, and a complex clustering technique may acquire a good accuracy but poor interpretability. Since there is a paradox between the accuracy of the approximation and the degree of the interpretability in fuzzy systems, the balance between interpretability and accuracy is an interesting problem. The application of clustering techniques to increase the interpretability, accuracy or both, and used as a foundation to capture useful knowledge is another derivation problem resulting from the issue of interpretability and accuracy.

1.4.5 Summary

There are a lot of challenges to deal with regression and function approximation problems by learning and identification of fuzzy systems. For these challenges, two works are proposed in this thesis to overcome these challenges as far as possible. One is three-part input-output clustering-based approach for fuzzy system identification, and another one is similarity-base learning algorithm for fuzzy system identification with a two-layer optimisation scheme. So far as the first work is concerned, the problem of underfitting can be avoided by identifying the proper number of clusters and appropriate location of clusters, because which can make fuzzy systems more complete. So far as the second work is concerned, the well-interpretability can be achieved by the similarity-based pruning strategy, because which can make fuzzy systems more transparent as well as compact. Also, a required accuracy of fuzzy systems can be obtained by two-layer optimization scheme, and the problem of underfitting and that of overfitting can be avoided by two-layer optimization scheme, which make fuzzy systems more precise.

1.5 Research Objectives and Outline

Before describing the research of this thesis, an understanding of the research objectives and outline are required. The motivation behind this thesis and the objectives of the research work, the research outline of the proposed approaches are described as follows.

1.5.1 Research Objectives

The research work mainly aims at dealing with a variety of regression-type problems and function approximation problems by learning and identification of fuzzy systems, and the development of fuzzy system identification is constructed using Takagi-Sugeno fuzzy models. Through modelling fuzzy system identification, a variety of regression-type problems and function approximation problems, such as static function approximation problems, time-series prediction problems, as well as nonlinear dynamic system identification problems, are expected to work smoothly. The research consists of two research topics presented consecutively and include a three-part input-output clustering-based approach for fuzzy system identification and a similarity-based learning algorithm for fuzzy system identification with a two-layer optimisation scheme. The second research topic is based on the findings of the first topic.

1.5.2 Research Outline

The research consists of two topics presented consecutively in this thesis. The first, proposes a three-part input-output clustering approach for fuzzy system identification. The focus of this first topic is on constructing a well-defined fuzzy model by means of the clustering technique. The second research topic can be regarded as an extension of the first. In the second research topic, a similarity-based learning algorithm for fuzzy system identification with a two-layer optimisation scheme has been developed based on the results of the first research topic. The focus is on constructing either a well-defined fuzzy model or a highly-reliable fuzzy model by means of rule base simplification strategy and parameters optimisation scheme. The research outlined in this thesis is briefly illustrated in Fig. 1.4, and the complete outline of research is illustrated in Fig. 1.5.

1.5.3 A Three-Part Input-Output Clustering-Based Approach for Fuzzy System Identification

The first area of research is concerned with the clustering technique for constructing fuzzy models in fuzzy system identification. As the determination of the proper number of clusters and the appropriate location of clusters is one of the primary considerations on constructing an effective fuzzy model, the task of the clustering technique is to recognise the proper number of clusters and their appropriate location. This is good preparation for the construction of initial fuzzy models. In order to acquire a mutually exclusive performance by constructing an effective fuzzy model, a modular method of fuzzy system identification, based on a hybrid clustering-based technique has been considered. Due to the above reasons, a hybrid clustering algorithm concerning the input, output, generalisation and specialisation has hence been used in this work. Moreover, the research outlined of this work is illustrated in Fig. 3.1.

1.5.4 A Similarity-Based Learning Algorithm for Fuzzy System Identification with a Two-Layer Optimisation Scheme

The second area of research is an evolution of the earlier research, and two ways to improve the original work were considered, including the pruning strategy for the premise structure of the fuzzy system and the optimisation scheme for the initial fuzzy model. So far as the pruning strategy is concerned, the purpose of this is to refine the rule base by similarity analysis of fuzzy sets, fuzzy numbers, fuzzy membership functions or fuzzy rules. In other words, through similarity analysis, the complete rules can be kept and the redundant rules can be reduced within the rule base of the fuzzy system. The optimisation scheme can be regarded as a two-layer parameter optimisation, because the parameters of the initial fuzzy model have been fine tuned by a two phase gradation. Hence, in this second area of research, the focus is on enhancing the performance of the initial fuzzy models towards a positive reliability in the final fuzzy models. Moreover, the research outlined of this work is illustrated in Fig. 4.1.

1.5.5 The Main Contribution of this Thesis

This research puts focus on dealing with regression-type problems and function approximation problems by considering on learning and identification of fuzzy systems. The idea of the contributions of this thesis is originated from Occam's Razor, because the primary research of this thesis can achieve a reliable performance on dealing with regression-type problems and function approximation problems by learning and identification of fuzzy systems without much complicated mechanisms and computational resource.

The primary research work can positively discover the proper number of clusters and their appropriate location by integrating a variety of existing clustering properties without much complicated mechanisms. Also, a better local minimum for parameters optimisation can be acquired by an optimisation scheme without much complicated mechanisms and computational resource as well. Additionally, the primary research of this thesis can avoid the problems of underfitting and those of overfitting.. Hence, “Make everything as simple as possible, but not simpler.” is the primary contribution of learning and identification of fuzzy systems proposed in this thesis.

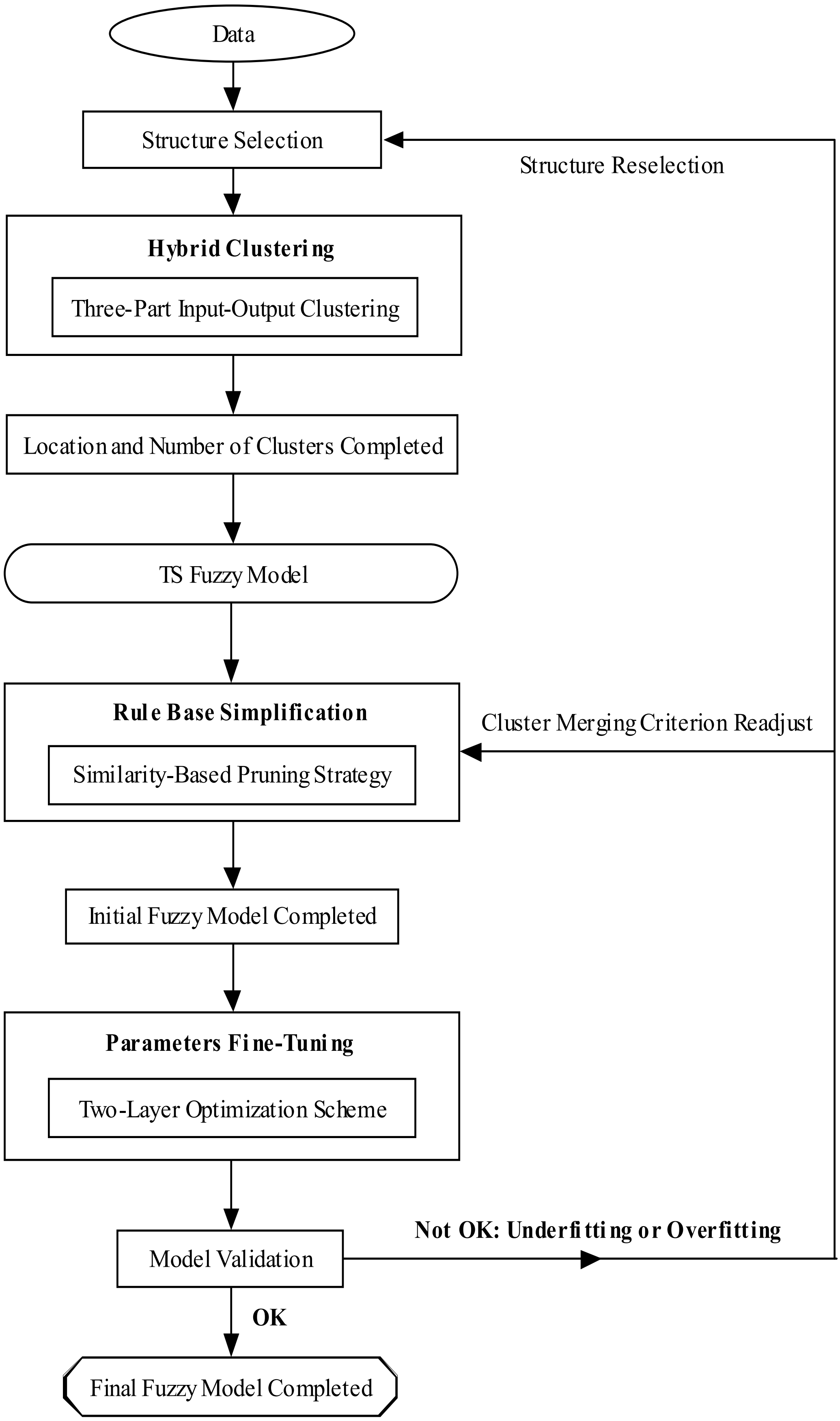

Fig. 1.4 Brief structure of research outline

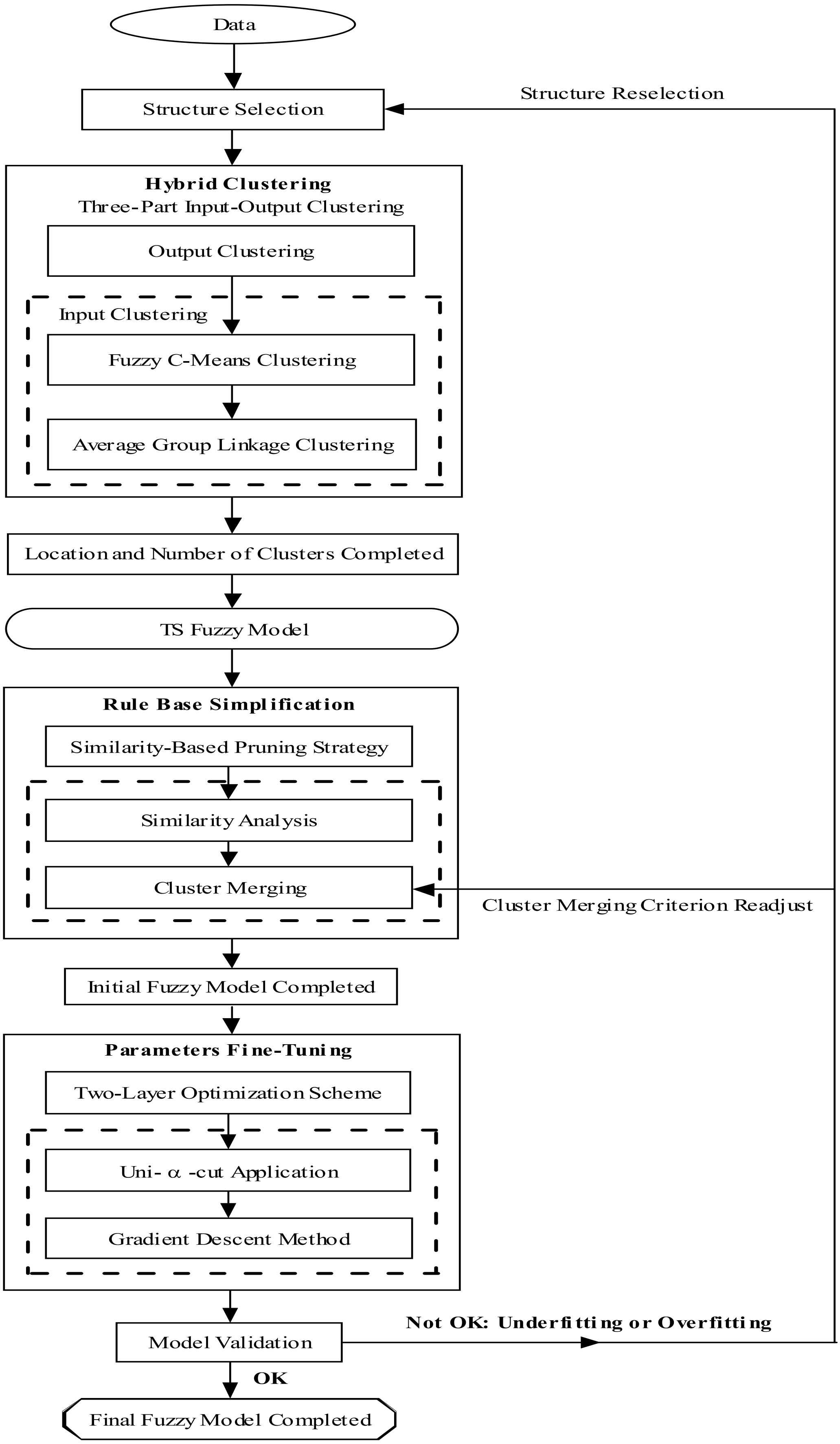

Fig. 1.5 Complete structure of research outline

1.6 Thesis Structure

For clarity, the rest of this thesis is organised as follows:

In Chapter 2, there is a comprehensive introduction to the background of fuzzy system identification and learning and identification of fuzzy systems are reviewed. Eight main themes are examined: fuzzy sets and membership functions, fuzzification and defuzzification, fuzzy system identification, literature reviews of learning and identification of fuzzy systems, cluster analysis, similarity-based pruning strategy, parameters optimisation, and relational database estimation.

In Chapter 3, a three-part input-output clustering-based approach for fuzzy system identification is proposed for the purpose of constructing a sound fuzzy model by discovering the proper number of clusters and their appropriate location.

In Chapter 4, a similarity-based learning algorithm for fuzzy system identification with a two-layer optimisation scheme is proposed for refining the rule base of a fuzzy system and is used to generate a reliable final fuzzy model.

In Chapter 5, the work is concluded and the contributions summarised, and finally future research which could advance this area is put forward.

Chapter 2 RELATED BACKGROUND AND LITERATURE REVIEW

In this chapter, a comprehensive introduction to the relevant background and a review of research will be described by following the essential steps involved, including; fuzzy sets and membership functions, fuzzification and defuzzification, fuzzy system identification, literature reviews of learning and identification of fuzzy systems, cluster analysis, similarity-based pruning strategy, and parameters optimisation. For each, the literature is reviewed and some existing methods are discussed in detail.

2.1 Fuzzy Sets and Membership Functions

A good representation of knowledge is a unique feature of fuzzy systems, and the concepts of fuzzy sets as well as membership functions are essential components in understanding fuzzy systems. Due to the interpretability of fuzzy sets and membership functions, it is not difficult to understand the detailed information that appears in fuzzy systems.

2.1.1 Fuzzy Sets

A fuzzy set is different from a crisp set which has a crisp boundary, because a fuzzy set is a set with a fuzzy degree. A fuzzy degree is different to a physical possibility, in that the fuzzy degree represents the percentage of the fuzzy set that possesses an action or event. Basically, each fuzzy set is associated with a membership value. Let U be the universe of discourse, and $\mu_A(x)$ be the membership function of the fuzzy set A. Therefore, the definition of a fuzzy set A in U can be represented as a set of ordered pairs of a generic element x and its membership value by (Wang, 1997):

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad \textbf{(2.1)}$$

2.1.2 The Concept of α-Cuts of Fuzzy Sets

An α-cut of a fuzzy set *A* is a crisp set that A_α contains all the elements in the universe of discourse *U* that have membership values in *A* greater than or equal to *α*, and it can be defined as follows (Wang, 1997):

$$A_\alpha = \{x \mid \mu_A(x) \geq \alpha,\ x \in U\} \quad \textbf{(2.2)}$$

where $\alpha \in (0, 1]$. According to the equivalence relation of α, α-cut of fuzzy sets can be classified into two types, including α-cut and strong α-cut. The difference between general α-cut and strong α-cut is that the membership grade of the general α-cut is bigger than or equal to the value of α, but the membership grade of the strong α-cut is only bigger than the value of α. Therefore, a general α-cut of a fuzzy set can be defined by equation 2.2, and a strong α-cut of a fuzzy set can be defined by:

$$A_\alpha = \{x \mid \mu_A(x) > \alpha, x \in U\} \quad \textbf{(2.3)}$$

2.1.3 Fuzzy Membership Functions

In the procedure of fuzzification, each element or crisp set are converted to a fuzzy set by mapping fuzzy membership functions in the fuzzy system. So far as the type of fuzzy membership functions are concerned, there are two types of membership functions, including continuous membership function and discrete membership function. The differentiation between these two types of membership functions depends on the universe of discourse *U* that it belongs to. If the universe of discourse *U* is continuous, the type of the membership function would be the continuous membership function. In contrast, if the type of the universe of discourse U is discrete, then the type of the membership function would be the discrete membership function. Hence, the definition of the continuous membership function is represented as follows (Wang, 1997):

$$A = \int_U \mu_A(x)/x \qquad \textbf{(2.4)}$$

The definition of the discrete membership function is represented as follows (Wang, 1997):

$$A = \sum_U \mu_A(x)/x \qquad \textbf{(2.5)}$$

Another significant definition of fuzzy membership function is the complement of fuzzy membership function (Wang, 1997). In equation 2.9, $\mu_{\bar{A}}(x)$ is the complement of $\mu_A(x)$. In other words, $\bar{A}$ is the complement of the fuzzy set A, and the sum of the value of A and $\bar{A}$ is 1.0.

$$\mu_A(x) + \mu_{\bar{A}}(x) = 1 \qquad \textbf{(2.6)}$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \qquad \textbf{(2.7)}$$

The relationship between A and $\bar{A}$ in the fuzzy membership function is illustrated in Fig. 2.1 (Wang, 1997). In Fig. 2.1, the definition of the relationship is $A + \bar{A}$ =1.0 absolutely. On the other hand, when $\bar{A}$ increases, then A decreases with the synchronisation. The sum of the value of A and $\bar{A}$ is 1.0. However, there exist certain exceptions when the membership functions irregularly overlap each other simultaneously. In order to understand how to deal with this kind of situation, the concept of fuzzy normalisation will be briefly introduced.

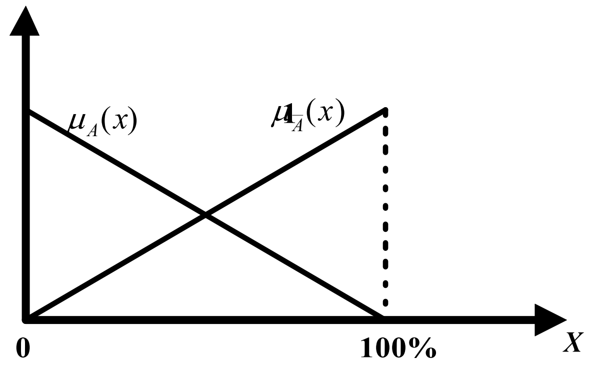

Fig. 2.1 The relationship between A ($\mu_A(x)$) and Ā ($\mu_{\bar{A}}(x)$)

2.1.4 Shape of Fuzzy Membership Functions

According to the shape of fuzzy membership functions, there are about eleven shapes of fuzzy membership functions, and three of these are in common use, including the triangular membership function, the trapezoidal membership function, and the Gaussian membership function. To understand the mechanism of these three shapes of fuzzy membership functions, they are discussed below.

2.1.4.1 Triangular Membership Function

Generally, there are three scalar parameters to construct a triangular membership function, which consists of *left vertex*, *centroid*, and *right vertex* of a triangular shape. Generally, the triplet (*left vertex, centroid, right vertex*) of the triangular membership function can be defined as (a,b,c) as in Fig. 2.2. Fig. 2.2 can be regarded as either symmetric type or asymmetric type, and the calculation of triplet (a,b,c) of the triangular membership function can be represented by:

$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \le x \le b, \\ \dfrac{c-x}{c-b}, & b \le x \le c, \\ 0, & otherwise. \end{cases} \qquad \textbf{(2.8)}$$

where $\mu_A(x)$ presents the membership of x, a, b, and c presents the *left vertex*, *centroid*, and *right vertex* of a triangular membership function, respectively.

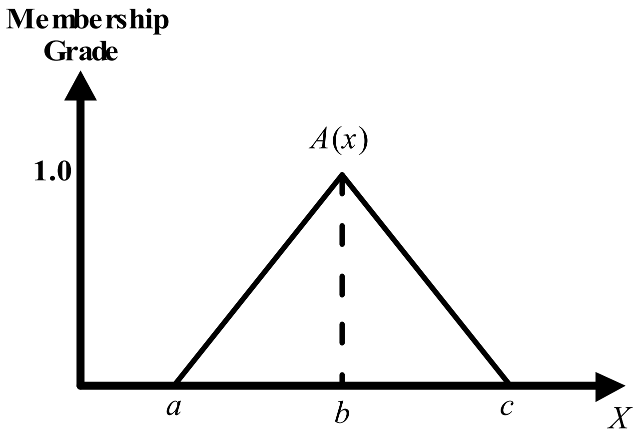

Fig. 2.2 Triangular Membership Function

Furthermore, there is a strict definition to generate the triangular membership function by distinguishing the triangular membership function into symmetric type and asymmetric type. As illustrated in Fig. 2.3, the symmetric triangular membership function for the fuzzy system based on the triplet $\left(x^{\min}{}_i, centroid_i, x^{\max}{}_i\right)$ can be calculated as follows:

$$\mu_i = \prod_{j=1}^{m} A_{ij}\left(x_j\right) \tag{2.9}$$

$$\begin{aligned} &\text{If } \left|x_j - a_{ij}\right| \le b_{ij}/2\,,\ \text{then } A_{ij}(x_j) = 1 - \frac{2 \cdot \left|x_j - a_{ij}\right|}{b_{ij}} \\ &\text{else},\ 0 \end{aligned} \tag{2.10}$$

$$a_{ij} = \frac{x^{\min}{}_i + x^{\max}{}_i}{2} \tag{2.11}$$

$$b_{ij} = \left|x_i^{\max} - x_i^{\min}\right| \tag{2.12}$$

$$y_j = \frac{\mu_i \cdot w_i}{\sum_{i=1}^{n} \mu_i} \quad \textbf{(2.13)}$$

where μ_i is a membership value of the antecedent part, $x_i^{\min}$ is the value of the minimum unit or datum of each partition, $x_i^{\max}$ is the value of the maximum unit or datum of each partition, a_{ij} is the centroid of the triangular membership function, b_{ij} is the width of the triangular membership function, y_i is the system output, and w_i is the weight associated with each rule.

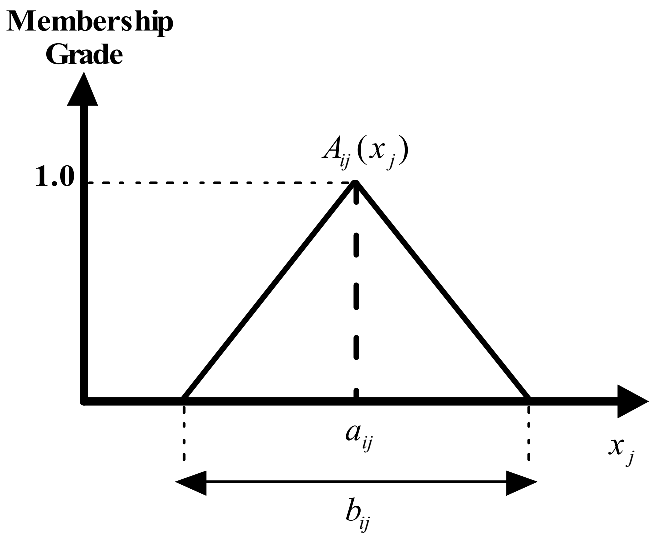

Fig. 2.3 Symmetric triangular membership function

As illustrated in Fig. 2.4, in the asymmetric triangular membership function, there are two widths b_{ij} and c_{ij} to be calculated, and which can be represented by:

$$a_{ij} - b_{ij} \le x_j \le a_{ij} : \quad A_{ij}(x_j) = 1 + \left(x_j - a_{ij}\right) / b_{ij} \quad \textbf{(2.14)}$$

$$a_{ij} \le x_j \le a_{ij} + c_{ij}: \quad A_{ij}(x_j) = 1 - {(x_j - a_{ij})} \big/ {c_{ij}} \tag{2.15}$$

$$others, \; 0$$

where b_{ij} is the antecedent-part width, and c_{ij} is the consequent-part width.

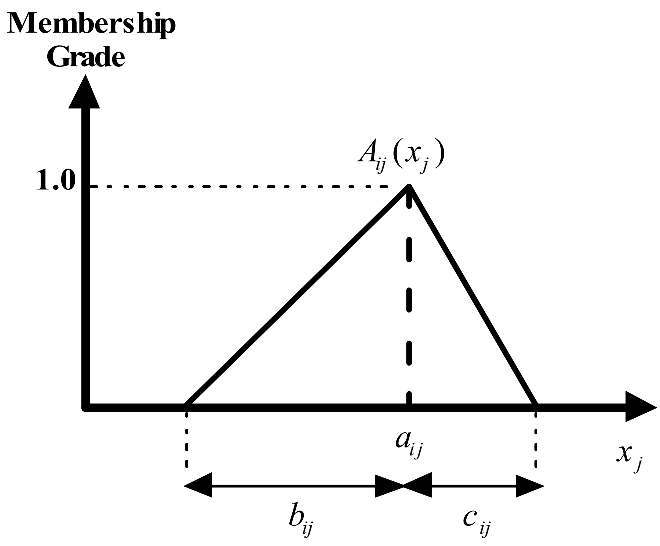

Fig. 2.4 Asymmetric triangular membership function

2.1.4.2 Trapezoid Membership Function

In the trapezoidal membership function, there are four scalar parameters to construct a triangular membership function, which consists of *left vertex*, *upper length starting point, upper length terminal point*, and *right vertex* of a trapezoidal shape. Generally, the quaternion (*left vertex*, *upper length starting point, upper length terminal point*, *right vertex*) of the trapezoidal membership function can be defined as (a,b,c,d) as in Fig. 2.5, and the calculation of quaternion (a,b,c,d) of the trapezoidal membership function can be represented by:

$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \le x \le b, \\ 1, & b \le x \le c, \\ \dfrac{d-x}{d-c}, & c \le x \le d, \\ 0, & \textit{otherwise.} \end{cases} \quad \textbf{(2.16)}$$

where $\mu_A(x)$ represents the membership of x, a, b, c, and d represents the *left vertex*, *upper length starting point, upper length terminal point*, and *right vertex* of a trapezoidal of a trapezoidal membership function, respectively.

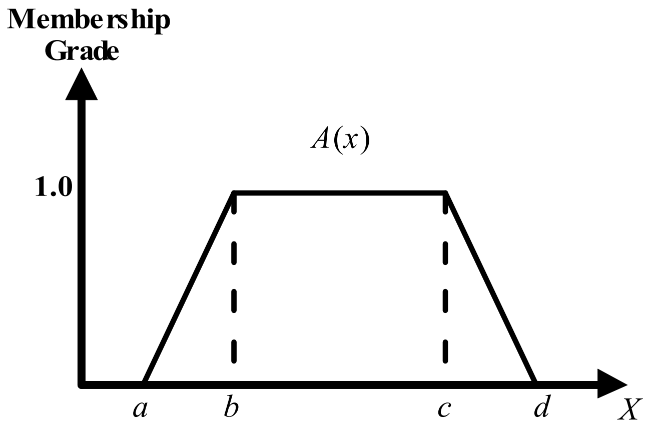

Fig. 2.5 Trapezoidal Membership Function

2.1.4.3 Gaussian Membership Function

In a Gaussian membership function, there are two parameters, which consist of the centre and the width of the Gaussian membership function. Generally, the centre and the width of Gaussian membership function can be defined as (v, σ) as in Fig. 2.6, and the calculation of the twosome (v, σ) of a symmetric Gaussian function can be represented by:

$$\mu_A(x) = \exp\left(\frac{-(x-v)^2}{2\sigma^2} \right) \quad \textbf{(2.17)}$$

where $\mu_A(x)$ represents the membership of v and σ represents the centre and width of the Gaussian membership function, respectively.

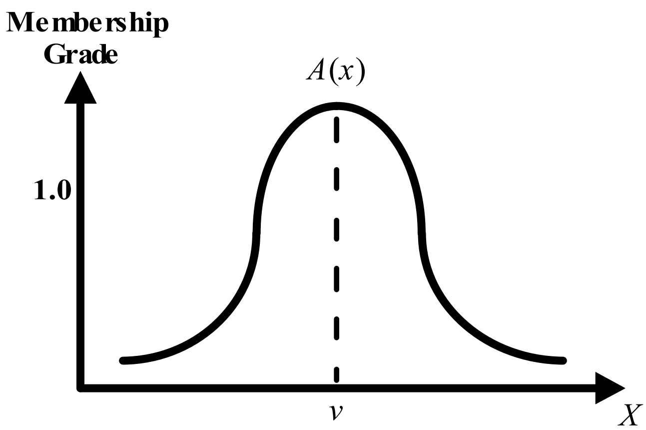

Fig. 2.6 Gaussian Membership Function

2.2 Fuzzification and Defuzzification

Fuzzification and defuzzification are important components relating to the fuzzifier and the defuzzifier in fuzzy models, which take the responsibility of the front door and rear door in fuzzy systems respectively. In order to understand the concept of fuzzification and defuzzification, they will be described below.

2.2.1 Fuzzification

As mentioned before, fuzzification is the process of converting crisp inputs or numeric inputs into a fuzzy set using fuzzy membership functions stored in a fuzzy rule base. The crisp values of $\wp_k(t)$, $k = 1, 2, ..., r.$ are transformed into fuzzy set A_k^i with membership function $\mu_{A_k^i}(u_k) \in U_k$.

Moreover, the fuzzification of a singleton fuzzifier can be called singleton fuzzifiction, and any fuzzy set produced by singleton fuzzification is called a singleton. The advantage of singleton fuzzification is that lots of computational costs and time can be saved, but contrarily the disadvantage is that the recognition performance is not good enough as a result of insufficient diversity of the fuzzy sets. A fuzzy set A_k^i with a membership function $\mu_{A_k^i}(u_k) \in U_k$ can be defined by:

$$\mu_{A_k^i}(u_k) = \begin{Bmatrix} 1, & u_k = \wp_k(t) \\ 0, & otherwise \end{Bmatrix} \tag{2.18}$$

2.2.2 Defuzzification

As mentioned before, defuzzification is the process of converting output fuzzy sets to crisp values $\hat{y} \in R$. Before the crisp value has been produced by defuzzification, all of the fuzzy outputs are aggregated with a union operator. Therefore, the overall output can be obtained as the union of fuzzy outputs by $\mu_A = \bigcup_i (\mu_i(x))$. So far as the methods for defuzzification are concerned, there is a variety of defuzzification methods which have been developed recently based on different mechanisms. Below three popular defuzzification methods are introduced, including the centre of gravity method, the mean of maximum method, and the weighted average method.

2.2.2.1 Centre of Area Defuzzification Method

The centre of area method (COA) is the most applied method of weighted average in defuzzification, and the COA method also can be called the centre of gravity method (COG) or centroid method. The main feature of the COA method is that it considers the entire possibility of membership functions to calculate the defuzzified values. Due to this feature, lots of computational costs may be incurred during the procedure of defuzzification, especially for the continuous domain. According to the type of fuzzy set, the equation of the method can be represented by a continuous type or discrete type. If the membership function is defined as a continuous type, the calculation of defuzzification can be represented by:

$$\widehat{y}_{COA} = \frac{\int_Y \mu(y) \cdot y dy}{\int_Y \mu(y) \cdot dy} \quad \textbf{(2.19)}$$

where $\int$ is algebraic integration, $\widehat{y}_{COA}$ is the defuzzified output by the COA method, y is the output variable, and $\mu(y)$ is the aggregated membership function. However, if the output fuzzy set is defined as a discrete type, the calculation of defuzzification can be represented by:

$$\widehat{y}_{COA} = \frac{\sum_{i=1}^{k} y_i \cdot \mu(y_i)}{\sum_{i=1}^{k} \mu(y_i)} \quad \textbf{(2.20)}$$

where $\widehat{y}_{COA}$ is the defuzzified output, y_i is the output variable, and $\mu_i(y)$ is the aggregated membership function.

2.2.2.2 Mean of Maximum Method

The mean of maximum (MOM) method is usually used for discrete fuzzy sets, and a crisp output $\widehat{y}$ in this method is chosen to represent the mean value of all elements whose membership in K is the maximum. In other words, the defuzzified result represents the mean value of all elements whose membership functions reach the maximum. The calculation of defuzzification can be represented by:

$$\widehat{y}_{MOM} = \sum_{i=1}^{K} \frac{y_i}{K} \quad \textbf{(2.21)}$$

where $\widehat{y}_{MOM}$ is the defuzzified output by the MOM method, y_i is the element whose membership function reaches the maximum, and K is the cardinality of y_i

2.2.2.3 Weighted Average Method

In the weighted average method, the defuzzified output is obtained by weighting each membership function in the output by its respective maximum membership value. The weighted average method is generally valid for symmetric output membership functions. Though the representation of formula of the weighted average method is very similar with that of the centre of area method, the weighted average method is less computationally intensive. The calculation of the weighted average method can be represented by:

$$\hat{y}_{WA} = \frac{\sum \mu(y) * y}{\sum \mu(y)} \quad \textbf{(2.22)}$$

where $\hat{y}_{WA}$ is the defuzzified output by the weighted average method, $\mu(y)$ is the output membership function of each rule, and y is the weight associated with each rule.

2.2.2.4 Summary

As a whole, there is no systematic procedure or absolute criterion for choosing a good defuzzification scheme, because the selection of defuzzification methods most depends on the properties of the models or problem-oriented. So far as a sound defuzzification scheme is concerned, a good quality defuzzification approach should include *continuity*, *non-ambiguity*, *plausibility*, and *computational simplicity* (Wang, 1997).

- *Continuity*, a small change in the input should not result in a large change in the output.
- *Non-Ambiguity*, defuzzification methods should always result in a unique value without any ambiguity.
- *Plausibility*, the defuzzified value should represent the output of fuzzy systems from an intuitive point of view.
- *Computational Simplicity*, less computational costs is important for the procedure of defuzzification, especially for a real-time system.

2.3 Fuzzy System Identification

When fuzzy systems are used as a class of models for system identification, the process is called "fuzzy system identification" (Takagi & Sugeno, 1985)

Since *Zadeh* proposed the concept of fuzzy sets in 1965, fuzzy theory has hence been structured and developed gradually. As a result of the reliable representation capability of fuzzy systems, solving the system identification problem by the development of fuzzy system identification has extensively researched. Briefly, the purpose of fuzzy systems aims at dealing with the problem of system identification, the process of which is called "fuzzy system identification." Further, if fuzzy system identification is modelled as a black box, the structure is entirely estimated from data. However, if fuzzy system identification is modelled as a grey box way, the structure is provided by experts.

Takagi and Sugeno (1985) have proposed a rudimentary approach and relevant applications to fuzzy system identification. With the development of fuzzy system identification, lots of relevant methods have been proposed to solve identification problems. Generally, there are four types of major approaches including inductive learning approaches and neural-network-based approaches.

Inductive learning approaches can be reviewed in the work of Wang and Mendel (1992), Delgado and Gonzalez (1993), Hall and Lande (1996) and Hong and Lee (1996), where fuzzy rules from training examples based on inductive learning were generated. So far as neural-network-based approaches are concerned, the approaches can be classified into three categories (Ishibuchi, 1996); fuzzy-rule-based systems with learning ability, fuzzy systems represented by network architectures, and neural networks for fuzzy reasoning. Also, in the category of neural-network-based approaches, several papers can be reviewed on trainable fuzzy systems (Nomura *et al*., 1992), fuzzy adaptive learning control networks (Lin *et al*., 1995), and neural network driven fuzzy reasoning with learning function (Hayashi *et al*., 1992).

2.3.1 Fuzzy Models

Fuzzy logic provides actions or decisions by simulating human thinking with a fuzzy degree, the feature of which is that each action is not absolutely positive or negative. The process of a system modelled by fuzzy logic is fuzzy modelling. Fuzzy models are mathematical models based on fuzzy logic and which has been developed and proposed for a variety of purposes (Mamdani, 1974; Sugeno, 1985; Lee-1, 1990; Lee-2, 1990; Aoki *et al*., 1990; Tong *et al*., 1980; Miyamoto *et al*., 1986; Gupta *et al*., 1986). The configuration of a normal fuzzy model involves six components (Hellendoorm & Driankov, 1997), includes *scaling*, *regressor generator*, *linguistic database*, *fuzzifier*, *fuzzy inference engine*, and *defuzzifier*.

- *Scaling* can be regarded as a component to simplify the data $z(t)$ in magnitude to $\wp(t)$, and that $\hat{y}_s(t \mid \theta) = \hat{y}(t \mid \theta)$
- *Regressor generator*, the task of which is to provide a static map from $\wp(t) \in R^r$ to $\hat{y}(t \mid \theta) \in R$

- *Linguistic database* is the heart of a fuzzy model, and a collection of fuzzy *IF-THEN* rules are stored in a fuzzy rule base.
- *Fuzzifier*, maps the crisp values of $\wp(t)$ into a fuzzy set.
- *Fuzzy inference engine*, produces fuzzy sets in the output *y*, the fuzzy set is interpreted by the fuzzy inference engine mapping fuzzy rules in the fuzzy rule base.
- *Defuzzifier* is the last step of the procedure, which converts the output fuzzy sets to a crisp value $\hat{y}(t \mid \theta) \in R$.

The structure of a fuzzy model (Hellendoorm & Driankov, 1997) can be illustrated as in Fig. 2.7, where the thin arrows indicate the computational flow and the thick arrows indicate the information flow.

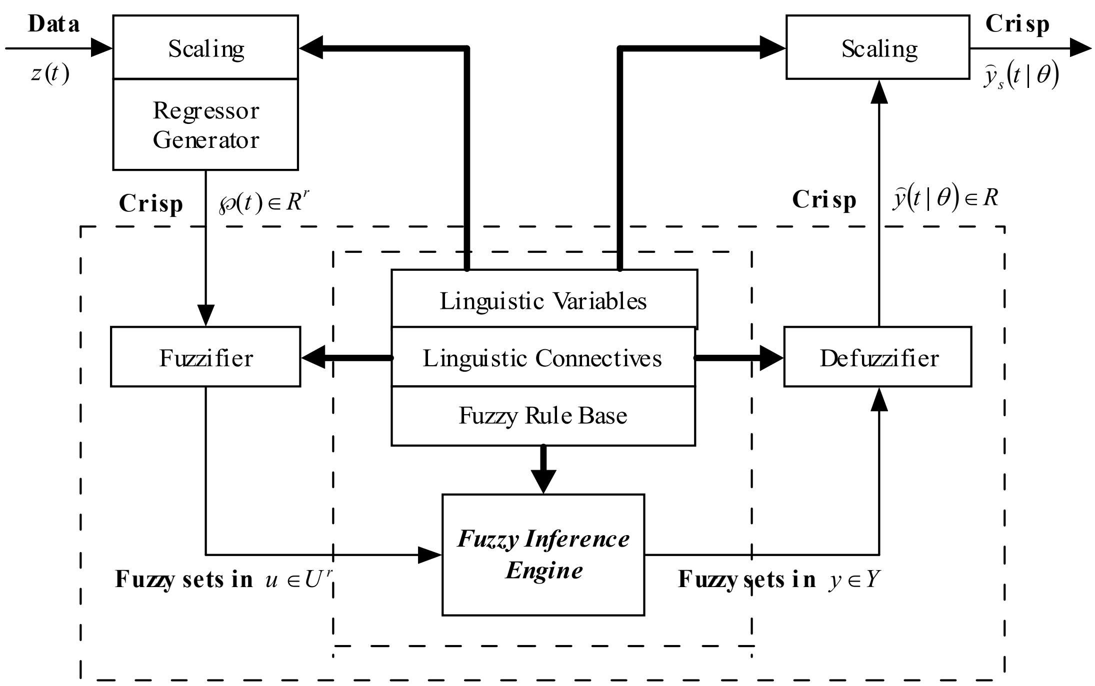

Fig. 2.7 The procedure of fuzzy system identification

So far as the type of modelling is concerned, the fuzzy system can be modelled either by linguistic rules or by a relation matrix. A fuzzy system modelled by linguistic rules, is represented as a rule-based form and is called a rule-based fuzzy model (Sugeno, 1985; Sugeno & Kang, 1988). However, a fuzzy system modelled by a relation matrix, is represented as a relation-based form and is called a relation-based fuzzy model (Pedrycz, 1983; Pedrycz, 1984;.Graham & Newell, 1988).

2.3.1.1 Rule-Based Fuzzy Model

In rule-based fuzzy models, the structure and parameters are identified by modelling. A rule-based fuzzy model consists of two parts, the structure identification and the parameter identification (Sugeno & Kang, 1988; Takagi & Sugeno, 1985; Pedrycz, 1983; Pedrycz, 1984; Czogola & Pedrycz, 1981; Graham & Newell, 1988). Moreover, according to the essential framework of a fuzzy system, the framework is organised as a premise part and a consequent part. Hence, the structure identification can be decomposed into premise structure identification and consequent structure identification, and the parameter identification can be decomposed into premise parameters identification and consequent parameters identification.

The entire identification of a rule-based fuzzy model consists of the premise structure identification, the consequent structure identification, the premise parameters identification, and the consequent parameters identification. The task of the premise structure identification is to determine the premise structure by finding the most simplified fuzzy subspaces, by the fewest variables with an optimal performance. The task of the premise parameter identification is to adjust the parameters by minimising the error of the criterion $\varepsilon(t \mid \theta) = y(t) - \hat{y}(t \mid \theta)$, where $\varepsilon(t \mid \theta)$ is an error between desired output and the model output, $y(t)$ is desired output, and $\hat{y}(t \mid \theta)$ is the model output. The task of the consequent structure and parameter identification is to determine the consequent structure and calculate the model outputs. Further, a rule-based fuzzy model for identification can be represented by:

$$I_i : \text{IF} \quad x_1 \ \text{is} \ A_1^i, \ x_2 \ \text{is} \ A_2^i, \ldots, x_n \ \text{is} \ A_n^i$$

$$\text{THEN} \quad y_i = c_0^i + c_1^i x_1 + c_2^i x_1 + \ldots + c_n^i x_n \qquad \textbf{(2.23)}$$

where I_i means the *i*-th rule, 1<*i*<*m*, A_j^i is the *j*-th fuzzy variable of the *i*-th rule, x_j means the input variable, 1<*j*<*n*, y_i means the output from the *i*-th rule, and c_j^i means a consequent parameter.

2.3.1.2 Relation-Based Fuzzy Model

In the relation-based fuzzy model, the relation matrix is identified by modelling, and can be regarded as finding the fuzzy relation from the greatest and least boundaries of the relation matrix. A relation-based fuzzy model for identification can be represented by:

$$x = u \circ R \qquad \textbf{(2.24)}$$

$$x = u^1 \circ u^2 \circ \cdots u^N \circ R \qquad \textbf{(2.25)}$$

where $\circ$ is the composition operator, and u^N is the universe of discourse, and *R* the relation.

2.3.2 Mamdani-Type Fuzzy Model

According to the inference type of the consequent part of the fuzzy model, there are generally two types of fuzzy models. One is the Mamdani-type fuzzy model (Mamdani, 1974; Mamdani, 1977) and other is the Takagi-Sugeno fuzzy model (Takagi & Sugeno, 1985). The Mamdani-type fuzzy model was the first rule-based fuzzy model developed by Mamdani and Assilian (1973) for the purpose of a control system. The form of a Mamdani-type fuzzy model is composed by a collection of parallel rules:

$$R_i : \text{If } x \text{ is } A_i, \text{then } y \text{ is } B_i \qquad \textbf{(2.26)}$$

where $x \in R^p$ is the antecedent variable, $y \in R$ is the consequent variable, A_i represents the linguistic term defined by the fuzzy sets $\mu_{A_i}(x) : R^P \rightarrow [0,1]$, and B_i also represents the linguistic term defined by the fuzzy sets $\mu_{B_i}(y) : R \rightarrow [0,1]$, n is the number of variables relating to the *i*-th rule, and $i = 1, 2,..., n$.

Research based on a black-box identification of Mamdani-type fuzzy models can be reviewed in the works of Cordon and Herrera (1995; 1996), and that of Delgado, Vila, and Gomez-Skarmeta (1994). Work based on a grey-box identification can be reviewed in the work of Lindskog (1996).

2.3.3 Takagi-Sugeno Fuzzy Model

The Takagi-Sugeno (TS) fuzzy model was proposed by Takagi and Sugeno (1985). Compared to Mamdani-type fuzzy models, TS fuzzy models are more effective in dealing with complex and multi-dimensional systems because of their more objective formulation of the consequent part of the fuzzy rules. TS fuzzy models can be categorised into three types based on the inference type of the consequent part; affine TS fuzzy model, homogeneous TS fuzzy model, and singleton fuzzy model. The concepts of these models are as follows:

2.3.3.1 Affine Takagi-Sugeno Fuzzy Model

The affine TS fuzzy model combines a global rule-based description with a local functional description which in the context of black-box identification is chosen as a linear regression model (Hellendoorm & Driankov, 1997).

The form of an affine TS fuzzy model can be represented by

$$R_i : \text{If } x \text{ is } A_i, \text{ then } y_i = a_i^T x + b_i,\ i = 1, 2, ..., k \qquad \textbf{(2.27)}$$

where $x \in X \subset R^P$ is a crisp input vector, $y_i \in R$ is the output of the *i*-th rule, A_i is a fuzzy set $\mu_{A_i}(x) : X \to [0,1]$, $a_i \in R^P$ is a parameter vector and b_i is a scalar offset, and *k* is the number of rules in the rule base.

For black-box identification, the antecedent part of an affine TS fuzzy model defines a fuzzy region of the regressor and in most cases the consequent part of which is an autoregressive model with an exogenous input (Hellendoorm & Driankov, 1997). Research based on black-box identification of affine TS fuzzy models can be reviewed in the works of Babuska and Verbruggen (1994; 1995).

2.3.3.2 Homogeneous TS Fuzzy Model

A homogeneous TS fuzzy model occurs at the offsets $b_i = 0$, $i = 1, 2, ..., k$, and the form of a homogeneous TS fuzzy model can be represented by:

$$R_i : \text{If } x \text{ is } A_i, \text{ then } y_i = a_i^T x,\ i = 1, 2, ..., k \qquad \textbf{(2.28)}$$

where $x \in X \subset R^P$ is a crisp input vector, $y_i \in R$ is the output of the *i*-th rule, A_i is a fuzzy set $\mu_{A_i}(x) : X \to [0,1]$, $a_i \in R^P$ is a parameter vector, and *k* is the number of rules in the rule base.

The homogeneous TS fuzzy model is only applicable for a certain class of systems, as all the consequent models contain the origin.

2.3.3.3 Singleton Fuzzy Model

As the parameter of the consequent part b_i in the affine TS fuzzy model is constant without the parameter vector a_i, the fuzzy model is called a singleton fuzzy model. As the output level of this fuzzy model is constant, this fuzzy model is also called a zero-order TS fuzzy model. The form of a singleton fuzzy model can be represented by:

$$R_i : \text{If } x \text{ is } A_i, \text{ then } y_i = b_i, \quad i = 1, 2, ..., k \qquad \textbf{(2.29)}$$

where $x \in X \subset R^P$ is a crisp input vector, $y_i \in R$ is the output of the *i*-th rule, A_i is a fuzzy set $\mu_{A_i}(x) : X \to [0,1]$, the constants b_i are the consequents, and *k* is the number of rules in the rule base.

2.3.3.4 Summary

Basically, the fuzzification process of the antecedent part of Mamdani-type fuzzy model and Takagi-Sugeno fuzzy model is the same, and the key difference between these two models is the representation of output membership functions of the consequent part. The output membership functions of Mamdani-type fuzzy models are constant, but those of Takagi-Sugeno fuzzy models are either linear or constant. When the output membership functions of Takagi-Sugeno fuzzy models are linear, this Takagi-Sugeno fuzzy model can be called an affine Takagi-Sugeno fuzzy model and when the output membership functions of a Takagi-Sugeno fuzzy model are constant, this can be called a zero-order Takagi-Sugeno fuzzy model or fuzzy singleton model.

2.4 Learning and Identification of Fuzzy Systems Literature Reviews

Literature reviews of learning and identification of fuzzy systems are chosen to describe as follows.

2.4.1 ANFIS: Adaptive-Network-Based Fuzzy Inference Systems

Adaptive-Network-Based Fuzzy Inference Systems (ANFIS) was proposed by Jang (1993). The architecture of ANFIS can identify a feasible set of parameters by a hybrid learning rule combing the backpropagation gradient descent and the least-squares method, as illustrated in Fig. 2.8. For instance, the ANFIS architecture of a first-order TS fuzzy model contains two rules:

$$R_i: \text{If } X \text{ is } A_i \text{ and } Y \text{ is } B_i, \text{ then } f_i = p_i x + q_i y + r_i, \quad i = 1, 2.$$

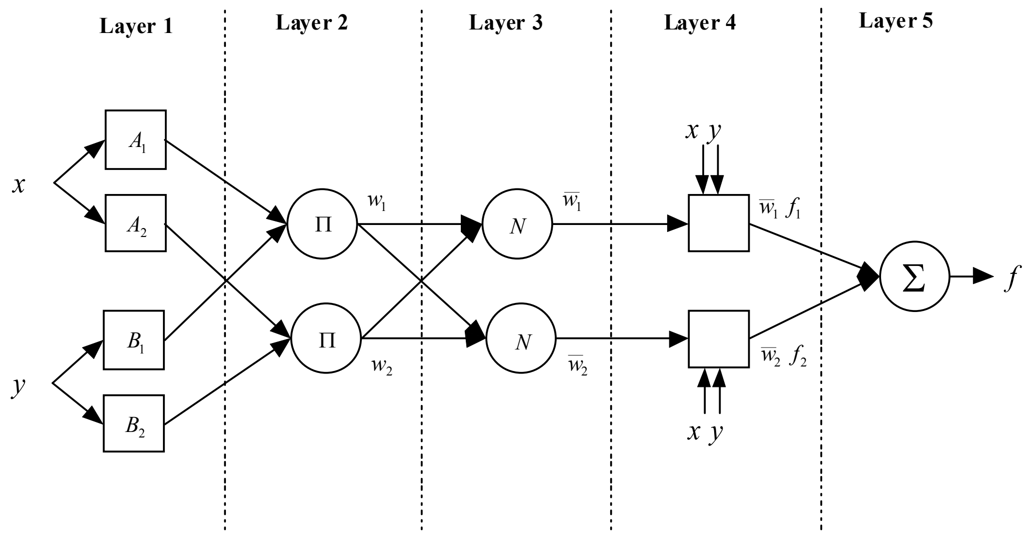

Fig. 2.8 The architecture of ANFIS

According to Fig. 2.8, the fuzzy reasoning mechanism of ANFIS architecture consists of five layers. The detail of these five layers is described as follows

(Jang, 1994):

Layer 1: Each node in this layer generates a membership grade of a linguistic label by bell-shaped membership function. The node function of the *i*-th node can be calculated by:

$$O_i^1 = \mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i}\right)^2\right]^{b_i}} \quad \textbf{(2.30)}$$

where *x* is the input to node *i*, A_i is the linguistic label associated with this node, (a_i, b_i, c_i) is the parameter set that changes the shapes of the membership function, and parameters in this layer are referred to as the premise parameters.

Layer 2: Each node in this layer calculates the firing strength of each rule by multiplication:

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1,2. \quad \textbf{(2.31)}$$

Layer 3: The *i*-th node of this layer calculates the normalisation of the *i*-th rule's firing strength by:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2. \quad \textbf{(2.32)}$$

Layer 4: Each Node *i* in this layer with a node function by:

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \quad \textbf{(2.33)}$$

where $\overline{w}_i$ is the output of layer 3, and (p_i, q_i, r_i) is the parameter set, and parameters in this layer will be referred to as the consequent parameters.

Layer 5: The single node in this layer computes the overall output as the summation of all incoming signals by:

$$O_i^5 = overall\ \ output = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad \textbf{(2.34)}$$

Final: The overall output *f* can be expressed as a linear combinations of the consequent parameters by:

$$f = \overline{w}_1 f_1 + \overline{w}_2 f_2 = (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 + (\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2 \quad \textbf{(2.35)}$$

ANFIS applies two techniques to update parameters. For premise parameters, the membership functions are defined, and ANFIS applies the gradient descent method to fine-tuning parameters. For consequent parameters, the coefficients of each output equations are defined, and ANFIS applies the least-squares method to identify parameters.

2.4.2 A Two-Phase Approach to Fuzzy System Identification

This approach puts focus on identifying fuzzy systems for both the function approximation and classification type of systems. This approach for fuzzy system identification consists of two phases, including Phase One – baseline design as well as Phase Two – Fine Tuning (Hung et al., 2003). The first phase generates a baseline design to identify a prototype fuzzy system from a collection of input-output data pairs by the subtractive clustering algorithm (Chiu, 1994) and the fuzzy c-means clustering algorithm. The second phase is processed to tune the parameters identified in the baseline design by the steepest descent method and the recursive least-squares estimation method. Moreover, the structure of two-phase approach to fuzzy system identification is illustrated in Fig. 2.9.

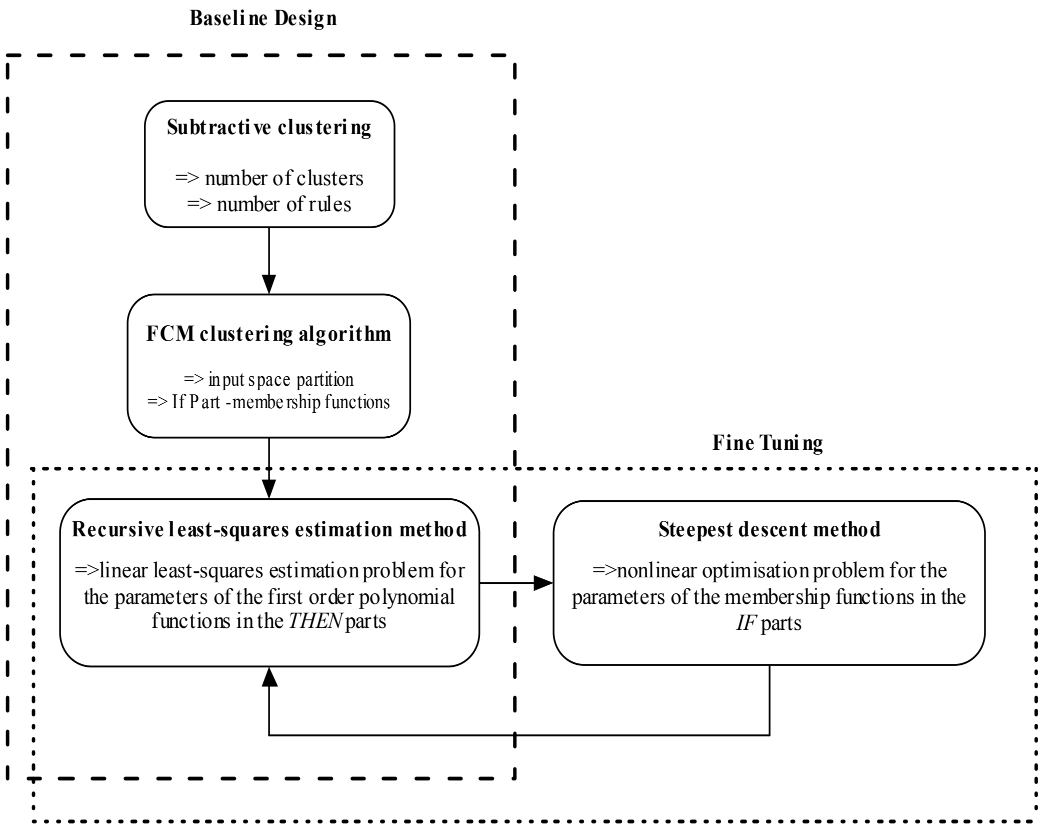

Fig. 2.9 The structure of two-phase approach to fuzzy system identification

Phase One – Baseline Design:

Two clustering techniques, the subtractive clustering method and the fuzzy c-means clustering algorithm, are used to determine the number of rules and the parameters of membership functions.

Phase Two – Fine Tuning:

The fine tuning process is an iterative process, and the steepest descent method as well as the recursive least-squares estimation method are used for fine tuning the parameters of membership functions. Also, the recursive least-squares estimation method is used to determine the parameters of the first-order polynomial functions in the *THEN* parts.

By combining the subtractive clustering method and the fuzzy c-means clustering algorithm, the determination of the number of clusters becomes straightforward for the optimisation-based fuzzy clustering algorithms and the computational effort is therefore kept small.

2.4.3 An Input-Output Clustering Method for Fuzzy System Identification

The purpose of input-output clustering (IOC) algorithm is to determine both the correct number of clusters and their appropriate location by considering both inputs and outputs (Wang et al., 2007). To achieve the above objective, IOC consists of three stages of clustering: rough clustering, refined clustering, and parameter refined training. The task of each stages can be described as follows:

- Rough clustering: determination of output partition
- Refined clustering: determination of the number of sub-clusters and their location within each output constriction
- Parameter refined training: parameters optimisation by the gradient descent method

In the stage of rough clustering, the output space is evenly partitioned based on hard interval partition. Consider $y = f(x_1, x_2, ..., x_n)$, where $y \in [\alpha, \beta]$. The output space is evenly partitioned by m, then $y \in [\alpha_0, \alpha_1) \cup ... \cup [\alpha_{m-1}, \alpha_m]$, where $\alpha_0 = \alpha$ and $\alpha_m = \beta$. Each interval based on the above definition can be recognized an output constriction, and the training data are roughly grouped based on this output partition to obtain a set of clusters.

In the stage of refined clustering, fuzzy c-means clustering algorithm is used to obtain a set of sub-clusters in each output constriction. In order to find the optimal number of sub-clusters, a criterion of separability is used for automatically find the optimal number of sub-clusters in each constriction after the fuzzy c-means clustering algorithm is completed. Moreover, the criterion of separability S^r is defined by:

$$S^r = \frac{\sum_{k=1}^{N^c}\left(\sum_{j=1}^{m^c}\left(A_j^c\left(x^k\right)\right)\right)}{\sum_{c=1}^{m}\left(\sum_{k=1}^{N^c}\left(\sum_{j=1}^{m^c}\left(A_j^c\left(x^k\right)\right)\right)\right)} \qquad \textbf{(2.36)}$$

where *j* is the *j*-th sub-cluster within the *r*-th output constriction, N^r is the total number of data located within the *r*-th output constriction, and $m^r\left(m^c\right)$ is the number of sub-cluster within the *r*-th(*c*-th) output constriction.

A large value of S^r indicates a better separability, and which means the data within the *r*-th output constriction can be well separated from the remaining data. Therefore, the optimal number of sub-cluster m^r with the largest value of separability S^r is the key point to find the correct number of sub-clusters for each output constriction. In other words, the largest value of S^r indicates the optimal number of sub-cluster m^r within the *r*-th output constriction. The optimal number of sub-cluster m^r can be calculated by:

$$m^r = \arg\left(\max_{1 \ldots m^{\max}}\left(s^r\right)\right) \qquad \textbf{(2.37)}$$

In the stage of parameter refined training, the parameters of Gaussian membership functions for the obtained fuzzy system are fine-tuned by the gradient descent method.

2.5 Clustering

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering (Han & Kamber, 2006)

Clustering, can also be called cluster analysis or data segmentation, and means assigning a set of observations or data into subsets or clusters. The observations or data located in the same subsets or clusters possess certain similar characteristics to each other as well as sharing certain properties. The task of clustering is to learn a classification from observations or data, and the purpose of cluster analysis is to recognise similar and dissimilar observations or data. Similar observations or data will be clustered in the same subsets, but dissimilar observations or data will be clustered to different subsets. In other words, the observations or data assigned to the same subset or cluster are more closely related than observations or data assigned to different subsets or clusters.

So far as the learning types are concerned, clustering algorithms can be recognised as unsupervised learning methods predicting category labels. Clustering can be used for the purpose of exploratory data analysis and generalisation, and it is a prevalent technique for statistical data analysis applied in machine learning, pattern recognition, and data mining. In order to understand the properties of clustering, a brief introduction follows.

2.5.1 Properties of Clustering

Clustering algorithms have a variety of properties, consisting of hierarchical, flat, hard, soft, iterative, and disjunctive. Among these properties, certain properties can be included together in a clustering algorithm, but others are excluded.

- ❖ **Hierarchical Clustering.** As implied by the name, a hierarchical clustering algorithm creates a hierarchy of clusters of decreasing generality, and a series of partitions take place by an agglomerative scheme or divisive scheme in the procedure of hierarchical clustering algorithms. As the framework of hierarchical clustering algorithms is a hierarchical structure, the specialisation ability of hierarchical clustering algorithms is better than that of flat clustering algorithms. Therefore, the advantages of hierarchical clustering algorithms are providing detailed data analysis and more information than flat clustering algorithms. Further, hierarchical clustering algorithms includes single linkage clustering algorithm, complete linkage clustering algorithm, average linkage clustering algorithm, average group linkage clustering algorithm, and Ward's hierarchical clustering algorithm.

- **Flat Clustering.** Different from hierarchical clustering algorithms, flat clustering algorithm creates a flat set of clusters without any explicit structure relating clusters to each other. Flat clustering algorithms are efficient for dealing with large data sets in clustering. Therefore, flat clustering algorithms usually have better generalisation ability than hierarchical clustering algorithms. The k-means clustering algorithm and the fuzzy c-means clustering algorithm are the most used methods of flat clustering algorithms.

- **Hard Clustering.** In hard clustering algorithms, each observation or data is assigned to exactly one cluster. Hierarchical clustering algorithms and the k-means clustering algorithm belong to hard clustering algorithms.

- **Soft Clustering.** In soft clustering algorithms, each observation or data is assigned to a cluster based on the probability of belonging to a cluster. In other words, each observation or data has fractional membership in different clusters. The fuzzy c-means clustering algorithm belongs to soft clustering algorithms, as each observation or data carries a different membership value to different subsets or clusters.

- ❖ **Iterative Clustering.** Iterative clustering algorithms start with an initial data set of clusters and assign each data set to the appropriate cluster with the iterative procedure until the algorithms entirely converge. The k-means clustering algorithm and the fuzzy c-means clustering algorithm are the most used clustering algorithms of iterative clustering algorithms.
- ❖ **Disjunctive Clustering.** In disjunctive clustering algorithms, an observation or data can belong to more than one cluster. The fuzzy c-means clustering algorithm can be regarded as a kind of disjunctive clustering algorithm, as each observation or data carries a different membership value to different subsets or clusters.

To sum up, each clustering property or clustering algorithm has its own niche. There is no absolute clustering property or clustering algorithm, because different clustering algorithms are suitable for different problems or applications. The more clusters generated cannot reach a better performance for clustering purposes, because a large number of clusters cannot guarantee both the correct number of clusters as well as the appropriate location for which have been determined. By other means, the correct number of clusters as well as the appropriate location of the clusters is the primary requirements when justifying a positive clustering technique.

2.5.2 Clustering Techniques for Classification Problems and Function Approximation Problems

Clustering techniques have been used to solve classification and pattern recognition problems. To date, a great diversity of clustering techniques have been developed, one of which is to apply clustering techniques to construct initial models for function approximators. Before applying clustering techniques to work on function approximation problems, there are certain concepts between classification problems and function approximation problems which need to be (Gonzalez *et al.,* 2002).

- ❖ In classification problems, the output variable is discrete and takes values in a finite label set which is defined *a priori*.
- ❖ In function approximation problems, the output variable is continuous and takes any of the infinite values within an interval of real numbers.

As the values of the output variable for function approximation problems is different to the values of the output variable for classification problems, the system output values may be accepted if the system output values are "sufficiently close" to the desired output values. However, this behaviour is not desirable for classification problems, because the values of a discrete output variable may not be related by any numerical values in a classification problem. Clustering techniques do not take the interpolation properties into account, because it is not necessary when dealing with discrete variables present in classification problems (Gonzalez *et al*., 2002).

The following sections will introduce the most used clustering algorithms, including the k-means clustering algorithm, the fuzzy c-means clustering algorithm, and the average group linkage clustering algorithm.

2.5.3 K-Means Clustering Algorithm

The k-means clustering algorithm is an algorithm that is simultaneously a flat, hard, as well as an iterative clustering property based on Euclidian distance, which assigns observations or data to subsets or clusters by iteratively calculating the shortest distance between each data and the corresponding cluster centroid. Basically, the k-means clustering algorithm groups data based on its attributes into K number of groups. K is an integer whose value is equal to or greater than

1.0. The grouping determines the value of *K* first, and then selects the best cluster centroid by minimising the sum of squares of distance between data and the corresponding cluster centroid iteratively, until the best cluster centroid of each group is selected. The brief procedure of the algorithm of k-means clustering can be described as follows:

- Step 1: Determine the number of the partition "*K*".
- Step 2: Determine the initial cluster centroid of each cluster.
- Step 3: Calculate the distance between each data and the corresponding cluster centroid by Euclidean distance.
- Step 4: Determine the best cluster centroid by minimising the sum of squares of distance between data and the corresponding cluster centroid.
- Step 5: Keep performing the procedure until no data needs to move groups.
- End: The entire algorithm converges.

Also, the complete procedure of the k-means clustering algorithm can be illustrated in Fig. 2.10 (Teknomo, 2004):

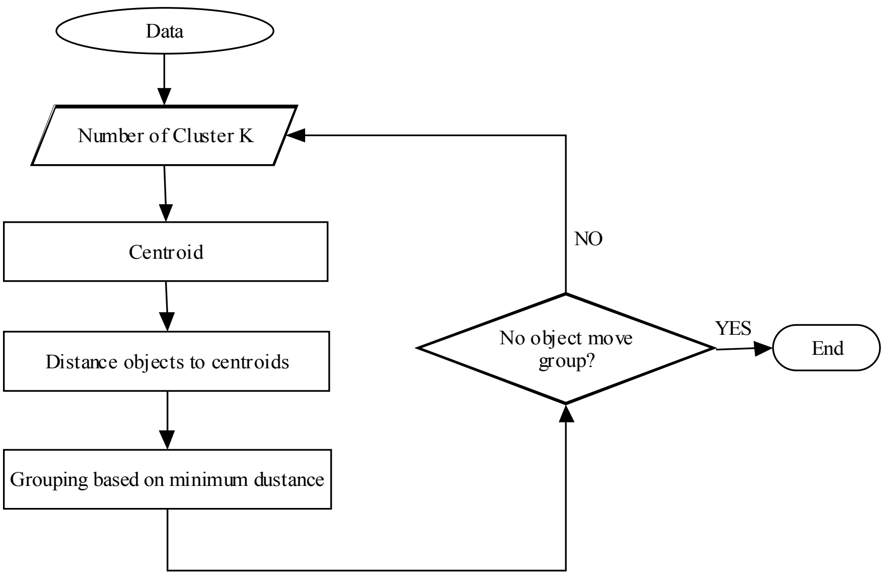

Fig. 2.10 The system of the algorithm of K-Mean Clustering

As the entire procedure converges, the following properties are present:

- The number of K cluster(s) has been generated.
- The best cluster centroid of each cluster (K cluster) has been calculated.
- The distance between each data set to the corresponding cluster centroid is the minimum distance in the whole topology.

Finally, according to the proposition of *Kardi Teknomo*, the weakness of k-means Clustering can be described as follows (Teknomo, 2004):

- ❖ When the numbers of data are not so many, initial grouping will determine the cluster significantly.
- ❖ The number of cluster, K, must be determined beforehand.
- ❖ We never know the real cluster, using the same data, if it is inputted in a different way may produce different cluster if the number of data is a few.
- ❖ We never know which attribute contributes more to the grouping process since we assume that each attribute has the same weight.
- ❖ A less value of K may result in the problem of underfitting, and an excessive value of K may result in the problem of overfitting

2.5.4 Fuzzy C-Means Clustering Algorithm

Fuzzy Clustering algorithms can be regarded as a well-identified family of rule induction techniques, which organise and categorise data into homogeneous groups, and each partition or group represents a rule associated to each cluster (Guillaume, 2001). Meanwhile, fuzzy c-means is one of the essential fuzzy clustering algorithms, and allows data to belong to more than one cluster. The fuzzy c-means clustering algorithm was developed in 1973 by Dunn, and improved by Bezdek in 1981. The concepts of the fuzzy c-means clustering algorithm represents the mechanism of fuzzy logic, which regards each cluster as a membership function respectively, and each datum is bound to each cluster by means of a membership function. After the iterative process has been finished, the datum will be considered to be located at the highest degree among all the clusters. So far as the basic mechanism of fuzzy c-means clustering algorithm is concerned, the clustering algorithm possesses several properties of clustering, including soft clustering, flat clustering, iterative clustering, and disjunctive clustering. In addition to the properties described above, the fuzzy c-means clustering algorithm is based on minimisation of the following criterion (Matteucci, 2003):

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij}^{m} \left\| x_i - c_j \right\|^2 , \quad 1 \le m < \infty \qquad \textbf{(2.38)}$$

where m is a real parameter greater than 1.0, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the i-th of d-dimensional measured data, c_j is the d-dimension centre of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the centre.

Generally, there is no specific value of the parameter *m* of equations 2.38. The value of the parameter *m* of equations 2.38 is usually set equal to 2.0, because this is equivalent to normalising the coefficient linearly to make their sum 1.0. When the value of *m* is close to 1, then the cluster centre closest to the point is given much more weight than the others. Based on the iterative process of equation 2.38, the update of membership u_{ij} and the cluster centre C_j can be calculated as follows:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}} \qquad \textbf{(2.39)}$$

$$c_j = \frac{\sum_{i=1}^{N} \mu_{ij}^{m} \cdot x_i}{\sum_{k=1}^{N} \mu_{ij}^{m}} \qquad \textbf{(2.40)}$$

The iteration keeps performing until either the algorithm perfectly converges or the termination criterion has been reached. Meanwhile, the termination criterion can be determined as follows:

$$\max_{ij} \left\{ \left| \mu_{ij}^{(k+1)} - \mu_{ij}^{(k)} \right| \right\} < \varepsilon \qquad \textbf{(2.41)}$$

where ε is a termination criterion between 0 and 1.0, and k are the iteration steps.

The value of ε is determined according to the requirements, such as a higher accuracy or lower computational cost, and the above equation 2.41 converges to a local minimum or a saddle point of J_m. Generally, the smaller values of ε indicates a higher accuracy with increased computational cost, and the bigger values of ε indicates a lower accuracy with less computational cost.

2.5.5 Average Group Linkage Clustering Algorithm

The average group linkage method is one of the hierarchical clustering methods, and calculates the distance between clusters in hierarchical cluster analysis. Therefore, it is a kind of distance-based clustering with hierarchical clustering analysis. In this method, the clusters are represented by the mean value of each variable, and thus the distance between clusters is calculated by the average values or mean vector of clusters. As the distance between two clusters is the minimum, these two clusters will be merged into a single cluster. In other words, the new cluster has the minimum average distance between the data points it contains. The equation for the average group linkage method can be represented by:

$$D(C_m, C_n) = D(Mean(C_m), Mean(C_n)) \quad \textbf{(2.42)}$$

where $D(C_m, C_n)$ is the distance between cluster C_m and cluster C_n, and $D(Mean(C_m), Mean(C_n))$ is the distance between the mean vectors of these two clusters. Also, Cluster C_m and Cluster C_n are merged when $D(C_m, C_n)$ is minimum.

The procedure of the average group linkage clustering algorithm can be summarised as follows:

- Step 1: Associate each data to a separate cluster
- Step 2: Calculate every possible distance between clusters
- Step 3: Merge the clusters with the minimum distance to each other to a new single cluster.
- Step 4: Calculate every possible distance between the new cluster and all of other clusters.
- Step 5: Keep repeating Step 3 and Step 4 until no clusters are required to be merged.

The advantage of the average group linkage clustering algorithm is the inherent similarity between data and the detailed information of the *a priori* knowledge which may be positively discovered by the property of the hierarchical clustering of the average group linkage clustering algorithm.

2.6 Similarity-Based Pruning Strategy

The application of similarity analysis has evolved from the concept of similarity relations. The rudimentary concept of similarity relations between fuzzy sets, fuzzy numbers or fuzzy membership functions were proposed by Zadeh (1971).

Zadeh introduced the concept of similarity relations to indicate the degree of similarity between elements in a universe of discourse for concretising the utility of the theory of fuzzy sets, and the theory of relations to fuzzy sets has therefore been considered as an extension of the theory of fuzzy sets. With the positive significance of similarity analysis, several methods have applied the concept of similarity relations by calculating the degree of similarity between fuzzy sets, fuzzy numbers or fuzzy membership functions, and the rudimentary development of the relevant research is outlined by Pappis and Karacapilidis (1993), Chen and Chen (1995), Hsieh and Chen, (1999) and Lee (1999).

2.6.1 Three Types of Basic Similarity Measures of Fuzzy Values

In 1993, Pappis and Karacapilidis presented three types of basic measures of similarity of fuzzy values, including a measure based on the operations of union and intersection, a measure based on the maximum difference, and a measure based on the difference as well as the sum of grades of membership.

2.6.1.1 Measure Based on the Operations of Union and Intersection

So far as the measure based on the operations of union and intersection is concerned, the degree of similarity $M_{A,B}$ of the fuzzy sets A and B can be defined by:

$$M_{A,B} = \frac{\sum_i (a_i \wedge b_i)}{\sum_i (a_i \vee b_i)} \tag{2.43}$$

where $a \wedge b$ denotes $\min(a,b)$, $a \vee b$ denotes $\max(a,b)$, and a_i as well as b_i are the values of fuzzy set A and B respectively. Moreover, $0 \le M_{A,B} \le 1$,

$M_{A,B} = M_{B,A}$, and A and B are identical if and only if $M_{A,B} = 1$. A larger value of $M_{A,B}$ indicates a higher degree of similarity between fuzzy sets.

2.6.1.2 Measure Based on the Maximum Difference

So far as the measure based on the maximum difference is concerned, the degree of similarity $L_{A,B}$ of the fuzzy sets A and B can be defined by:

$$L_{A,B} = 1 - \max_i \left(\left| a_i - b_i \right| \right) \qquad \textbf{(2.44)}$$

Moreover, $0 \le L_{A,B} \le 1$, $L_{A,B} = L_{B,A}$, and A and B are identical if and only if $L_{A,B} = 1$. A larger value of $L_{A,B}$ indicates a higher degree of similarity between fuzzy sets.

2.6.1.3 Measure Based on the Difference and the Sum of Grades of Membership

So far as the measure based on the difference and the sum of grades of membership is concerned, the degree of similarity $S_{A,B}$ of the fuzzy sets A and B can be defined by:

$$S_{A,B} = 1 - \frac{\sum_i \left| a_i - b_i \right|}{\sum_i \left(a_i + b_i \right)} \qquad \textbf{(2.45)}$$

Moreover, $0 \le S_{A,B} \le 1$, $S_{A,B} = S_{B,A}$, and A and B are identical if and only if $S_{A,B} = 1$. A larger value of $S_{A,B}$ indicates a higher degree of similarity between fuzzy sets.

2.6.2 Similarity Measure Based on the Distance between Fuzzy Numbers

Based on the work proposed by Pappis and Karacapilidis (1993), Chen and Lin (1995) extended the work and applied the concept of similarity to fuzzy risk analysis. In the work of Chen and Lin, the calculation of the degree of similarity is based on the distance between fuzzy numbers, and the criterion of the degree of similarity $S(\widetilde{A},\widetilde{B})$ between fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$ can be calculated by:

$$S(\widetilde{A},\widetilde{B}) = 1 - \frac{\sum_{i=1}^{n} |a_i - b_i|}{n} \quad \textbf{(2.46)}$$

where $\widetilde{A} = (a_{1,} a_2, ..., a_n)$, $\widetilde{B} = (b_{1,} b_2, ..., b_n)$, and *n* is the number of vertex of the fuzzy number. As *n* is 3, the shape of the fuzzy number is triangular and the type of membership function is a triangular membership function, where $\widetilde{A} = (a_{1,} a_2, a_3)$, $\widetilde{B} = (b_{1,} b_2, b_3)$. In cases where *n* is 4, the shape of the fuzzy number is trapezoid and the type of membership function is a trapezoid membership function, where $\widetilde{A} = (a_{1,} a_2, a_3, a_4)$, $\widetilde{B} = (b_{1,} b_2, b_3, b_4)$. Moreover, the value of $S(\widetilde{A},\widetilde{B})$ is between 0 and 1, and a larger value of $S(\widetilde{A},\widetilde{B})$ indicates a higher degree of similarity between fuzzy numbers.

To facilitate the computation of the aggregation method for fuzzy opinions of group decision making, Lee (1999) proposed a new criterion to measure the degree of similarity for fuzzy numbers for the purpose of simple computation and the identification of disjointed fuzzy numbers. Meanwhile, the degree of similarity $S(\widetilde{A},\widetilde{B})$ between fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$ can be calculated by:

$$S(\widetilde{A},\widetilde{B}) = 1 - \frac{\left\| \widetilde{A} - \widetilde{B} \right\|_{lp}}{\left\| U \right\|} \times n^{-\frac{1}{p}} \quad \textbf{(2.47)}$$

where $\widetilde{A}=(a_{1,}a_{2},\dots,a_{n})$, $\widetilde{B}=(b_{1,}b_{2},\dots,b_{n})$, n is the number of vertex of the fuzzy number, l_p metric is defined as $\left\|\widetilde{A}-\widetilde{B}\right\|_{lp}=(\sum_{i=1}^{n}(|a_i-b_i|)^p)^{\frac{1}{p}}$, the default value of p is 1, U is the universe of discourse, and $\|U\|=\max(U)-\min(U)$. As n is 3, the shape of the fuzzy number is triangular and the type of membership function is a triangular membership function, where $\widetilde{A}=(a_{1,}a_{2},a_{3})$, $\widetilde{B}=(b_{1,}b_{2},b_{3})$. In cases where n is 4, the shape of the fuzzy number is trapezoid and the type of membership function is a trapezoid membership function, where $\widetilde{A}=(a_{1,}a_{2},a_{3},a_{4})$, $\widetilde{B}=(b_{1,}b_{2},b_{3},b_{4})$. Moreover, the value of $S(\widetilde{A},\widetilde{B})$ is between 0 and 1, and a larger value of $S(\widetilde{A},\widetilde{B})$ indicates a higher degree of similarity between fuzzy numbers.

2.6.3 Similarity Measure Based on Graded Mean Integration Representation Distance

The work proposed by Hsieh and Chen (1999) was different to those described above as it applied a graded mean integration representation distance to measure the degree of similarity between fuzzy numbers. The degree of similarity $S(\widetilde{A},\widetilde{B})$ between fuzzy numbers can be calculated by:

$$S(\widetilde{A},\widetilde{B})=\frac{1}{1+d(\widetilde{A},\widetilde{B})} \qquad \textbf{(2.48)}$$

where the distance of the graded mean integration representation can be defined to be $d(\widetilde{A},\widetilde{B})=\left|P(\widetilde{A})-P(\widetilde{B})\right|$, and $P(\widetilde{A})$ as well as $P(\widetilde{B})$ are the graded mean integration representation of the fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$.

The definitions of the graded mean integration representation of fuzzy numbers are different based on the different shape of the fuzzy numbers, and can be triangular and trapezoid types. So far as the triangular fuzzy number is concerned, the definition of fuzzy number $\widetilde{A} = (a_{1,} a_2, a_3)$ and $\widetilde{B} = (b_{1,} b_2, b_3)$ can be defined as follows:

$$P(\widetilde{A}) = \frac{a_1 + 4a_2 + a_3}{6}, \quad P(\widetilde{B}) = \frac{b_1 + 4b_2 + b_3}{6} \tag{2.49}$$

So far as the trapezoid fuzzy number is concerned, the definition of fuzzy number $\widetilde{A} = (a_{1,} a_2, a_3, a_4)$ and $\widetilde{B} = (b_{1,} b_2, b_3, b_4)$ can be defined as follows:

$$P(\widetilde{A}) = \frac{a_1 + 2a_2 + 2a_3 + a_4}{6}, \quad P(\widetilde{B}) = \frac{b_1 + 2b_2 + 2b_3 + b_4}{6} \tag{2.50}$$

Moreover, $0 \le S(\widetilde{A}, \widetilde{B}) \le 1$ as well as $S(\widetilde{A}, \widetilde{B}) = S(\widetilde{B}, \widetilde{A})$, and a larger value of $S(\widetilde{A}, \widetilde{B})$ indicates a higher degree of similarity between fuzzy numbers.

2.6.4 Similarity Measure Based on Centre-of-Gravity Point

In order to make fuzzy numbers more clear, Chen (1985) put the membership value as the essence of fuzzy numbers for recognising the normal fuzzy number or not. In principle, Chen's work defined the generalised trapezoidal fuzzy number $\widetilde{A}$ as $\widetilde{A} = (a_{1,} a_2, a_3, a_4; w_{\widetilde{A}})$, where $w_{\widetilde{A}}$ presents the maximum membership value of fuzzy number $\widetilde{A}$, $0 < w_{\widetilde{A}} \le 1$, and $\{a_i \mid a_i \in R\}$. Basically, the maximum membership value of a normal fuzzy number is 1. Though the shape of the generalised fuzzy number is defined as a trapezoidal fuzzy number based on $a_1 \ne a_2 \ne a_3 \ne a_4$, the type of the fuzzy number can be altered by adjusting the relationship between the essences of fuzzy numbers. Since a normal fuzzy number has been defined as $\widetilde{A} = (a_{1,} a_2, a_3, a_4)$, a rectangular fuzzy number can be recognised by $a_1 = a_2$ and $a_3 = a_4$; the triangular fuzzy number can be recognised by $a_2 = a_3$; and the crisp value is defined as $a_1 = a_2 = a_3 = a_4$.

2.6.4.1 Centre-of-Gravity Point of Fuzzy Numbers

According to the definition of generalised fuzzy number as defined by Chen (1985) Chen's later work (2000) proposed a method with which to calculate the centre-of-gravity (COG) point for handling the fuzzy ranking and the defuzzification problems based on the concepts of geometry. The formula for calculating the centre-of-gravity (COG) point (x^*, y^*) of a trapezoidal fuzzy number $\widetilde{A} = (a_1, a_2, a_3, a_4; w_{\widetilde{A}})$, $COG(\widetilde{A}) = (x_{\widetilde{A}}^*, y_{\widetilde{A}}^*)$, can be represented as follows:

$$y_{\widetilde{A}}^* = \begin{cases} \dfrac{w_{\widetilde{A}} \times \left(\dfrac{a_3 - a_2}{a_4 - a_1} + 2 \right)}{6}, & \text{if } a_1 \neq a_4 \text{ and } 0 < w_{\widetilde{A}} \leq 1, \\ \dfrac{w_{\widetilde{A}}}{2}, & \text{if } a_1 = a_4 \text{ and } 0 < w_{\widetilde{A}} \leq 1 \end{cases}, \qquad \textbf{(2.51)}$$

$$x_{\widetilde{A}}^* = \frac{y_{\widetilde{A}}^*(a_3 + a_2) + (a_4 + a_1)(w_{\widetilde{A}} - y_{\widetilde{A}}^*)}{2w_{\widetilde{A}}}. \qquad \textbf{(2.52)}$$

2.6.4.2 Calculation of Similarity Measure Based on COG Point

In order to acquire a more precise degree of similarity between fuzzy numbers, Chen & Chen (2001) further investigated the previous work by Chen (1985), Chen and Lin (1995) and Chen and Chen (2000) and integrated them into one method. In Chen (2001), the COG points $\mathrm{COG}(\widetilde{A}) = (x_{\widetilde{A}}^*, y_{\widetilde{A}}^*)$ of fuzzy number $\widetilde{A}$ and $\mathrm{COG}(\widetilde{B}) = (x_{\widetilde{B}}^*, y_{\widetilde{B}}^*)$ of fuzzy number $\widetilde{B}$ are obtained using equations 2.51 and 2.52 respectively, and the degree of similarity between the trapezoidal fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$ can be calculated by:

$$S(\widetilde{A},\widetilde{B}) = \left[1 - \frac{\sum_{i=1}^{4}|a_i - b_i|}{4}\right] \times \left(1 - |x_{\widetilde{A}}^{*} - x_{\widetilde{B}}^{*}|\right)^{\left\lceil \frac{S_{\widetilde{A}} + S_{\widetilde{B}}}{2} \right\rceil} \times \frac{\min\left(y_{\widetilde{A}}^{*}, y_{\widetilde{B}}^{*}\right)}{\max\left(y_{\widetilde{A}}^{*}, y_{\widetilde{B}}^{*}\right)}, \qquad \textbf{(2.53)}$$

where $S(\widetilde{A},\widetilde{B}) \in [0,1]$; if $\frac{S_{\widetilde{A}} + S_{\widetilde{B}}}{2} = 0$, then $\left\lceil \frac{S_{\widetilde{A}} + S_{\widetilde{B}}}{2} \right\rceil = 0$; if $0 < \frac{S_{\widetilde{A}} + S_{\widetilde{B}}}{2} \le 1$, then $\left\lceil \frac{S_{\widetilde{A}} + S_{\widetilde{B}}}{2} \right\rceil = 1$; where $S_{\widetilde{A}}$ and $S_{\widetilde{B}}$ are the bases of the generalised trapezoidal fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$ respectively, and which can be defined as $S_{\widetilde{A}} = a_4 - a_1$ as well as $S_{\widetilde{B}} = b_4 - b_1$. Also, $\left\lceil \frac{S_{\widetilde{A}} + S_{\widetilde{B}}}{2} \right\rceil$ denotes the upper boundary of the value of $\frac{S_{\widetilde{A}} + S_{\widetilde{B}}}{2}$.

Moreover, $0 \le S(\widetilde{A},\widetilde{B}) \le 1$, $S(\widetilde{A},\widetilde{B}) = S(\widetilde{B},\widetilde{A})$, and $\widetilde{A}$ and $\widetilde{B}$ are identical if and only if $S(\widetilde{A},\widetilde{B}) = 1$. A larger value of $S(\widetilde{A},\widetilde{B})$ indicates a higher degree of similarity between fuzzy numbers. Based on equation 2.53, the degree of similarity between the triangular fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$ can be calculated by adjusting the essences of the trapezoidal fuzzy number $\widetilde{A} = (a_{1,} a_2, a_3, a_4; w_{\widetilde{A}})$ to $a_2 = a_3$.

2.6.5 Similarity Measure Based on Overlapped Area

In addition to the above methods, the methods of Lin (1995) and Dubois and Prade (1982) measure the degree of similarity by calculating the overlapping area between fuzzy sets, fuzzy numbers, or fuzzy membership functions, and the fuzzy similarity measure between fuzzy set A and fuzzy set B can be defined as follows:

$$S(A,B)=\frac{M(A\cap B)}{M(A\cup B)}=\frac{M(A\cap B)}{M(A)+M(B)-M(A\cap B)} \quad \textbf{(2.54)}$$

where ∩ denotes the intersection of fuzzy sets A and B, ∪ denotes the union of fuzzy sets A and B, and M(˙) is the size of a fuzzy set.

Moreover, $0\le S(A,B)\le 1$, $S(A,B)=S(B,A)$, and A and B are identical if and only if $S(A,B)=1$. A larger value of $S(A,B)$ indicates a higher degree of similarity between fuzzy sets, fuzzy numbers, or fuzzy membership functions.

2.6.6 Similarity Measure Based on Six Cases by Leng's Work

In order to decrease the complex of the neuro-fuzzy system, as well as to improve the overlap between membership functions, Leng *et al.* (2009) proposed a pruning strategy based on similarity analysis to eliminate redundant membership functions and merge similar membership functions into a new membership function. According to equation 2.54, Leng's work performed the similarity analysis by finalising six different cases based on triangular membership functions. These six different cases totally cover all possible similarity relationships based on triangular membership functions, and are classified into:

- Case 1: One membership function is the subset of the other membership function, $M(B)$ is the subset of $M(A)$, $M(B)\subseteq M(A)$;
- Case 2: There is no intersection between membership functions $M(A)$ and

$M(B)$, $M(A) \cap M(B) = 0$;

- Case 3: Two membership functions $M(A)$ and $M(B)$ possess single intersection point;
- Case 4: Two membership functions $M(A)$ and $M(B)$ possess two intersection points between the both side of membership function $M(A)$ and the left side of membership function $M(B)$;
- Case 5: Two membership functions $M(A)$ and $M(B)$ possess two intersection points between the right side of membership function $M(A)$ and the both side of membership function $M(B)$;
- Case 6: Two membership functions $M(A)$ and $M(B)$ possess three intersection points between the both side of membership functions $M(A)$ and $M(B)$.

Basically, these six different cases based on the relationship of triangular membership functions can be illustrated by Fig. 2.11-2.16. If the degree of similarity between two fuzzy sets $A(\alpha_1, \beta_1, \gamma_1)$ and $B(\alpha_2, \beta_2, \gamma_2)$ exceeds the threshold value λ, $S(A,B) \geq \lambda$, where $0 < \lambda \leq 1$, the fuzzy memberships can be merged into one new fuzzy set $C(\alpha, \beta, \gamma)$. In addition to the similarity analysis, Leng's work also provided the calculation of the new fuzzy set or triangular membership function.

Case 1:

In case 1, illustrated in Fig. 2.11, the membership function $M(B)$ has been included in the membership function $M(A)$, $M(B) \subseteq M(A)$, and $M(A)$ as well as $M(B)$ possess the same centre $\beta_1 = \beta_2$ with no intersection, where $M(A) = \Delta P\alpha_1\gamma_1$, $M(B) = \Delta P\alpha_2\gamma_2$, and Δ means the triangle area. Based on equation 2.54, the degree of similarity between fuzzy sets $A(\alpha_1, \beta_1, \gamma_1)$ and $B(\alpha_2, \beta_2, \gamma_2)$ is $M(A \cap B) = M(B)$, and a new fuzzy set $C(\alpha, \beta, \gamma)$ merged by fuzzy sets $A(\alpha_1, \beta_1, \gamma_1)$ and $B(\alpha_2, \beta_2, \gamma_2)$ can be calculated by:

$$\alpha = \min(\alpha_1, \alpha_2),\ \ \beta = \beta_1 = \beta_2,\ \ \gamma = \max(\gamma_1, \gamma_2) \qquad \textbf{(2.55)}$$

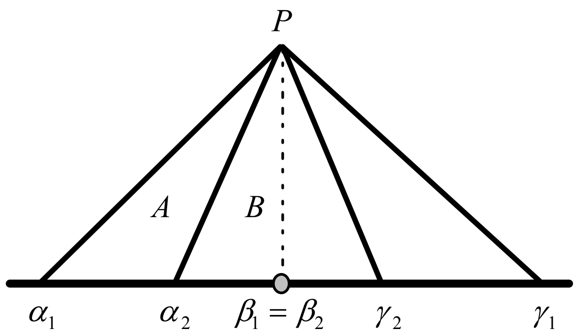

Fig. 2.11 Case 1 of Leng's work, $M(B)$ is the subset of $M(A)$, $M(B) \subseteq M(A)$.

Case 2:

In case 2, illustrated in Fig. 2.12, there is no intersection between the membership functions $M(A)$ and $M(B)$. Therefore, the intersection of $M(A)$ and $M(B)$ is $M(A \cap B) = 0$, and the degree of similarity for case 2 is $S(A, B) = 0$. As $S(A, B) = 0$, there is no combination required in this case.

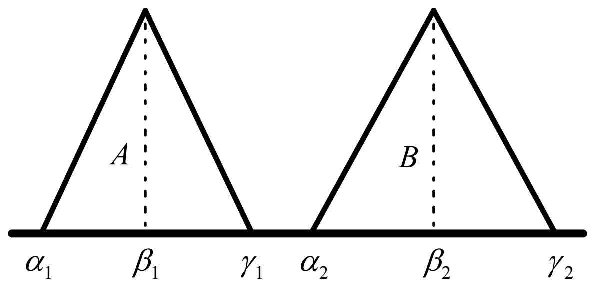

Fig. 2.12 Case 2 of Leng's work, $M(A)$ and $M(B)$ with no intersection, $M(A) \cap M(B) = 0$.

Case 3:

In case 3, illustrated in Fig. 2.13, the membership functions $M(A)$ and $M(B)$ possess single intersection point at (s,h), and the degree of similarity between membership functions $M(A)$ and $M(B)$, $S(A,B)$, can be calculated by:

$$h = \frac{\gamma_1 - \alpha_2}{\gamma_1 - \beta_1 + \beta_2 - \alpha_2} \quad \textbf{(2.56)}$$

$$M(A \cap B) = \frac{1}{2}(\gamma_1 - \alpha_2)h \quad \textbf{(2.57)}$$

$$M(A \cup B) = \frac{1}{2}(\gamma_1 - \alpha_1 + \gamma_2 - \alpha_2) - M(A \cap B) \quad \textbf{(2.58)}$$

Moreover, if $S(A,B) \geq \lambda$, the new fuzzy set $C(\alpha,\beta,\gamma)$ merged by fuzzy sets $A(\alpha_1,\beta_1,\gamma_1)$ and $B(\alpha_2,\beta_2,\gamma_2)$ can be calculated by:

$$\alpha = \alpha_1,\ \beta = \frac{1}{2}(\beta_1 + \beta_2),\ \gamma = \gamma_2. \quad \textbf{(2.59)}$$

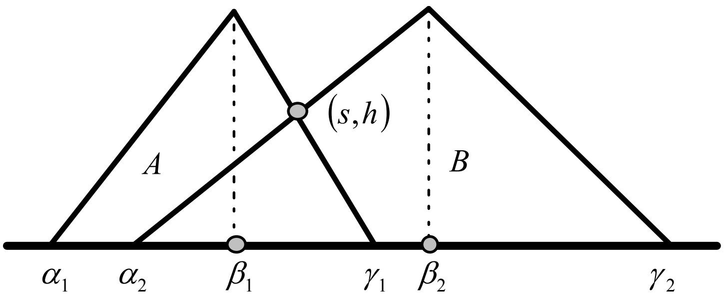

Fig. 2.13 Case 3 of Leng's work, $M(A)$ and $M(B)$ with one intersection point at (s,h).

Case 4:

In case 4, illustrated in Fig. 2.14, the membership functions $M(A)$ and $M(B)$ possess two intersection points at (s_1,h_1) and (s_2,h_2) between both sides of membership function $M(A)$ and the left side of membership function $M(B)$, and the degree of similarity between membership functions $M(A)$ and $M(B)$, $S(A,B)$, can be calculated by:

$$M(A \cap B) = \frac{1}{2}\{(\beta_1-\alpha_1)h_1^2+(\gamma_1-\beta_1)h_2^2+(h_1+h_2)[(\gamma_1-\alpha_1)-(\beta_1-\alpha_1)h_1+(\gamma_1-\beta_1)h_2]\} \quad \textbf{(2.60)}$$

$$\text{where } h_1 = \frac{\alpha_1-\alpha_2}{\beta_2-\alpha_2+\alpha_1-\beta_1},\ h_2 = \frac{\gamma_1-\alpha_2}{\beta_2-\alpha_2+\gamma_1-\beta_1}, \quad \textbf{(2.61)}$$

$$\text{and } M(A \cup B) = \frac{1}{2}(\gamma_1-\alpha_1+\gamma_2-\alpha_2)-M(A \cap B). \quad \textbf{(2.62)}$$

Moreover, if $S(A,B) \geq \lambda$, the new fuzzy set $C(\alpha,\beta,\gamma)$ merged by fuzzy sets $A(\alpha_1,\beta_1,\gamma_1)$ and $B(\alpha_2,\beta_2,\gamma_2)$ can be calculated by:

$$\alpha = \alpha_2, \beta = \frac{1}{2}(\beta_1 + \beta_2), \gamma = \gamma_2. \quad \textbf{(2.63)}$$

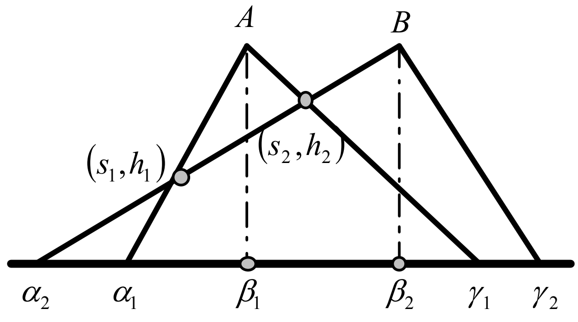

Fig. 2.14 Case 4 of Leng's work, $M(A)$ and $M(B)$ with two intersection point at (s_1, h_1) and (s_2, h_2).

Case 5:

In case 5, illustrated in Fig. 2.15, the membership functions $M(A)$ and $M(B)$ possess two intersection points at (s_1, h_1) and (s_2, h_2) between the right side of membership function $M(A)$ and both side of membership function $M(B)$, and the degree of similarity between membership functions $M(A)$ and $M(B)$, $S(A, B)$, can be calculated by:

$$M(A \cap B) = \frac{1}{2}\{(\beta_2 - \alpha_2)h_1^2 + (\gamma_2 - \beta_2)h_2^2 + (h_1 + h_2)[(\gamma_2 - \alpha_2) - (\beta_2 - \alpha_2)h_1 + (\gamma_2 - \beta_2)h_2]\} \quad \textbf{(2.64)}$$

$$\text{where } h_1 = \frac{\gamma_1 - \alpha_2}{\gamma_1 - \beta_1 + \beta_2 - \alpha_2}, \ h_2 = \frac{\gamma_1 - \gamma_2}{\gamma_1 - \beta_1 + \beta_2 - \gamma_2}, \quad \textbf{(2.65)}$$

$$\text{and } M(A \cup B) = \frac{1}{2}(\gamma_1 - \alpha_1 + \gamma_2 - \alpha_2) - M(A \cap B). \quad \textbf{(2.66)}$$

Moreover, if $S(A, B) \geq \lambda$, the new fuzzy set $C(\alpha, \beta, \gamma)$ merged by fuzzy sets

$A(\alpha_1, \beta_1, \gamma_1)$ and $B(\alpha_2, \beta_2, \gamma_2)$ can be calculated by:

$$\alpha = \alpha_1,\ \beta = \frac{1}{2}(\beta_1 + \beta_2),\ \gamma = \gamma_1. \qquad \textbf{(2.67)}$$

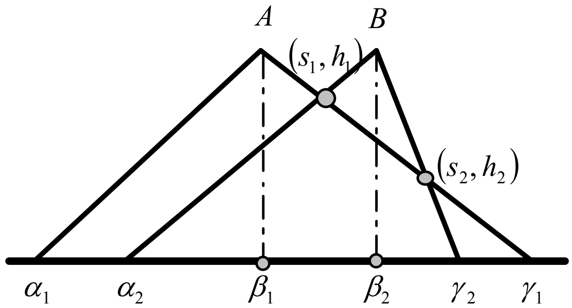

Fig. 2.15 Case 5 of Leng's work, $M(A)$ and $M(B)$ with two intersection point at (s_1, h_1) and (s_2, h_2).

Case 6:

In case 6, illustrated in Fig. 2.16, the membership functions $M(A)$ and $M(B)$ possess three intersection points at (s_1, h_1), (s_2, h_2) and (s_3, h_3), and the degree of similarity between membership functions $M(A)$ and $M(B)$, $S(A,B)$, can be calculated by:

$$\begin{aligned} &M(A \cap B) = \\ &\frac{1}{2}\{(\beta_1 - \alpha_1)h_1^2 + (\gamma_2 - \beta_2)h_2^2 + (h_1 + h_3)[(\beta_2 - \alpha_2)h_3 - (\alpha_1 + \alpha_2) - (\beta_1 - \alpha_1)h_1] \qquad \textbf{(2.68)} \\ &+ (h_2 + h_3)[(\gamma_2 - \alpha_1) - 2(\beta_1 - \alpha_1)h_1 - (\gamma_2 - \beta_2)h_2 - (\beta_2 - \alpha_2)h_3 - (\alpha_1 + \alpha_2)]\} \end{aligned}$$

where $h_1 = \dfrac{\alpha_1 - \alpha_2}{\beta_2 - \alpha_2 + \alpha_1 - \beta_1}$, $h_2 = \dfrac{\gamma_1 - \gamma_2}{\gamma_1 - \beta_1 + \beta_2 - \gamma_2}$, $h_3 = \dfrac{\gamma_1 - \alpha_2}{\gamma_1 - \beta_1 + \beta_2 - \gamma_2}$ **(2.69)**

and $M(A \cup B) = \dfrac{1}{2}(\gamma_1 - \alpha_1 + \gamma_2 - \alpha_2) - M(A \cap B)$. **(2.70)**

Moreover, if $S(A,B) \geq \lambda$, the new fuzzy set $C(\alpha, \beta, \gamma)$ merged by fuzzy sets

$A(\alpha_1, \beta_1, \gamma_1)$ and $B(\alpha_2, \beta_2, \gamma_2)$ can be calculated by:

$$\alpha = \alpha_2,\ \beta = \frac{1}{2}(\beta_1 + \beta_2),\ \gamma = \gamma_1. \tag{2.71}$$

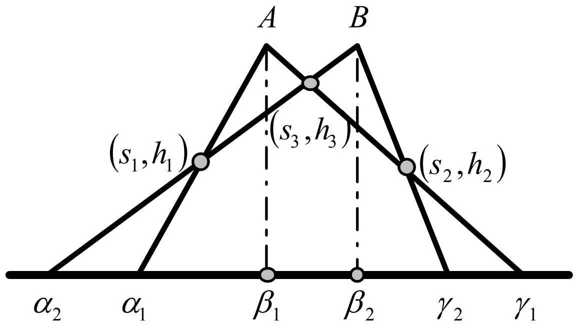

Fig. 2.16 Case 6 of Leng's work, $M(A)$ and $M(B)$ with three intersection point at (s_1, h_1), (s_2, h_2), and (s_3, h_3).

Moreover, in the above cases, the intersection point between two triangular fuzzy membership functions, (s_1, h_1), (s_2, h_2), and (s_3, h_3), can be calculated by straight-line equation.

2.6.7 Further Issue

There are certain reviews of earlier research works of similarity measure based on overlapped area mentioned in the Appendix I and Appendix II, including Chao's work (1996) and Jin's work (1999). According to the work proposed by Lin and Lee (1992), the Gaussian membership functions can be approximately transformed into the isosceles triangular membership functions for decreasing the computational cost or the sake of computational simplicity. In Jin *et al.* two Gaussian membership functions $A(c_1, w_1)$ and $B(c_2, w_2)$, where c_i and w_i are the centre and the width respectively, can be transformed into the triangular membership function as $a_i = c_i - w_i\sqrt{\pi}$, $b_i = c_i + w_i\sqrt{\pi}$ and $i = 1,2$. In addition to case 4, the transformed membership functions can be applied to case 1, 2, and 3, but in case 4 based on Gaussian membership functions this does not happen for the isosceles triangular membership functions.

2.7 Parameters Optimisation

Since the initial model of fuzzy system identification has been constructed, the parameters of the initial fuzzy model will be fine-tuned for refining the model as the final fuzzy model. Generally, in the later procedure of system identification, the parameters of the initial model will be fine-tuned by optimisation methods, such as the genetic algorithm, Newton's method, the Quasi-Newton's method, the gradient descent method, and so on. So far as the types of optimisation method are concerned, the methods can be categorised into two types, the global minimum type and the local minimum type. In global minimum methods, such as genetic algorithms, this method aims at finding the global minimum by iteratively picking out the best chromosome. In local minimum methods, such as the gradient descent method, this method aims at finding the local minimum point of the objective function by iterative steps.

Gradient Descent Method

The concept of the gradient descent method, also known as the steepest descent method, was proposed by Cauchy (1847). The gradient descent method is a first-order optimisation algorithm, and its purpose is to converge towards a local minimum of the objective function by taking proportional steps to the negative of the gradient of the function from the initial point. Therefore, the steps of the gradient descent method taking iterations to converge towards a local minimum can be described by

- Determine the descent direction, the negative of the gradient of the function from the initial point.

- Determine the size of the step.
- Keep taking iterations to converge towards a local minimum.

Further, the equation of gradient descent method can be represented by:

$$X_{n+1} = X_n - \gamma_n \nabla F(X_n) \ , \ n \geq 0 \ , \ \gamma > 0. \qquad \textbf{(2.72)}$$

where X is the point required to be tuned by the method, $\nabla F(X_n)$ is the gradient of objective function at X_n, and γ_n is the size of step. In equation 2.72, the objective function $F(X)$ should be absolutely defined and be differentiable, and the size of γ can be changed at every iteration.

The gradient descent method has the advantage of finding the local minimum on univariate, because it can effectively reach the local minimum with less computational cost. The gradient descent method also can find the local minimum on multivariate by following the steepest descent path, and that is why the gradient descent method is called the steepest descent method as well. Moreover, the determination of the size of the step γ is also a key point to determine the performance brought by the gradient descent method. In case the algorithm chooses a smaller size of step, the algorithm would take too long to converge with an inefficient performance. However, in case the algorithm chooses a bigger size of step, the algorithm would lead to the potentially bad results. Hence, the determination of the proper size of step γ can either keep the efficiency of converging or save the computational time for converging. Further, the disadvantage of the gradient descent method on converging towards a local minimum in common is the converging speed will become slower as more closing the local minimum.

2.8 Summary

Through this chapter, we have reviewed, fuzzy sets and membership functions, fuzzification and defuzzification, fuzzy system identification, literature reviews of learning and identification of fuzzy systems, cluster analysis, similarity-based pruning strategy, and parameters optimisation. For each, the primary roles and relevant research have been discussed in detail, and each is associated with the research in this thesis.

Chapter 3 A Three-Part Input-Output Clustering-Based Approach For Fuzzy System Identification

This chapter presents a three-part input-output clustering-based approach for fuzzy system identification. The motivation behind the work in this chapter considers obtaining complete and consistent fuzzy systems by integrating a variety of existing clustering properties without much complicated mechanism and computational resource. In order to acquire a mutually exclusive performance by constructing an effective fuzzy model as far as possible, this chapter presents a modular method to identify fuzzy systems based on a hybrid clustering-based technique. The determination of the proper number of clusters and the appropriate location of these clusters are one of primary considerations in constructing an effective fuzzy model. Due to the above reasons, a hybrid clustering algorithm concerning input, output, generalisation and specialisation is introduced in this chapter. The proposed clustering technique, a three-part input-output clustering algorithm (Lee & Zeng, 2010), integrates a variety of clustering features simultaneously, which usefully combines the advantages of several clustering methods in the domain, including the advantages of input clustering, output clustering, flat clustering, and hierarchical clustering, to identify the clustering problem. The primary purpose of the hybrid clustering-based approach aims at generating a sound initial fuzzy model by recognising the approximate location of clusters as far as possible. In order to realise the objectives of the research work in this chapter, the outline of the research is illustrated in Fig. 3.1.

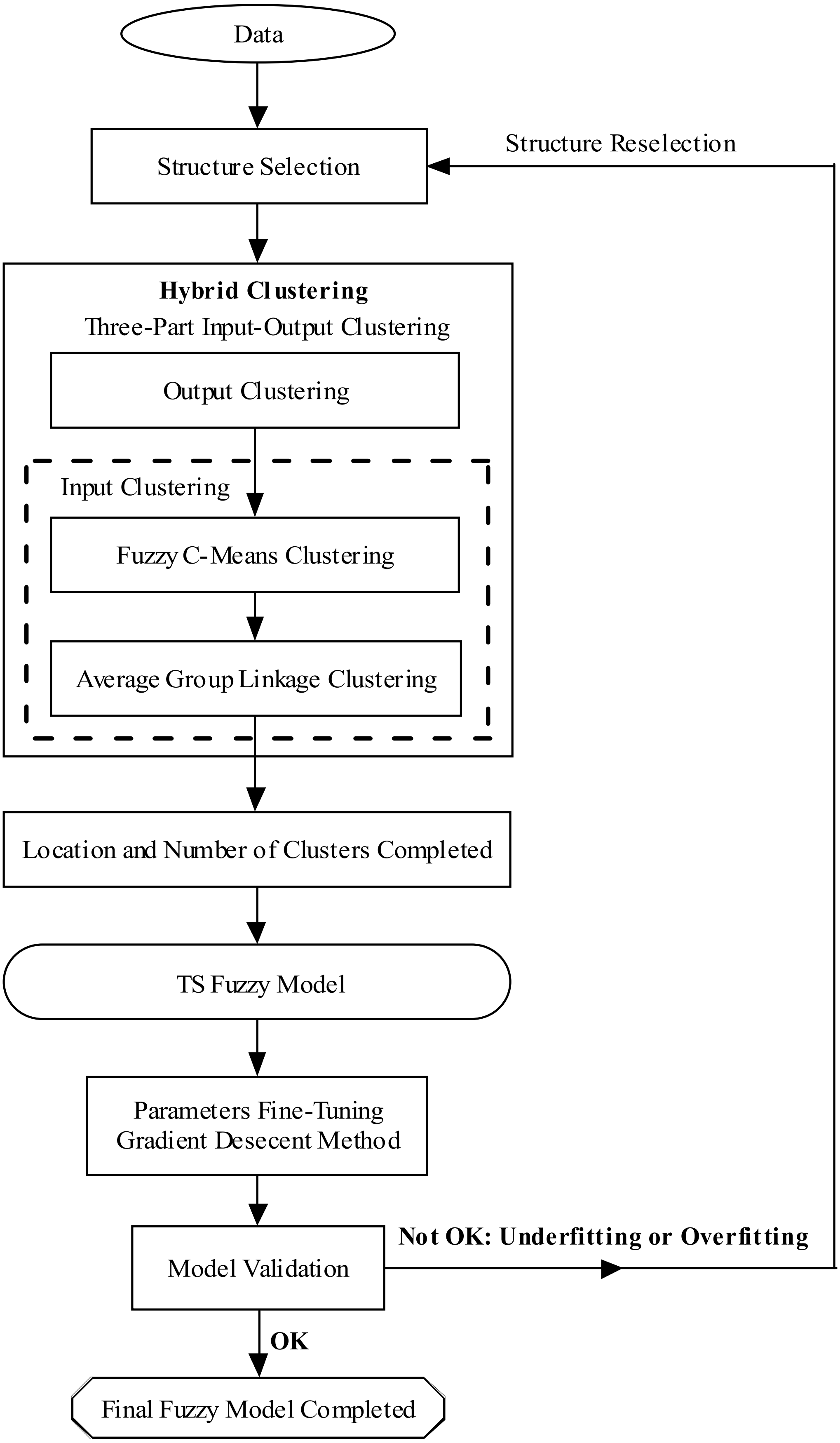

Fig. 3.1 Structure of the first research work

The rest of this chapter is organised as follows: a comprehensive introduction to this work is described in Section 3.1; in Section 3.2, the concept of hybrid clustering based on *IF-THEN-IF* is introduced; then the three-part input-output clustering technique of the proposed work is discussed in Section 3.3; in Section 3.4, the procedure of the proposed work is described in detail; in Section 3.5, the proposed work will be validated by simulation examples; and a conclusion is given in Section 3.6.

3.1 Introduction

Fuzzy systems have been applied at a variety of areas as a result of their representation capability. Due to good knowledge representation, the development of fuzzy system identification have been considered and proposed for the purpose of solving regression-type problems as well as function approximation problems. So far as the modelling of fuzzy system identification is concerned, there are two ways commonly implemented for the initial fuzzy model. One is to define fuzzy rules by grid partition, but grid-based partition approaches easily touch the problem of the “curse of dimensionality” with its large computational costs. Another one is to partition or cluster the observations or data first, and each resultant subset or cluster is a fuzzy rule in the initial fuzzy model. In other words, the predetermined partitioning method before constructing the fuzzy system is also a way of achieving a better performance in the overall identification of the fuzzy system, because the problem of the “curse of dimensionality” can be relieved in most cases in advance. As mentioned in section 1.4.3, an effective clustering technique can relieve the problem of the “curse of dimensionality”.

Due to the above reason, clustering-based approaches for fuzzy system identification have therefore been considered, such as Chiu’s method (1994), the method proposed by Kim *et al*. (1997), the method proposed by Delgado *et al*.

(1997), as well as the method proposed by Chen *et al*. (1998). The purpose of these is to develop a sound clustering technique evolved from the strength of the existing clustering algorithms to reinforce the performance of fuzzy system identification.

So far as a pure black box type of approach is concerned, it is not easy to develop a sound clustering technique concerning everything, because without prior knowledge, sometimes the performance is strictly limited under certain conditions. Generally, most schemes of fuzzy system identification start out using clustering techniques with the features of flat clustering or hierarchical clustering used to determine the partition of the initial model, such as the k-means clustering algorithm or the average linkage clustering algorithm, and finally end applying the methods of parameter optimisation to reinforce the accuracy of the approximation of the final model, such as the gradient descent method, Newton's method, or the genetic algorithm.

In order to effectively make a positive performance corresponding to the target system, a hybrid clustering algorithm integrating the advantages of the existing clustering techniques has been considered. A three-part input-output clustering technique has therefore been developed for enhancing the positive performance of fuzzy system identification and then carries out this as a foundation to capture useful knowledge in the target system. Hence, the hybrid clustering algorithm which has been proposed in this chapter originally considers input clustering and output clustering together, so that the proposed hybrid clustering method is diversely constructed by integrating the concept of input-output clustering and the technique of three-part clustering successively.

The framework of the proposed fuzzy model is constructed on an *IF-THEN IF* rule-base scheme by implementing a main module carrying sub-modules. The purpose of this paper is to try to develop a clustering-based approach for the identification of fuzzy systems based on a hybrid clustering-based technique with positive reliability. The structure of the proposed method is composed of three phases, comprising clustering determination, fuzzy rules generation, and parameters optimisation. In the rudimentary phase, the proposed method tries to determine the proper number of clusters and their appropriate location by using training data. In the second phase, the fuzzy system will be constructed by the resulting cluster determined by the first phase. In the final phase, parameters of the model will be fine-tuned by the gradient descent method.

3.2 Hybrid Clustering Based on *IF-THEN IF* Rule Base Scheme

Clustering algorithms have been widely used to solve pattern recognition problems and moreover effectively perform the reliability of fuzzy system identification by constructing an initial model, because the feature of clustering algorithms are usually used to determine the initial rules for fuzzy systems. Generally, most input clustering algorithms achieve a good performance on constructing the initial model of system identification. However, it is insufficient to only consider the inputs for clustering because the optimal number of clusters and the appropriate location of clusters cannot easily to be identified by considering the inputs only. The output for clustering has therefore been considered for a clustering algorithm to positively find the optimal number of clusters and their appropriate location. Generally, there are two ways of considering outputs for clustering. One is to incorporate the outputs into the training set, and then by combining the inputs x and outputs y with the weight w of the outputs as a new vector $z = (x, wy)$ for clustering (Gonzalez *et al.,* 2001). Another one is to cluster inputs x in the output context (Pedrycz, 1996; Pedrycz, 2002; Pedrycz, 2006; Leski, 2003). In order to develop a mutually exclusive high-reliability clustering technique, a hybrid clustering algorithm concerning input, output, generalisation and specialisation has been developed in this work. This hybrid clustering algorithm integrates the features of flat clustering and that of hierarchical clustering. In other words, the proposed clustering method does not only consider inputs and output, but also put focus on the generalisation performance and specialisation performance for clustering.

3.2.1 *IF-THEN IF* Rule Base Scheme

Basically, the structure of the proposed hybrid clustering algorithm is based on an *IF-THEN IF* rule base, and it originates from the concept of two-part clustering. In the entire structure, there is a main module comprising of primary clusters. Each primary cluster may carry sub-modules (sub-clusters) or none at all. According to the relationship of the distribution of the dataset, there are three types of distributions, including:

- ❖ Dataset with different output. As displayed in Fig. 3.2, dataset in cluster A possess different output with dataset in cluster B.
- ❖ Dataset with similar output but non-connective input. As displayed in Fig. 3.3, dataset in cluster A possess similar output but non-connective input with dataset in cluster B, and dataset in cluster C possess similar output but non-connective input with dataset in cluster D.
- ❖ Dataset with similar output and connective input. As displayed in Fig. 3.4, dataset in cluster A possess similar output and connective input, and dataset in cluster B possess similar output and connective input.

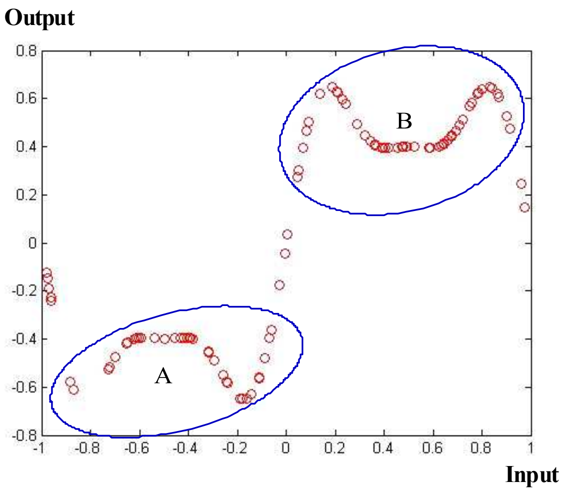

Fig. 3.2 Dataset with different output

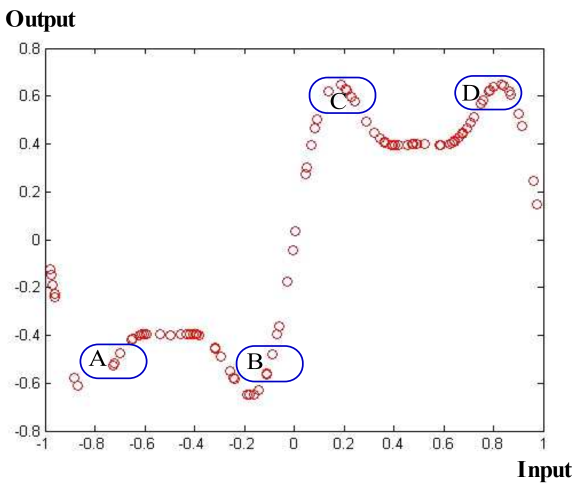

Fig. 3.3 Dataset with similar output but non-connective input

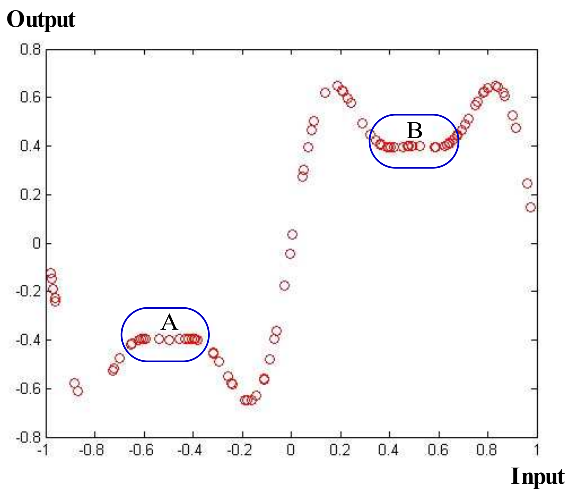

Fig. 3.4 Dataset with similar output and connective input

In order to identify these three types of dataset distributions clearly and then partition precisely, a hierarchical structure using the rule form of "*IF - THEN IF*", as illustrated in Fig. 3.5, has been used in the rule base as the main framework of the proposed clustering algorithm. The first *IF* concerns the primary rules of the main module, and the second *IF* concerns the respective sub-module and the secondary rule set. According to the scheme, the dataset will be considered for location at one of the primary clusters first, and then it will be decided whether the dataset will be considered for locating at a sub-module (sub-cluster) of the corresponding primary cluster or not. If the dataset is located at a high certainty region (high population region) in a primary cluster of the main module, then this dataset could be recognised as locating at a primary cluster only. However, if the dataset is located at a low certainty region (low population region) in a primary cluster of the main module, then this dataset could be recognised as locating at a sub-module of the corresponding primary cluster of the main module.

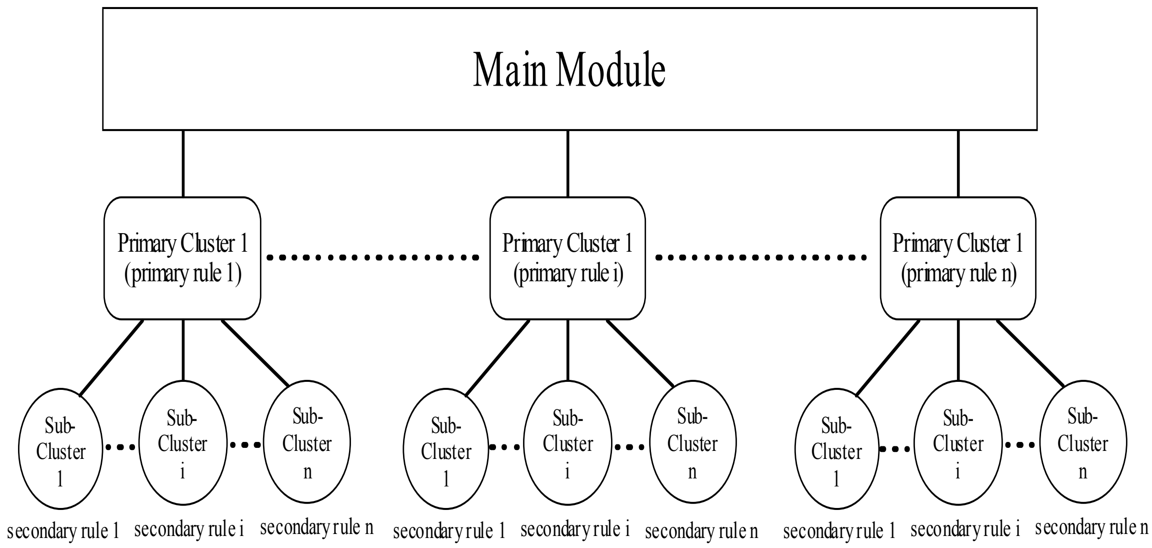

Fig. 3.5 IF-THEN IF Rule Base Scheme

The positive scheme of the hierarchical framework can perform these three types of dataset distributions effectively, and can be described as follows:

- ❖ Dataset with different output → Different primary clusters→ Different rules
- ❖ Dataset with similar output but non-connective input → The same primary cluster, but different sub-clusters → Different rules
- ❖ Dataset with similar output and connective input → The same primary cluster→ The same rule

3.2.2 Input-Output Clustering Technique

In order to carry out the *IF-THEN IF* rule base scheme by identifying the proper number of clusters and the probable location of these clusters approximately, the proposed hybrid clustering algorithm applies the concept of input-output clustering to support a good performance with a positive achievement. This is because the input-output clustering technique can effectively perform with good reliability as well as less uncertainty when processing a high-dimensional dataset in fuzzy system identification. In order to increase the accuracy and decrease the fuzziness for clustering together, the dataset will be initially partitioned based on the output space by output clustering, and the resultant clusters will be further partitioned by input clustering. Though the input-output clustering technique is much more complicated than the input clustering technique, the input-output clustering technique can discover a more precise cluster by carefully taking the relationship between the input variable and the output variable into consideration simultaneously. By satisfying a variety of requirements and conditions, the number of output constrictions for output clustering can be tuned properly. For instance, in case more partition is required, the number of output constrictions can be increased. In contrast, in cases where fewer partitions are required, the number of output constrictions can be decreased appropriately.

3.2.3 The Constriction for Output Clustering

The purpose of output clustering is to determine the number of output constrictions in the output space. Through determining the number of output constrictions in advance, the sophisticated relationship between the dataset being located at a variety of places before starting the process of clustering can be effectively decreased. Moreover, the output space is evenly partitioned based on hard interval partition, so we therefore considers $y = f(x_1, x_2, ..., x_n)$, where $y \in [\alpha, \beta]$. The output space is evenly partitioned by *m*, then: $y \in [\alpha_0, \alpha_1) \cup ... \cup [\alpha_{m-1}, \alpha_m]$, where $\alpha_0 = \alpha$ and $\alpha_m = \beta$. As displayed in Fig. 3.6, each interval based on the above definition can be recognized an output constriction respectively, and the training data are roughly grouped based on this output partition to obtain a set of clusters.

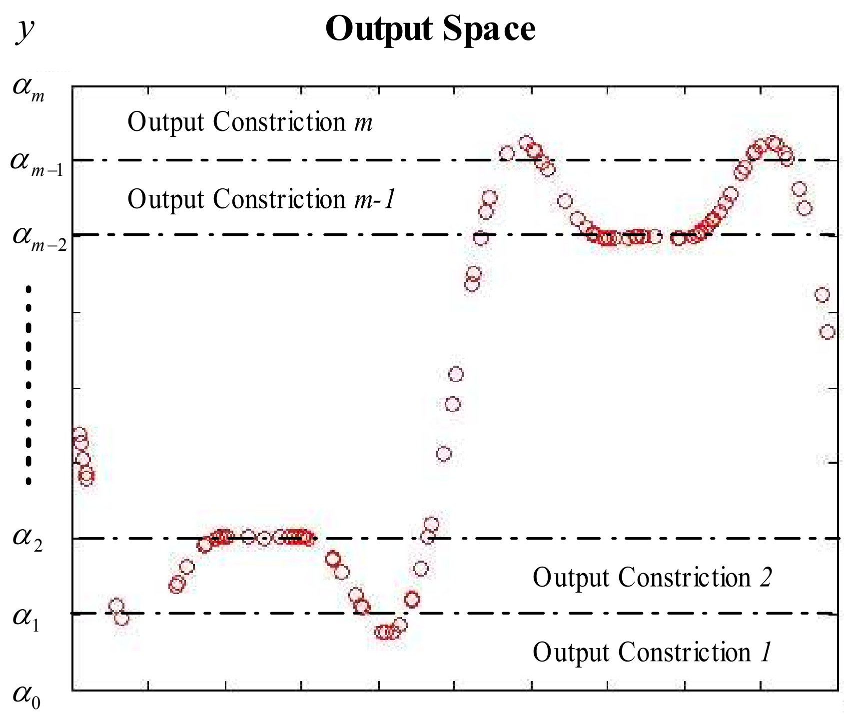

Fig. 3.6 Output Constrictions

3.3 Three-Part Input-Output Clustering Technique

The proposed hybrid clustering method can be regarded as a three-part input-output clustering algorithm for clearly identifying the three types of dataset distribution for system identification discussed above step by step, and the clustering technique consists of output clustering and input clustering. In addition to output clustering at the beginning, input clustering can be decomposed into two parts; flat clustering and hierarchical clustering. In flat clustering, the proposed method uses the fuzzy c-means clustering algorithm as the antecedent part of input clustering, as a result of generalisation performance. In hierarchical clustering, the proposed method uses the average group linkage clustering algorithm as the consequent part of the input clustering, as a result of specialisation performance.

The reason for using fuzzy c-means clustering as the second-part clustering is the properties of generalisation. Also, the reason for using the average group linkage clustering method as the third-part of the clustering is the properties of specialisation brought by hierarchical clustering and its average distance criterion. Basically, the average group linkage clustering algorithm applies average distance criterion, where the distance between the average values or mean vector of clusters will be computed. By other means, the average group linkage clustering method can perform effectively under the dataset are distributed by an average way. Moreover, as the clusters are generated by fuzzy c-means clustering algorithm, each resultant cluster can be assumed that contains the dataset distributed by an average way approximately. As an aspect of specialisation, the average group linkage clustering method is objective producing smaller clusters, which are helpful for data discovery for specialisation.

In principle, the proposed hybrid clustering method comprises of three steps, consisting of output constriction for output clustering, the fuzzy c-means algorithm for the first input clustering, and the average group linkage algorithm for the second input clustering. Basically, the first-part clustering is the rudimentary clustering based on output clustering, and aims to identify “dataset with different output”. The second-part clustering (the primary cluster of the main module) is determined by the fuzzy c-means clustering algorithm, and aims to identify “dataset with similar output and connective input”. Finally, the last-part of the clustering (the sub-cluster of the primary cluster) is determined by the average group linkage clustering algorithm, and aims to identify “dataset with similar output but non-connective input”. Hence, the proposed hybrid clustering method integrates the advantages of input clustering, output clustering, flat clustering, and hierarchical clustering, to effectively perform the identification of the clustering problem. According to the concepts discussed above, the regular steps of the proposed method can be definitely stated as follows:

- Step 1: Determine the rudimentary cluster (first-part clustering) by processing output clustering
- Step 2: Determine the primary clusters of the main module (second-part clustering) by processing the fuzzy c-means algorithm
- Step 3: Determine the sub-clusters of primary cluster (third-part clustering) by processing the average group linkage clustering algorithm.
- Initial fuzzy system identification model completed.
- Step 4: Tune the parameters of the initial fuzzy models by the gradient descent method.
- End: Hybrid clustering algorithm completed.

3.4 The Procedure of the Proposed Method

As described in the introduction, the method for modelling fuzzy system identification is based on a three-part input-output clustering technique, consisting of cluster determination, fuzzy rule generation and parameter optimisation.

Normally, in the phase of clustering determination, the first-part input-output clustering is processed by output clustering, and afterwards the second-part input-output clustering is processed by the fuzzy c-means clustering algorithm using the equations 2.38, 2.39, 2.40 and 2.41. Finally, the third-part input-output clustering is processed by the average group linkage clustering method using the equation 2.42.

In the phase of fuzzy rule generation, fuzzy rules are generated by triangular membership functions based on the resultant clusters generated by the phase of cluster determination using the equations 2.9, 2.10, 2.11, 2.12, 2.13, 2.14 and 2.15, with the fuzzy modelling based on the TS fuzzy model. The reason for using the triangular membership functions in this work is their transparency and interpretability as a result of its finite partition range of input space.

As the fuzzy system is modelled by the symmetric triangular membership functions above, there are three types of parameters required to be tuned by the gradient descent method. The parameters of the triangular membership function in the antecedent part of the fuzzy system are the centre a_{ij}, the width b_{ij} of the triangular membership function, and those of the triangular membership function in the consequent part of the fuzzy system are the output variables w_i of the output membership function.

The target of the gradient descent method aims at minimising the square error, and which can be represented by:

$$E = \frac{1}{2} \times (y - y^r)^2 \quad \textbf{(3.1)}$$

where y is the system output of fuzzy reasoning and y^r is the desirable output. According to equations 2.9, 2.10, and 2.13, the formulas for fine tuning the parameters of the fuzzy model (Nomura et al., 1991; Nomura et al., 1992; Guely & Siarry, 1993) evolved by the gradient descent method (equation 2.75) can be represented by:

$$a_{ij}(n+1) = a_{ij}(n) - \gamma \cdot \frac{\partial E}{\partial a_{ij}} \quad \textbf{(3.2)}$$

$$b_{ij}(n+1) = b_{ij}(n) - \gamma \cdot \frac{\partial E}{\partial b_{ij}} \quad \textbf{(3.3)}$$

$$w_i(n+1) = w_i(n) - \gamma \cdot \frac{\partial E}{\partial w_i} \quad \textbf{(3.4)}$$

where
$$\frac{\partial E}{\partial a_{ij}} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial \mu_i} \cdot \frac{\partial \mu_i}{\partial A_{ij}} \cdot \frac{\partial A_{ij}}{\partial a_{ij}} \quad \textbf{(3.5)}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial \mu_i} \cdot \frac{\partial \mu_i}{\partial A_{ij}} \cdot \frac{\partial A_{ij}}{\partial b_{ij}} \quad \textbf{(3.6)}$$

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial w_i} \quad \textbf{(3.7)}$$

The partial derivatives of equations 3.5, 3.6, and 3.7 can be obtained by:

$$\frac{\partial E}{\partial y} = \left(y - y^r\right) \quad \textbf{(3.8)}$$

$$\frac{\partial y}{\partial w_i} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \tag{3.9}$$

$$\frac{\partial y}{\partial \mu_i} = \frac{(w_i - y)}{\sum_{i=1}^{n} \mu_i} \tag{3.10}$$

$$\frac{\partial \mu_i}{\partial A_{ij}} = \frac{\mu_i}{A_{ij}(x_j)} \tag{3.11}$$

$$\frac{\partial A_{ij}}{\partial a_{ij}} = \frac{2 \cdot \operatorname{sgn}(x_j - a_{ij})}{b_{ij}} \tag{3.12}$$

$$\frac{\partial A_{ij}}{\partial b_{ij}} = \frac{(1 - A_{ij}(x_j))}{b_{ij}} \tag{3.13}$$

Therefore, the gradients of the objective function ($\frac{\partial E}{\partial a_{ij}}, \frac{\partial E}{\partial b_{ij}}, \frac{\partial E}{\partial w_i}$) can be derived as follows:

$$\frac{\partial E}{\partial a_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \cdot (w_i - y) \cdot \operatorname{sgn}(x_j - a_{ij}) \cdot \frac{2}{b_{ij} \cdot A_{ij}(x_j)} \tag{3.14}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \cdot (w_i - y) \cdot \frac{1 - A_{ij}(x_j)}{A_{ij}(x_j)} \cdot \frac{1}{b_{ij}} \tag{3.15}$$

$$\frac{\partial E}{\partial w_i} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \tag{3.16}$$

In equation 3.12 and 3.14, the *signum function* is denoted by *sgn*, and which can be defined by:

$$\operatorname{sgn}(x) = \begin{Bmatrix} -1, x < 0 \\ 0, x = 0 \\ 1, x > 0 \end{Bmatrix} \tag{3.17}$$

The algorithm converges until either reaching the local minimum of the function or the stopping criterion $\left|y(t) - y^r\right| \le \varepsilon$ is satisfied.

As the fuzzy system is modelled by asymmetric triangular membership functions, there are four types of parameters required to be tuned by the gradient descent method. In addition to the centroid a_{ij} and the output variable w_i, there are two widths b_{ij} and c_{ij} to be tuned. Therefore, the second width c_{ij} can be defined by gradient descent method using:

$$\frac{\partial E}{\partial c_{ij}} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial \mu_i} \cdot \frac{\partial \mu_i}{\partial A_{ij}} \cdot \frac{\partial A_{ij}}{\partial c_{ij}} \tag{3.18}$$

The partial derivatives of equations 3.5, 3.6, and 3.18 for an asymmetric triangular membership function based on equations 2.14 and 2.15 can be obtained by:

$a_{ij} - b_{ij} \le x_j \le a_{ij}$:

$$\frac{\partial A_{ij}}{\partial a_{ij}} = \frac{-1}{b_{ij}}, \quad \frac{\partial A_{ij}}{\partial b_{ij}} = \frac{\left(a_{ij} - x_j\right)}{b_{ij}^{\,2}}, \quad \frac{\partial A_{ij}}{\partial c_{ij}} = 0 \tag{3.19}$$

$a_{ij} \le x_j \le a_{ij} + c_{ij}$:

$$\frac{\partial A_{ij}}{\partial a_{ij}} = \frac{1}{c_{ij}}, \quad \frac{\partial A_{ij}}{\partial b_{ij}} = 0, \quad \frac{\partial A_{ij}}{\partial c_{ij}} = \frac{\left(x_j - a_{ij}\right)}{c_{ij}^{\,2}} \tag{3.20}$$

Therefore, the gradients of the objective function ($\frac{\partial E}{\partial a_{ij}}, \frac{\partial E}{\partial b_{ij}}, \frac{\partial E}{\partial w_i}$) for an asymmetric triangular membership function can be derived as follows:

$a_{ij} - b_{ij} \le x_j \le a_{ij}$:

$$\frac{\partial E}{\partial a_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \cdot (w_i - y) \cdot \frac{-1}{b_{ij} \cdot A_{ij}(x_j)} \tag{3.21}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \cdot (w_i - y) \cdot \frac{1}{A_{ij}(x_j)} \cdot \frac{(a_{ij} - x_j)}{b_{ij}^{\ 2}} \tag{3.22}$$

$$\frac{\partial E}{\partial c_{ij}} = 0 \tag{3.23}$$

$a_{ij} \le x_j \le a_{ij} + c_{ij}$:

$$\frac{\partial E}{\partial a_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \cdot (w_i - y) \cdot \frac{1}{c_{ij} \cdot A_{ij}(x_j)} \tag{3.24}$$

$$\frac{\partial E}{\partial b_{ij}} = 0 \tag{3.25}$$

$$\frac{\partial E}{\partial c_{ij}} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \cdot (y - y^r) \cdot (w_i - y) \cdot \frac{1}{A_{ij}(x_j)} \cdot \frac{(x_j - a_{ij})}{c_{ij}^{\ 2}} \tag{3.26}$$

In both $a_{ij} - b_{ij} \le x_j \le a_{ij}$ and $a_{ij} \le x_j \le a_{ij} + c_{ij}$, the partial derivatives of $\frac{\partial E}{\partial w_i}$ is the same with using equation 3.16. The algorithm converges until either reaching the local minimum of the function or the stopping criterion $\left| y(t) - y^r \right| \le \varepsilon$ is satisfied.

Generally, the gradient descent method is the most used method for parameter optimisation in fuzzy system identification, and the limitation of the gradient descent method has already been described in section 2.7. In addition the main reason for using the gradient descent method in this work is its lower computational cost compared to other optimisation methods for fuzzy modelling. There are other optimisation methods for parameter optimisation, such as Newton's method or a genetic algorithm. The reason for not using Newton's method is its huge computational cost, and the reason of not using a genetic algorithm is its huge computational time based on a mechanism of global minimum type as mentioned in section 2.7.

Initially, the gradient descent method can be processed very fast as a result of the linear shape of the triangular membership function. However, the processing of the gradient descent method slows as it reaches the peak of the triangular membership function. Though the triangular membership function is not smooth at the peak, the gradient descent method still can reach the probable local minimum by tracking back. This means that the gradient descent method can track back as the point passes over the minimum point to another side of the shape of the triangular membership function. The movement, is like throwing a small bead into a triangular cup, where the bead can finally reach the bottom of the triangular cup with the movement of a pendulum.

3.5 Simulation

In order to validate the reliability of the performance of the proposed method, comparisons of the simulation results between other methods and the proposed method are described in the following section by one three-input nonlinear function approximation, two classical benchmark problems, and one nonlinear dynamic system identification. Further, the experiments are only one run as well as not cross validated, and that the comparisons with others are 'general'. However, the proposed algorithm has been tested on many datasets, which confirms the positive results of the proposed algorithm.

3.5.1 Three-Input Nonlinear Function Approximation

The example of simulation can be generated by:

$$t = (1 + x^{0.5} + y^{-1} + z^{-1.5})^2 \tag{3.27}$$

where $x \in [1,6]$, $y \in [1,6]$, $z \in [1,6]$. 216 training data as well as 125 testing data are randomly generated from the above domain, The accuracy of the proposed method will be measured by MAPE, and can be represented by:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i^{\ r} - y_i}{y_i^{\ r}}\right| \times 100\% \tag{3.28}$$

where n is the number of training data, y_i is the system output of fuzzy reasoning and y_i^r is the desirable output.

The comparisons of the accuracy in MAPE between the proposed method and other methods are reported in Table 3.1. From Table 3.1, the proposed method holds a better MAPE of either training example or testing example than those of other methods in the comparison. Further, the proposed method uses fewer rules and parameters than other methods in the comparison as well.

Methods	*Rules (neurons)*	*Parameters*	*MAPE of Training*	*MAPE of Testing*
ANFIS (Jang, 1993)	8	50	0.0430	1.0660
GDFNN (Wu et. al,, 2001)	10	64	2.1100	1.5400
SOFNN (Leng *et al*., 2005)	9	60	1.1380	1.1244
Proposed Work – 1 By Affine TS Fuzzy Model	8	50	0.0015	0.0018

TABLE 3.1 COMPARISON OF RESULTS BASED ON (3.27)

3.5.2 Benchmark Problems: Mackey-Glass time-series prediction

The proposed method has been validated on a benchmark problem: the Mackey-Glass chaotic time series prediction. The chaotic time series is generated from the Mackey-Glass differential delay equation defined by (Chiu, 1994; Kasabov & Song, 2002; Angelov & Filev, 2004)

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t). \tag{3.29}$$

where $x(0)=1.2$, $x(t)=0$ for $t < 0$, and $\tau = 17$.

The target of the simulation aims at using the past values of x to predict the future value of x. Therefore, the prediction task is to predict the value $x(t+85)$ from the input vector $[x(t-18); x(t-12); x(t-6); x(t)]$ for any value of the time t. In this simulation, 3000 data points $t \in [201.3200]$ are generated for training, and 500 data points $t \in [5001.5500]$ are generated for testing.

Further, the accuracy of the proposed method will be measured by the non-dimensional error index (NDEI) defined as the ratio of the root mean square error (RMSE) over the standard deviation of the target data, and which can be represented by

$$NDEI = \frac{\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^r)\right]^{\frac{1}{2}}}{\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y_i})\right]^{\frac{1}{2}}} \tag{3.30}$$

where n is the number of training data, y is the system output of fuzzy reasoning, y^r is the desirable output, and $\overline{y}$ is the mean of y^r.

The comparisons of the accuracy in NDEI between the proposed method and other methods are reported in Table 3.2. From Table 3.2, the proposed method attains a better NDEI of the testing example than that of the other methods in the comparison. Further, the proposed method uses fewer rules than other methods' in the comparison as well.

Method	*Rules (neurons)*	*NDEI for Testing*
DENFIS (Kasabov & Song, 2002)	58	0.276
DENFIS (Kasabov & Song, 2002)	883	0.033
RAN (Platt, 1991)	113	0.373
ESOM (Deng & Kasabov, 2000)	114	0.32
ESOM (Deng & Kasabov, 2000)	1000	0.044
EFuNN (Kasabov, 1998)	193	0.401
EFuNN (Kasabov, 1998)	1125	0.094
Neural gas (Fritze, 1995)	1000	0.062
eTS (Angelov & Filev, 2004)	113	0.095
Simple_eTS (Angelov & Filev, 2005)	11	0.394
exTS (Angelov & Zhou, 2006)	10	0.331
exTS (Angelov & Zhou, 2006)	9	0.361
SAFIS (Rong *et al.*, 2006)	6	0.376
Proposed Work – 1 By Zero-Order TS Fuzzy Model	16	0.1008
Proposed Work – 1 By Affine TS Fuzzy Model	81	0.0254

TABLE 3.2 COMPARISON OF RESULTS BASED ON (3.29)

3.5.3 Benchmark Problem: Box-Jenkins Gas Furnace

The well-known Box–Jenkins gas data (Box & Jenkins, 1976) has 296 data pairs $\{(u(t), y(t)) \mid t\in[1, 296]\}$, where $y(t)$ is the output CO_2 concentration and $u(t)$ is the input gas flow rate. Meanwhile, the series-parallel model can be obtained as:

$$y(t)=f(y(t-1),u(t-4)). \qquad \textbf{(3.31)}$$

In this simulation, 200 data samples were randomly chosen for training, and 90 data samples were randomly chosen for testing. Further, the accuracy of the proposed method was measured by NDEI.

Meanwhile, the comparisons of the accuracy in MSE as well as NDEI between the proposed method and other methods are reported in Table 3.3. From Table 3.3, the proposed method obtains better MSE for training as well as NDEI for testing than those of the other methods in the comparison. Further, the proposed method uses fewer rules than the other methods in the comparison as well.

Method	*Rules (neurons)*	*MSE for Training*	*NDEI for Testing*
Tong's (Tong, 1979)	19	0.469	N/A
Pedrycz's (Pedrycz, 1984)	81	0.320	N/A
Xu's (Xu & Lu, 1987)	25	0.328	N/A
Li's (Li & Mukaidono, 1995)	6	0.178	N/A
Wang's (Wang & Langari, 1996)	5	0.158	N/A
Cheng's (Cheng & Hsieh, 2001)	4	0.146	N/A
SOFNN-I (Leng *et al.*, 2005)	4	0.131	N/A
eTS (Angelov & Filev, 2004)	5	N/A	0.30571
Simple_eTS (Angelov & Filev, 2005)	3	N/A	0.30041
Proposed Work – 1 By Affine TS Fuzzy Model	4	0.074	0.24958

TABLE 3.3 COMPARISON OF RESULTS BASED ON (3.31)

3.5.4 Non-Linear Dynamic System Identification

Finally, the non-linear dynamic system identification (Wang & Yen, 1999) has been used for validating the performance of the proposed method, and a series-parallel model is performed to identify this system as:

$$y(k) = g(y(k-1), y(k-2)) + u(k-1) \quad \textbf{(3.32)}$$

where $g(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)[y(k-1)-0.5]}{1+y^2(k-1)+y^2(k-2)}$, $u(k) = \sin\left(\frac{2\pi t}{25}\right)$, $y(0) = y(1) = 0$.

5000 data are created for training, and 300 data are created for testing.

The comparisons of the accuracy in RMSE between the proposed method and other methods are reported in Table 3.4. From Table 3.4, the proposed method holds a better RMSE of either training example or testing example than those of the other methods in the comparison. Further, the proposed method uses fewer rules than other methods' in the comparison as well.

Method	*Rules*	*RMSE for Training*	*RMSE for Testing*
SAFIS (Rong *et al.*, 2006)	17	0.0539	0.0221
M-RAN (Lu *et al.*, 1997)	22	0.0371	0.0271
RANEKF Kadirkamanathan & Niranjan, 1993)	35	0.0273	0.0297
Simple_eTS (Angelov & Filev, 2005)	22	0.0528	0.0225
eTS (Angelov & Filev, 2004)	49	0.0292	0.0212
HA (Wang & Yen, 1999)	28	0.0182	0.0244
ICLA (Wang, *et al.*, 2008)	8	0.0321	0.0318
ICLA (Wang, *et al.*, 2008)	20	0.0012	0.0007
Proposed Work – 1 By Affine TS Fuzzy Model	8	0.0028	0.0022

TABLE 3.4 COMPARISON OF RESULTS BASED ON (3.32)

3.6 Conclusion

This chapter tries to present a modular method to identify fuzzy systems based on a hybrid clustering-based technique. A hybrid clustering algorithm concerning input, output, generalisation and specialisation has hence been introduced in this article. The proposed clustering technique, a three-part input-output clustering algorithm, integrates a variety of clustering features simultaneously, including the advantages of input clustering, output clustering, flat clustering, and hierarchical clustering, to effectively perform the identification of the clustering problem. According to the advantages given above, the proposed clustering algorithm is capable of determining the proper number of clusters and the appropriate location of these clusters by considering both inputs and outputs. Due to the capabilities and the simulation results above, the proposed method possesses the potential for slightly easing the problem of the "curse of dimensionality", and even provides positive reliability for fuzzy system identification.

Further, the contribution brought by the work in this chapter can be described as follows. Three-part input-output clustering-based approach to fuzzy system identification can positively discover the proper number of clusters and their appropriate location by integrating a variety of existing clustering properties effectively. Also, the contribution is developed based on Occam's Razor, because it applies the basic features of existing clustering algorithms to carry out the optimal performance without much complicated mechanisms and computational resource.

Chapter 4 A Similarity-Based Learning Algorithm For Fuzzy System Identification With A Two-Layer Optimisation Scheme

Since a three-part input-output clustering-based approach to fuzzy system identification has been introduced in the previous chapter, an evolution of this work has therefore occurred for acquiring a more complete, consistent and compact fuzzy system. The motivation behind the work in this chapter considers obtaining compact, complete and consistent fuzzy systems by applying a similarity-based pruning strategy as well as developing a positive optimisation scheme without much complicated mechanism and computational resource. Therefore, two ways to improve the work proposed in chapter 3 have been considered in the extended work in this chapter, including the pruning strategy for the simplification of the fuzzy rule base and the optimisation scheme for the fine tuning parameters of fuzzy systems. So far as the pruning strategy is concerned, the purpose of which is to refine the rule base by processing similarity analysis of fuzzy sets, fuzzy numbers, membership functions or fuzzy rules, and to merge similar pairs in a reliable way. Through the pruning strategy, the complete rules can be kept and the redundant rules can be reduced in the rule base of the fuzzy system. The optimisation scheme can be regarded as a two-layer parameters optimisation in this extended work, because the parameters of the initial fuzzy model have been fine tuned by a two phase gradation layer. Hence, in this chapter, the extended work primarily focuses on applying the pruning strategy and the optimisation scheme to refine the initial fuzzy model to the final fuzzy model. The outline of the extended work in the chapter is illustrated in Fig. 4.1.

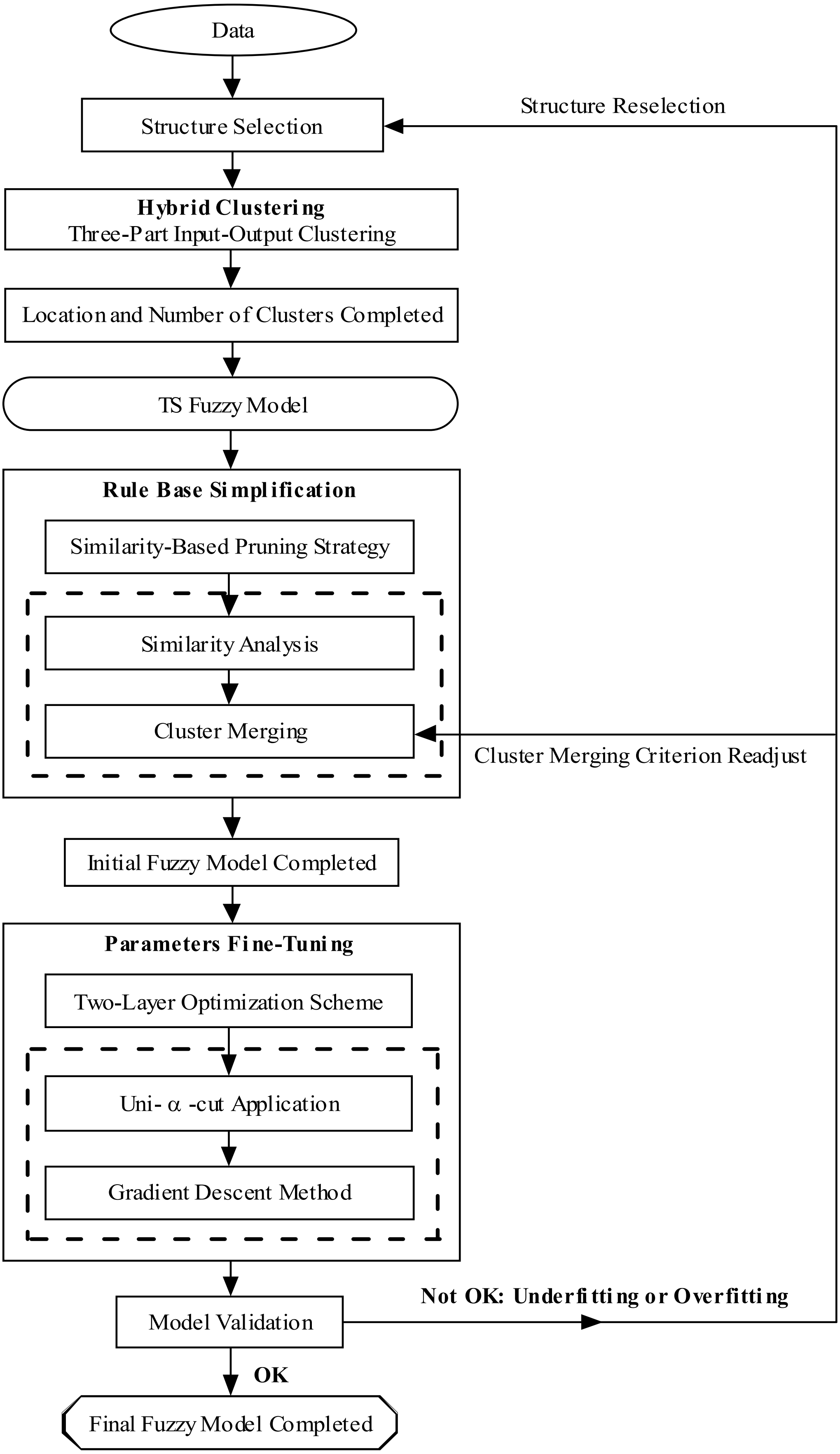

Fig. 4.1 Structure of the extended research work

The rest of this chapter is organised as follows: a comprehensive introduction of the proposed work is described in Section 4.1; in Section 4.2, the concept of the similarity-based pruning strategy and the application of an optimisation scheme in the extended work are discussed; then a similarity-based learning algorithm to fuzzy system identification with a two-layer optimisation scheme is introduced in Section 4.3; in Section 4.4, the procedure of the proposed work is described in detail; in Section 4.5, the proposed work will be validated by simulation examples; a conclusion is given in Section 4.6.

4.1 Introduction

In addition to the advantage of a good learning ability, one of the unique features of a fuzzy system is its knowledge presentation, and a compact fuzzy system makes this feature more obvious as a result of the well-interpretability. As a compact and transparent fuzzy system is also one of the increasingly important issues for fuzzy system identification, lots of research has been undertaken. Generally, most of these methods apply a variety of pruning strategies to make the optimal simplification of the fuzzy rule base. The simplification of the fuzzy rule base consists of two stages. The first stage is the similarity analysis between fuzzy sets, fuzzy membership functions, or fuzzy numbers, the purpose being to distinguish between similar pairs and dissimilar pairs. The second stage is to merge similar pairs into clusters, and to refine the fuzzy rule base. Briefly, the policy of the pruning strategy is that dissimilar clusters will be distinguished and similar clusters will be merged in an intelligent way.

As the initial fuzzy model is constructed, the size of the obtained structure of the model is dependent on the performance of the predetermined clustering algorithm. In cases where the predetermined clustering algorithm performs, the size of the obtained structure of the initial fuzzy model is much smaller. In contrast, where the predetermined clustering algorithm performs ineffectively, the size of the obtained structure of the initial fuzzy model is much bigger. In other words, a larger size of the structure of the fuzzy model is a result of an incomplete performance by the predetermined clustering algorithm. As mentioned before, a larger sized structure of the fuzzy model may be due to a large number of redundant rules in the fuzzy system. Due to the disadvantages brought about by the incomplete performance of the predetermined clustering algorithm, pruning strategies have been investigated to simplify the structure for refining the rule base. The purpose of most pruning strategies is to keep the useful rules and reduce the useless rules by positively pruning the structure as far as possible.

The behaviour of the proposed pruning strategy in the extended work consists of two stages, including the simplification of the size of the obtained structure and the merging of similar pairs. The primary task of the first stage of the pruning strategy aims at how to simplify the structure of the initial fuzzy model based on similarity analysis, and that of the second stage of the pruning strategy aims at how to merge the similar fuzzy rules resulting from the simplification of the structure of the initial fuzzy model based on the generalisation consideration. This is one of the effective ways to acquire a compact fuzzy system by refining the rule base with the similarity-based pruning strategy. The is because the size of the structure of the initial fuzzy model has been simplified, the highly similar clusters have been merged, the number of redundant rules have been reduced and similar rules have been integrated in the rule base of the fuzzy system by the positive performance of the similarity-based pruning strategy.

In addition to the similarity-based pruning strategy, this extended work also develops a two-layer optimisation scheme to refine the initial fuzzy model by fine tuning the parameters of the initial fuzzy model. Basically, the procedure of the two-layer optimisation scheme consists of the application of uni-α-cuts of fuzzy sets and the gradient descent method. In the first layer of the optimisation scheme, the application of uni-α-cuts of fuzzy sets, roughly tunes the parameters of the initial fuzzy model by discovering the most proper degree of uni-α-cuts of fuzzy sets of the initial fuzzy model. Consequently, the second layer of the optimisation scheme, the gradient descent method, tunes the parameters of the initial fuzzy model iteratively until the algorithm converges or the stopping criterion has been reached.

Further, the framework of the fuzzy model in chapter 3 is constructed on the *IF-THEN IF* rule-base scheme based on a three-part input-output clustering-based approach. The purpose of the extended work in this chapter is to apply the similarity-based pruning strategy to merge highly similar sub-clusters and implement a two-layer optimisation scheme to reinforce the positive performance of the final fuzzy model.

4.2 Preliminaries

To attain a compact and complete fuzzy system, the similarity-based pruning strategy has been widely used to simplify the structure of fuzzy systems by reducing the complexity of the initial fuzzy model or the obtained system. An effective pruning strategy makes a fuzzy system more compact and transparent, and a similarity-based pruning strategy does not only increase the transparency of the fuzzy system but also decreases the complexity of the obtained system.

Lots of similarity-based pruning strategies (Pappis & Karacapilidis, 1993; Chen & Chen, 1995; Lin, 1995; Chao *et al*., 1996; Hsieh & Chen, 1999; Lee, 1999; Jin *et al*., 1999) for mitigating overlap among membership functions have been proposed and were described in section 2.6. It is obvious that similarity analysis plays a significant role in the pruning strategy, and it reduces the complexity of the initial model for fuzzy system identification as well.

To make the optimal balance between simplicity and accuracy of the obtained system, parameters optimisation is required for obtaining the final fuzzy model by fine tuning parameters of the fuzzy model. The α-cuts of fuzzy sets is also one of the essential concepts in the parameter optimisation scheme in this work, because it is useful to discover the most appropriate fuzzy configuration to represent an unknown relationship. There are several papers that apply the concept of the α-cuts of fuzzy sets to support a positive performance in fuzzy modelling. Wu and Chen's work (1999) proposed a fuzzy learning algorithm based on the α-cuts of equivalence relations and the α-cuts of fuzzy sets to construct membership functions of the input variables and the output variables of fuzzy rules and to induce the fuzzy rules from the numerical training data set. In Lee and Zeng's works (2008) as well as Lee and Wang's work (2009), the concept of the α-cuts of the fuzzy set can be regarded as determining the proper degree of sensitivity for the system under observation.

The Implementation of α-Cuts of Fuzzy Sets

Through discovering the optimal global degree of the α-cuts of fuzzy sets, the parameters of the triangular membership functions in the fuzzy system can be fine tuned. The triplet of the triangular fuzzy set $(\textit{left vertex}, \textit{centre}, \textit{right vertex})$ can be defined as (a_i, b_i, c_i), and the calculation for constructing the triangular membership function based on α-cuts in Lee's works can be presented by (Lee & Zeng, 2008; Lee & Wang, 2009):

$$b_i = \frac{x_{i,\min} + x_{i,\max}}{2}, \qquad a_i = b_i - \frac{b_i - x_{i,\min}}{\alpha}, \qquad c_i = b_i + \frac{x_{i,\max} - b_i}{\alpha} \quad . \quad \textbf{(4.1)}$$

where $x_{\min}$ is the value of the minimum unit of each partition or cluster , $x_{\max}$ is the value of the maximum unit of each partition or cluster, α represents the degree of the α-cuts of fuzzy sets, and $\alpha \in (0,1]$.

4.3 A Similarity-Based Fuzzy Learning Algorithm with Two-Layer Optimisation Scheme

Fuzzy system identification generally consists of clustering for constructing the initial fuzzy model and parameter optimisation for finalising the final fuzzy model. As mentioned earlier, this work is an evolutionary work developed from the structure of the three-part input-output clustering method proposed in chapter 4. In addition to the original structure of the three-part input-output clustering method, a similarity-based pruning strategy is applied to refine the fuzzy rule base for the purpose of obtaining a compact and complete fuzzy model, and a two-layer optimisation scheme is developed for parameter optimisation for the purpose of obtaining a precise fuzzy model with a positive generalisation performance.

So far as the similarity-based pruning strategy is concerned, the extended work in this chapter adopts the similarity-based pruning strategy proposed by Leng *et al.* (2009) as the strategy for the simplification of the fuzzy rule base. The reason for this is the similarity-based pruning strategy proposed by Leng *et al.* is the most complete for the similarity analysis of triangular fuzzy membership functions or triangular fuzzy sets. All six cases proposed by Leng *et al.* have already covered all probable similarity relationships between two triangular fuzzy membership functions or triangular fuzzy sets. The criteria for combining two triangular fuzzy membership functions or triangular fuzzy sets with a high similarity in equations 2.55, 2.59, 2.63, 2.67 and 2.71, essentially takes the generalisation performance of fuzzy systems into account, because the *a priori* knowledge is totally unavailable in black box type modelling.

So far as the parameter optimisation is concerned, most of the fuzzy system identification methods apply single parameters optimisation. Basically, the single parameter optimisation can perform well based on an effective initial fuzzy model. However, it is insufficient to only consider single parameters optimisation in the optimisation scheme, especially for local minimum optimisation method. Therefore, the reasons for developing a two-layer optimisation scheme can be described as follows:

- ❖ A much better local minimum may be acquired in potential. For instance, as shown in Fig. 4.2, local minimum B is better than local minimum A.
- ❖ The reinforcement for the initial fuzzy model

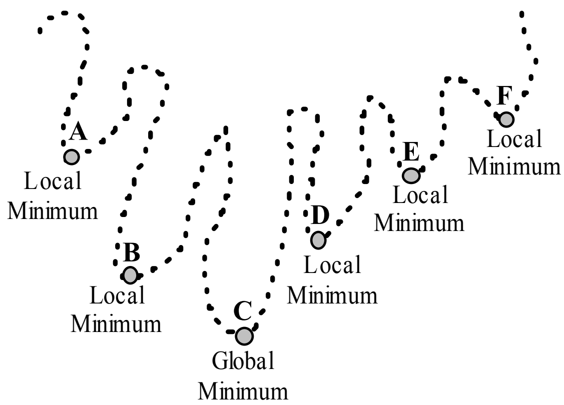

Fig. 4.2 Local Minimums

Due to the above reasons, a two-layer optimisation scheme for parameter optimisation has therefore been developed. The two-layer optimisation scheme consists of the application of uni-α-cuts of fuzzy sets and the gradient descent method. The first-layer is implemented by the application of uni-α-cuts of fuzzy sets, the purpose of which aims at achieving a rough parameter optimisation and to discover a better starting point for the gradient descent method. Generally, a many layered optimisation scheme results in the problem of overfitting, and may lead to inconsistent results for the validation as well. To improve the performance and avoid the problem of overfitting, the concept of uni-α-cuts of fuzzy sets has therefore applied been in the optimisation scheme.

4.3.1 Similarity-Based Pruning Strategy

The structure of the similarity-based pruning strategy consists of two phases. The first phase is similarity analysis between triangular fuzzy numbers, triangular fuzzy sets, or triangular fuzzy membership functions, and the purpose of which aims at distinguishing between similar pairs and dissimilar pairs. The second phase is the criterions of merging similar pairs, as the phase of similarity analysis has been completed. Further, based on Leng's work (Leng *et al*., 2009), the similarity-based pruning strategy has been processing by equations 2.55-2.71.

4.3.2 Two-Layer Optimisation Scheme

A two-layer optimisation scheme consists of the concept of uni-α-cuts of fuzzy sets and the gradient descent method has been applied for refining the final fuzzy model. Basically, the motivation of the two-layer optimisation scheme originated from using local minimum optimisation methods to approximately close the global minimum with much less computational resource than global minimum optimisation methods. Through discovering the optimal degree of the uni-α-cuts of fuzzy sets, a better local minimum by the gradient descent method can be reached. In other words, the optimal degree of the uni-α-cuts of fuzzy sets can provide a "better starting point" for the gradient descent method.

The Application of Uni-α-Cuts of Fuzzy Sets

From the practical viewpoint of the α-cuts of fuzzy sets, to decrease the uncertainty brought by the system under observation, the α-cuts of fuzzy sets can be applied to discover the optimal degree of sensitivity for the system under observation. In other words, uncertainty can be decreased by finding the optimal degree of sensitivity to the system. As there is just one input variable in a single-input-single-output (SISO) fuzzy system, only one specific degree of the α-cuts of fuzzy sets is. However, in a multi-input-single-output (MISO) fuzzy system, membership functions of each input variable also have their own specific degree of the α-cuts of fuzzy sets, so there are a variety of degrees of the α-cuts of fuzzy sets of input variables. The number of specific degrees of the α-cuts of fuzzy sets required to be calculated depends on the number of input variables in a MISO fuzzy system. The training accuracy would be highly satisfactory if the specific degree of the α-cuts of fuzzy sets of each input variable in the MISO fuzzy system has been acquired entirely.

However, there are two questions to be considered. The first question is the problem of overfitting for validation in testing, and the second question is the huge computational costs spent in finding all specific degrees of the α-cuts of fuzzy set.

So far as the first question is concerned, the training accuracy can be acquired precisely by finding all specific degrees of the α-cuts of fuzzy sets, but the performance of all of the obtained specific degrees of the α-cuts of fuzzy sets of each input variable in the MISO fuzzy system cannot be completely guaranteed to be consistent with the testing examples in validation. Consequently, the problem of overfitting for validation in testing arises.

So far as the second question is concerned, it takes higher computational costs to figure out the specific degrees of the α-cuts of fuzzy sets in a MISO fuzzy system than in a SISO fuzzy system. For instance, if there are three input variables and one output variable in a MISO fuzzy system, three specific degrees of the α-cuts of fuzzy sets are required as a result of three input variables in the MISO fuzzy system. In contrast, only one specific degree of the α-cuts of the fuzzy set has to be calculated in a SISO fuzzy system. This is why the calculation for finding the specific degree of the α-cuts of fuzzy sets in a MISO fuzzy system has a higher computational cost than in a SISO fuzzy system.

To relieve the disadvantages brought by the problems addressed above, the uni-α-cuts of fuzzy sets has been applied instead of the general α-cuts of fuzzy sets. The application of uni-α-cuts of fuzzy sets is derived from the concept of general α-cuts of fuzzy sets. The uni- α-cuts of fuzzy sets means only one global degree of α-cuts of fuzzy sets has to be considered whatever the number of input variables a MISO fuzzy system may possess, and the specific degree of uni-α-cuts of fuzzy sets covers all of the input variables in the MISO fuzzy system. In other words, the concept of uni- α-cuts of fuzzy sets deals with the multiple input variables in a MISO fuzzy system as a single input variable in a SISO fuzzy system. The mechanism of the uni- α-cuts of fuzzy sets only applies to one global degree of α-cuts of fuzzy sets to find the optimal degree of α-cuts of fuzzy sets of all of the input variables in a MISO fuzzy system. Moreover, the advantages can be stated as follows:

- A better local minimum may be acquired.
- The problem of overfitting for validation in testing can be relieved, and the reason is the difference between the training accuracy and the testing accuracy is not too much, as a result only one global degree of the α-cuts of fuzzy sets has been applied.
- Less computational costs are required, as only one global degree of the α-cuts of fuzzy set has to be applied by the process once.
- The preparation of obtaining a better starting point for the gradient descent method.

4.4 The Procedure of the Proposed Work

As described in the introduction, the method used is structured on modelling a fuzzy system identification based on a three-part input-output clustering technique, similarity-based pruning strategy, and a two-layer optimisation scheme. According to the concept discussed above, the steps of the proposed method can be stated as follows:

- **Step 1:** Three-part Input-Output Clustering approach
- **Step 2:** Similarity Analysis
- **Step 3:** Merge scheme.

Initial fuzzy model completed.

- **Step 4:** Tune the parameters of the initial fuzzy models by the uni-α-cuts of fuzzy sets.
- **Step 5:** Tune the parameters of the initial fuzzy models by the gradient descent method.

End: Final fuzzy model completed.

Normally, the three-part input-output clustering based on triangular membership functions can be achieved using the equations provided by the chapter 3. After the three-part input-output clustering approach is completed, the similarity-based pruning strategy can be processed by equations 2.55-2.71 based on the resultant clusters generated by the three-part input-output clustering approach. Further, the reason of using triangular membership functions is simple to calculate for a similarity-based pruning strategy because of its finite partition range of input space.

Once the new fuzzy rules are generated, the parameters of the initial fuzzy model are processed to the first stage of the two-layer optimisation scheme, by uni-α-cuts of fuzzy sets for parameters optimisation by equation 4.1. As described before, the task of this stage is to discover a "better starting point" for the gradient descent method. Afterwards, the new parameters generated by the first layer of the optimisation scheme, uni-α-cuts of the fuzzy set, are processed in the second layer of the optimisation scheme, the gradient descent method, using equations 3.1-3.26 until the algorithm converges, either by reaching the local minimum of the function or the stopping criterion $\left| y(t) - y^r \right| \le \varepsilon$ is satisfied.

4.5 Simulation

In order to validate the reliability of the diverse performance of the proposed method, the comparisons of the simulation results between other methods and the proposed method are described in the following content by (1) two-input nonlinear *sinc* function approximation problem, (2) two types of static function approximation problems, (3) synthetic one-dimension function approximation problem, (4) synthetic two-dimensions function approximation problem, (5) three-input nonlinear function approximation problem, (6) nonlinear dynamic system identification, and (7) two classical benchmark problems. Further, the experiments are only one run as well as not cross validated, and that the comparisons with others are 'general'. However, the proposed algorithm has been tested on many datasets, which confirms the positive results of the proposed algorithm.

4.5.1 Benchmark Problem: Mackey-Glass time-series prediction

The proposed method has been validated on a benchmark problem: the Mackey-Glass chaotic time series prediction. The chaotic time series is generated from the Mackey-Glass differential delay equation defined by:

$$x(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t). \qquad \textbf{(4.2)}$$

where $x(0) = 1.2$, $x(t) = 0$ for $t < 0$, and $\tau = 17$.

The target of the simulation aims at using the past values of x to predict the future value of x. In this simulation, 6000 data were generated by the fourth-order Runge-Kutta method with a step size of 0.1 for the purpose of training and testing, and aimed at predicting the value $x(t+\Delta T)$ from input vector $[x(t-18);x(t-12);x(t-6);x(t)]$ for any value of time t. According to the prediction of different $x(t+\Delta T)$, the simulation will be examined in two parts, including $x(t+85)$ and $x(t+6)$.

The accuracy of the proposed method will be measured by root mean square error (RMSE) as well as by a non-dimensional error index (NDEI) defined as the ratio of the RMSE over the standard deviation of the target data. The comparisons of the accuracy in RMSE and NDEI between the proposed method and other methods are reported in Tables 4.1 and 4.2. From these results the proposed method attains a better RMSE and NDEI with fewer rules than that of other methods in the comparison.

The simulation results have been compared with other methods, including the work of (1) Kasabov and Song (2002), (2) Platt (1991), (3) Deng and Kasabov (2000), (4) Kasabov (1998), (5) Fritze (1995), (6) Angelov and Filev (2004), (7) Angelov and Zhou (2006), (8) Wang and Zeng (2008), (9) Chen *et. al.* (1991), (10) Cho and Wang (1996), (11) Nauck and Kruse (1999), (12) Leng *et. al.* (2005), (13) Paiva and Dourado (2004), (14) Wu and Er (2000), (15 Leng *et. al.* (2009), (23) Angelov and Filev (2005), and (38) Rong and Sundararajan (2006).

(A) $Output=[x(t+85)]$**;** $Inputs=[x(t-18);x(t-12);x(t-6);x(t)]$**;**

In this simulation, the input data can be defined by $Inputs=[x(t-18);x(t-12);x(t-6);x(t)]$, and the output result can be defined by

$Output = [x(t+85)]$. In order to compare this with the other methods, 3000 data points $t \in [201, 3200]$ were generated for training, and 500 data points $t \in [5001, 5500]$ for testing.

Method	*Rules (neurons)*	*RMSE for Training*	*RMSE for Testing*	*NDEI for Training*	*NDEI for Testing*
DENFIS (Kasabov & Song, 2002)	58	N/A	N/A	N/A	0.27600
DENFIS (Kasabov & Song, 2002)	883	N/A	N/A	N/A	0.03300
RAN Platt (Platt, 1991)	113	N/A	N/A	N/A	0.37300
ESOM (Deng & Kasabov, 2000)	114	N/A	N/A	N/A	0.32000
ESOM (Deng & Kasabov, 2000)	1000	N/A	N/A	N/A	0.04400
EFuNN (Kasabov, 1998)	193	N/A	N/A	N/A	0.40100
EFuNN (Kasabov, 1998)	1125	N/A	N/A	N/A	0.09400
Neural gas (Fritze, 1995)	1000	N/A	N/A	N/A	0.06200
eTS (Angelov & Filev, 2004)	9	N/A	N/A	N/A	0.37200
eTS (Angelov & Filev, 2004)	113	N/A	N/A	N/A	0.09540
Simple_eTS (Angelov & Filev, 2005)	11	N/A	N/A	N/A	0.39400
exTS (Angelov & Zhou, 2006)	10	N/A	N/A	N/A	0.33100
exTS (Angelov & Zhou, 2006)	9	N/A	N/A	N/A	0.36100
SAFIS (Rong & Sundararajan, 2006)	6	N/A	N/A	N/A	0.37600
ICLA (Wang & Zeng, 2008)	12	0.00131	0.00131	N/A	N/A
Proposed Work – 2 By Zero-Order TS Fuzzy Model	16	0.05458	0.05458	0.24342	0.23783
Proposed Work – 2 By Affine TS Fuzzy Model	16	0.02008	0.01933	0.08954	0.08423
Proposed Work – 2 By Zero-Order TS Fuzzy Model	81	0.01708	0.01612	0.07617	0.07024
Proposed Work – 2 By Affine TS Fuzzy Model	81	0.00561	0.00564	0.02505	0.02459

TABLE 4.1 COMPARISON OF RESULTS BASED ON $Output = [x(t+85)]$

(B) *Output* $=[x(t+6)]$; *Inputs* $=[x(t-18); x(t-12); x(t-6); x(t)]$;

In this simulation, the input data can be defined by *Inputs* $=[x(t-18); x(t-12); x(t-6); x(t)]$, and the output result can be defined by *Output* $=[x(t+6)]$. In order to compare this with the other methods, 1000 data points $t \in [124, 1123]$ were generated for training, and 1000 data points $t \in [1124, 2123]$ for testing.

Method	*Rules (neurons)*	*RMSE for Training*	*RMSE for Testing*	*NDEI for Training*	*NDEI for Testing*
OLS (Chen et. al., 1991)	13	0.0158	0.0163	N/A	N/A
RBF-AFS (Cho & Wang, 1996)	21	0.0107	0.0128	N/A	N/A
NEFPROX (Nauck & Kruse, 1999)	129	0.0315	0.0332	N/A	N/A
SOFNN (Leng et. al., 2005)	4	0.0123	0.0118	N/A	N/A
Paiva's (Paiva & Dourado, 2004)	9	0.0228	0.0239	N/A	N/A
DFNN (Wu & Er, 2000)	5	0.0132	0.0131	N/A	N/A
SANFS (Leng et. al., 2009)	10	0.0084	0.0088	N/A	N/A
Proposed Work – 2 By Zero-Order TS Fuzzy Model	16	0.01148	0.0117	0.051136	0.052123
Proposed Work – 2 By Affine TS Fuzzy Model	16	0.00221	0.0022	0.009839	0.009968

TABLE 4.2 COMPARISON OF RESULTS BASED ON *Output* $=[x(t+6)]$

4.5.2 Benchmark Problem: Box-Jenkins Gas Furnace

The system identification of Box–Jenkins gas furnace is a well-known benchmark problem, and the Box–Jenkins gas data (Box & Jenkins, 1976) has 296 data pairs $\{(u(t), y(t)) \mid t \in [1, 296]\}$, where $y(t)$ is the output CO_2 concentration and $u(t)$ is the input gas flow rate. The series-parallel model can be obtained by $y(t) = f(y(t-1), u(t-4))$, $y(t) = f(y(t-1), u(t-3), u(t-4))$, and $y(t) = f(y(t-1), y(t-2), y(t-3), u(t), u(t-1), u(t-2))$. The accuracy of the proposed method will be measured by mean square error (MSE), RMSE) and NDEI. In this simulation, 200 data samples are randomly chosen for training, and 90 data samples for testing. From Tables 4.3, 4.4 and 4.5, the proposed method attains a better MSE, RMSE, and NDEI with fewer rules than for other methods in the comparison.

The simulation results have been compared with other methods, including (1) Box and Jenkins (1976), (2) Tong (1979), (3) Pedrycz (1984), (4) Xu and Lu (1987), (5) Li and Mukaidono (1995), (6) Wang and Langari (1996), (7) Cheng and Hsieh (2001), (8) Angelov and Filev (2004), (9) Angelov and Filev (2005), (10) Sugeno and Yasukawa (1993), (11) Castellano and Fanelli (2000), (12) Yen *et. al.* (1998), (13) Juang (2002), (14) Lee and Ouyang (2003), (15) Setnes *et. al*. (1998), (16) Ouyang *et. al.* (2005), (17) Sugeno and Tanaka (1991), (18) Kim *et. al.* (1997), (19) *Kim et. al.* (1998), (20) Tsekouras (2005), (21) Leng *et. al.* (2005), and (22) Leng *et. al.* (2009).

(A) $y(t) = f\left(y(t-1), u(t-4)\right)$**;**

In this simulation, 200 data samples were randomly chosen for training, and 90 data samples for testing.

Method	*Rules (neurons)*	*MSE for Training*	*RMSE for Training*	*RMSE for Testing*	*NDEI for Training*	*NDEI for Testing*
Tong's (Tong, 1979)	19	0.469	N/A	N/A	N/A	N/A
Pedrycz's (Pedrycz, 1984)	81	0.320	N/A	N/A	N/A	N/A
Xu's (Xu & Lu, 1987)	25	0.328	N/A	N/A	N/A	N/A
Li's (Li & Mukaidonom, 1995)	6	0.178	N/A	N/A	N/A	N/A
Wang's (Wang & Langari, 1996)	5	0.158	N/A	N/A	N/A	N/A
Cheng's (Cheng & Hsieh, 2001)	4	0.146	N/A	N/A	N/A	N/A
SOFNN-I (Leng et. al., 2005)	4	0.131	N/A	N/A	N/A	N/A
eTS (Angelov & Filev, 2004)	5	N/A	N/A	N/A	N/A	0.30571
Simple_eTS (Angelov & Filev, 2005)	3	N/A	N/A	N/A	N/A	0.30041
Proposed Work – 2 By Zero-Order TS Fuzzy Model	4	0.079	0.28192	0.722774	0.08994	0.29135
Proposed Work – 2 By Affine TS Fuzzy Model	4	0.074	0.27371	0.619156	0.08732	0.24958

TABLE 4.3 COMPARISON OF RESULTS BASED ON $y(t) = f\left(y(t-1), u(t-4)\right)$

(B) $y(t) = f\left(y(t-1), u(t-3), u(t-4)\right)$**;**

In this simulation, 145 data samples were randomly chosen for training, and 145 for testing.

Method	*Rules (neurons)*	*MSE for Training*	*RMSE for Training*	*RMSE for Testing*	*NDEI for Training*	*NDEI for Testing*
FLBA (Sugeno & Yasukawa, 1993)	6	0.190	N/A	N/A	N/A	N/A
SONFIN (Castellano & Fanelli, 2000)	6	0.185	N/A	N/A	N/A	N/A
SVD-QR-CP (Yen et. al., 1998)	7	N/A	N/A	N/A	0.748	1.036
ACA (Juang, 2002)	7	N/A	N/A	N/A	0.316	0.445
SCRG (Lee & Ouyang, 2003)	7	N/A	N/A	N/A	0.205	0.385
SM (Setnes et. al., 1998)	7	N/A	N/A	N/A	0.216	0.415
MFC (Ouyang et. al., 2005)	7	N/A	N/A	N/A	0.147	0.369
SANFS (Leng et. al., 2009)	2	N/A	N/A	N/A	0.089	0.132
Proposed Work – 2 By Zero-Order TS Fuzzy Model	8	0.02031	0.14252	0.565249	0.042	0.217
Proposed Work – 2 By Affine TS Fuzzy Model	8	0.01320	0.11491	0.554255	0.033	0.213

TABLE 4.4 COMPARISON OF RESULTS BASED ON $y(t) = f\left(y(t-1), u(t-3), u(t-4)\right)$

(C) $y(t)=f\big(y(t-1),y(t-2),y(t-3),u(t),u(t-1),u(t-2)\big)$**;**

In this simulation, all 296 data samples were used for training.

Method	*Rules (neurons)*	*MSE for Training*	*RMSE for Training*
Sugeno (Sugeno & Tanaka, 1991)	2	0.068	N/A
Kim *et al.* (Kim *et al.*, 1997)	2	0.055	N/A
Kim *et al.* (Kim *et al.*, 1998)	2	0.062	N/A
Box & Jen. (Box & Jenkins, 1976)	N/A	0.202	N/A
SOFNN-II (Leng et. al., 2005)	2	0.057	N/A
Tsekouras's (Tsekouras, 2005)	8	0.075	N/A
Proposed Work – 2 By Zero-Order TS Fuzzy Model	8	0.069	0.26346
Proposed Work – 2 By Affine TS Fuzzy Model	8	0.056	0.23746

TABLE 4.5 COMPARISON OF RESULTS BASED ON

$y(t)=f\big(y(t-1),y(t-2),y(t-3),u(t),u(t-1),u(t-2)\big)$

4.5.3 Two-Input Nonlinear *Sinc* Function

This example is used to demonstrate the performance of the proposed method, and the function is defined as:

$$z=\mathrm{sinc}(x,y)=\frac{\sin(x)\sin(y)}{xy}. \qquad \textbf{(4.3)}$$

where $x\in[-10,10]$, $y\in[-10,10]$.

In this simulation, 242 data were generated by equation 5.20 within the domain, and 121 data were randomly chosen for training and the remaining 121 data were used for testing. The accuracy of the proposed method was measured by RMSE. From Table 4.6, the proposed method attains a better RMSE with fewer rules than that of other methods in the comparison, and the simulation results have been compared with other methods, including (1) Leng *et. al.* (2004) and (2) Leng *et. al.* (2009).

Method	*Rules (neurons)*	*Parameters*	*RMSE for Training*	*RMSE for Testing*
SANFS (Leng et. al., 2009)	17	N/A	0.0428	0.0658
OSOFNN (Leng et. al., 2004)	N/A	45	0.0565	0.0956
Proposed Work – 2 By Zero-Order TS Fuzzy Model	9	27	0.0293	0.0416
Proposed Work – 2 By Affine TS Fuzzy Model	9	45	0.0195	0.0324

TABLE 4.6 COMPARISON OF RESULTS BASED ON (4.3)

4.5.4 Static Function Approximation I

The static function to be approximated is the Hermite polynomial:

$$f(x)=1.1(1-x+2x^2)\exp(-\frac{x^2}{2}). \quad \textbf{(4.4)}$$

where $x\in[-4,4]$

In this simulation, 200 data were generated by equation 4.4 within the domain, and all of the data were used for training. The accuracy of the proposed method wa measured by RMSE. From Table 4.7, the proposed method attains a better RMSE with fewer rules than those of the other methods in the comparison. The simulation results have been compared with other methods, including (1) Wu and Er (2000), (2) Chen *et. al.* (1991), (3) Lu *et. al.* (1997), and (4) Kadirkamanathan and Niranjan (1993).

Method	*Rules (neurons)*	*RMSE for Training*
DFNN (Wu & Er, 2000)	6	0.0056
OLS (Chen et. al., 1991)	7	0.0095
M-RAN (Lu et. al., 1997)	7	0.0090
RANEKF (Kadirkamanathan and Niranjan, 1993)	13	0.0262
Proposed Work – 2 By Zero-Order TS Fuzzy Model	6	0.0251
Proposed Work – 2 By Affine TS Fuzzy Model	6	0.0088

TABLE 4.7 COMPARISON OF RESULTS BASED ON (4.4)

4.5.5 Static Function Approximation II

The static function to be approximated is:

$$y = f(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1.5})^2 . \tag{4.5}$$

where $x_1 \in [1,5]$, $x_2 \in [1,5]$

In this simulation, 50 data were generated by equation 4.5 within the domain, and all of the data was used for training. The accuracy of the proposed method was measured by MSE. From Table 4.8, the proposed method attains a better MSE with fewer rules and parameters than that of the other methods in the comparison, and the simulation results have been compared with other methods, including (1) Sugeno and Yasukawa (1993), (2) Emani *et al.* (1998), (3) Nozaki *et al.* (1997), (4) Lee and Ouyang (2003), (5) Kim *et al.* (1997), (6) Kim *et al.* (1998), (7) Tsekouras (2005), and (8) Wang *et al.* (2007).

Method	*Rules*	*Parameters*	*MSE for Training*
Sugeno and Yasukawa (Sugeno & Yasukawa, 1993)	6	65	0.079
Emani *et al.* (Emani *et al.*, 1998)	8	91	0.0040
Nozaki *et al.* (Nozaki *et al.*, 1997)	25	125	0.0085
Lee and Ouyang (Lee & Ouyang, 2003)	10	N/A	0.0042
Kim *et al.* (Kim *et al.*, 1997)	3	21	0.0197
Kim *et al.* (Kim *et al.*, 1998)	3	21	0.0090
Tsekouras (Tsekouras, 2005)	8	40	0.0042
IOC (Wang *et al.*, 2007)	8	40	0.0075
IOC (Wang *et al.*, 2007)	13	65	0.0025
Proposed Work – 2 By Zero-Order TS Fuzzy Model	16	40	1.74E-05
Proposed Work – 2 By Affine TS Fuzzy Model	16	72	1.04E-06

TABLE 4.8 COMPARISON OF RESULTS BASED ON (4.5)

4.5.6 Synthetic One-Dimension Data

This function is used to demonstrate the performance of the proposed method, and the function is defined as:

$$y = 0.6\sin(\pi x) + 0.3\sin(3\pi x) + 0.1\sin(5\pi x)\,. \qquad \textbf{(4.6)}$$

where $x \in [-1,1]$

In this simulation, 200 data were generated by equation 4.6 within the domain, and 100 data were randomly chosen for training and the remaining 100 data were used for testing. The accuracy of the proposed method was measured by RMSE). From Table 4.9, the proposed method attains a better RMSE with fewer rules than that of the other methods in the comparison. The simulation results have been compared with other methods, including (1) Karyannis and Mi (1997), (2) Pedrycz (2006), and (3) Wang *et al.* (2007).

Method	*Rules (neurons)*	*RMSE for Training*	*RMSE for Testing*
RBFNN (Karyannis & Mi, 1997)	36	0.06±0.024	1.14±0.991
LM (Pedrycz, 2006)	36	0.05±0.006	0.06±0.007
IOC (Wang *et al*., 2007)	7	0.033597	0.035765
Proposed Work – 2 By Zero-Order TS Fuzzy Model	11	0.010075	0.027964
Proposed Work – 2 By Affine TS Fuzzy Model	11	0.002519	0.011110

TABLE 4.9 COMPARISON OF RESULTS BASED ON (4.6)

4.5.7 Synthetic Two-Dimension Data

This function is used to demonstrate the performance of the proposed method, and the function is defined as:

$$y = f(x_1, x_2) = 0.6 + 2x_1 + 4x_2 + 0.5x_1x_2 + 25\sin(0.5x_1x_2). \qquad \textbf{(4.7)}$$

where $x_1 \in [-4,6]$, $x_2 \in [-2,4]$

In this simulation, 200 data were generated by equation 4.7 within the domain, and 100 data were randomly chosen for training and the remaining 100 data were used for testing. The accuracy of the proposed method was measured by RMSE.

From Table 4.10, the proposed method attains a better RMSE with fewer rules than that of the other methods in the comparison. The simulation results have been compared with other methods, including (1) Karyannis and Mi (1997) and (2) Pedrycz (, 2006).

Method	*Rules (neurons)*	*RMSE for Training*	*RMSE for Testing*
RBFNN (Karyannis & Mi, 1997)	25	14.86±0.70	16.83±0.77
LM (Pedrycz, 2006)	25	10.16±0.74	11.90±0.90
Proposed Work – 2 By Zero-Order TS Fuzzy Model	16	3.502812	12.34929
Proposed Work – 2 By Affine TS Fuzzy Model	16	1.904039	7.623169

TABLE 4.10 COMPARISON OF RESULTS BASED ON (4.7)

4.5.8 Three-Input Nonlinear Function Approximation

This function is used to demonstrate the performance of the proposed method, and the function is defined as:

$$t = (1 + x^{0.5} + y^{-1} + z^{-1.5})^2 \qquad \textbf{(4.8)}$$

where $x \in [1,6]$, $y \in [1,6]$, $z \in [1,6]$. 216 training data as well as 125 testing data were randomly generated using the above equation.

The simulation results have been compared by MAPE with other methods, including (1) Jang (1993), (2) Wu *et al.* (2001), and (3) Leng *et al.* (2005). The comparisons of the accuracy in MAPE between the proposed method and the other methods are reported in Table 4.11. From Table 4.11, the proposed method holds a better MAPE as either a training example or testing example than those of other methods in the comparison. The proposed method also uses fewer rules and parameters than the other methods in the comparison.

Method	*Rules (neurons)*	*Parameters*	*APE of Training*	*APE of Testing*
ANFIS (Jang, 1993)	8	50	0.043	1.066
GDFNN (Wu *et al*., 2001)	10	64	2.11	1.54
SOFNN (Leng *et al*., 2005)	9	60	1.1380	1.1244
Proposed Work – 2 By Zero-Order TS Fuzzy Model	8	26	0.003410	0.003265
Proposed Work – 2 By Affine TS Fuzzy Model	8	50	0.000406	0.000563

TABLE 4.11 COMPARISON OF RESULTS BASED ON (4.8)

4.5.9 Nonlinear Dynamic System Identification

Finally, the proposed method has been used to identify a variety of nonlinear dynamic systems. The accuracy of the proposed method was measured by MSE, as well as RMSE. From Tables 4.12, 4.13, 4.14, and 4.15, the proposed method attains a better MSE and RMSE with fewer rules than those of the other methods in the comparison. The simulation results have been compared with other methods, including (1) Abonyi (1999), (2) Wang and Yen (1999), (3) Yen and Wang (1999), (4) Yen and Wang (1998), (5) Wang *et al.* (2007), (6) Angelov and Filev (2004), (7) Angelov and Filev (2005), (8) Lu *et al.* (1997), (9) Kadirkamanathan and Niranjan (1993), (10) Rong *et al.* (2006), (11) Wang and Yen (1999), (12) Wang *et al.* (2008), (13) Chen *et al.* (1991), (14) Cho and Wang (1996), (15 Wu and Er (2000), (16) Wu *et al.* (2001), (16) Leng *et al.* (2004), and (17) Leng *et al.* (2009).

(A) The nonlinear dynamic system can be identified by:

$$y(k) = g(y(k-1), y(k-2)) + u(k-1). \qquad \textbf{(4.9)}$$

where $g(y(k-1), y(k-2)) = \dfrac{y(k-1)y(k-2)[y(k-1)-0.5]}{1+y^2(k-1)y^2(k-2)}$, $u(k) = \sin\left(\dfrac{2\pi t}{25}\right)$, $y(0) = y(1) = 0$

In this simulation, data is generated by equation 4.9, and 200 data samples were randomly chosen for training, and 200 data samples were randomly chosen for testing. The accuracy of the proposed method was measured by MSE.

Method	*Rules*	*MSE for Training*	*MSE for Testing*
GG-TLS (Abonyi, 1999)	12	3.7E-04	2.9E-04
GG-LS (Abonyi, 1999)	12	3.7E-04	2.9E-04
EM-TI (Abonyi, 1999)	12	2.4E-04	4.1E-04
EM-NI (Abonyi, 1999)	12	3.4E-04	2.3E-04
Wang (Wang and Yen, 1999)	28	3.3E-04	6.0E-04
Yen (Yen & Wang, 1999)	20	6.8E-04	2.4E-04
Yen (Yen & Wang, 1998)	23	3.2E-05	1.9E-03
IOC (Wang *et al*., 2007)	8	4.28E-04	4.41E-04
IOC (Wang *et al*., 2007)	12	4.9E-05	5E-05
Proposed Work – 2 By Zero-Order TS Fuzzy Model	8	1.837E-03	1.812E-03
Proposed Work – 2 By Affine TS Fuzzy Model	8	1.19E-08	3.21E-09

TABLE 4.12 COMPARISON OF RESULTS BASED ON (4.9)

(B) The nonlinear dynamic system can be identified by:

$$y(k) = g(y(k-1), y(k-2)) + u(k-1). \qquad \textbf{(4.10)}$$

where $g(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)[y(k-1)-0.5]}{1+y^2(k-1)+y^2(k-2)}$, $u(k) = \sin\left(\frac{2\pi t}{25}\right)$, $y(0) = y(1) = 0$

In this simulation, data was generated by equation 4.10, and 5000 data samples were randomly chosen for training, and two sets of 200 data samples and 300 data samples were randomly chosen for testing. The accuracy of the proposed method was measured by RMSE.

Method	*Rules*	*RMSE for Training*	*RMSE for Testing*
SAFIS (Rong *et al*., 2006)	17	0.0539	0.0221
M-RAN (Lu *et al*., 1997)	22	0.0371	0.0271
RANEKF (Kadirkamanathan & Niranjan, 1993)	35	0.0273	0.0297
Simple_eTS (Angelov & Filev, 2005)	22	0.0528	0.0225
eTS (Angelov & Filev, 2004)	49	0.0292	0.0212
HA (Wang & Yen, 1999)	28	0.0182	0.0244
Proposed Work – 2 By Zero-Order TS Fuzzy Model	8	0.007619	0.007006
Proposed Work – 2 By Affine TS Fuzzy Model	8	9.47E-06	1.1E-10

TABLE 4.13 COMPARISON OF RESULTS BASED ON (4.10) BY 200 TESTING DATA

Method	*Rules*	*RMSE for Training*	*RMSE for Testing*
SAFIS (Rong *et al*., 2006)	17	0.0539	0.0221
M-RAN (Lu *et al*., 1997)	22	0.0371	0.0271
RANEKF (Kadirkamanathan & Niranjan, 1993)	35	0.0273	0.0297
Simple_eTS (Angelov & Filev, 2005)	22	0.0528	0.0225
eTS (Angelov & Filev, 2004)	49	0.0292	0.0212
HA (Wang & Yen, 1999)	28	0.0182	0.0244
ICLA (Wang *et al*., 2008)	8	0.0321	0.0318
ICLA (Wang *et al*., 2008)	20	0.0012	7.48E-04
Proposed Work – 2 By Zero-Order TS Fuzzy Model	8	0.007619	0.007006
Proposed Work – 2 By Affine TS Fuzzy Model	8	9.47E-06	1.1E-10

TABLE 4.14 COMPARISON OF RESULTS BASED ON (4.10) BY 300 TESTING DATA

(C) The nonlinear dynamic system can be identified by:

$$y(t+1) = g(y(t), y(t-1)) + u(t)\,. \qquad \textbf{(4.11)}$$

where $g(y(t), y(t-1)) = \frac{y(t)y(t-1)[y(t)+2.5]}{1+y^2(t)+y^2(t-1)}$, $\quad u(k) = \sin\left(\frac{2\pi t}{25}\right)$, $y(0) = y(1) = 0$

In this simulation, the data was generated by equation 4.11, and 200 data samples were generated within the domain for training, and 200 data samples for testing. The accuracy of the proposed method was measured by RMSE

.

Training samples: 200 data samples, $t \in [1,200]$

Testing samples: 200 data samples, $t \in [401,600]$

Method	*Rules*	*Parameters*	*RMSE for Training*	*RMSE for Testing*
DFNN (Wu & Er, 2000)	6	48	0.0283	N/A
GDFNN (Wu & Er, 2001)	6	48	0.0241	N/A
OSOFNN (Leng *et al*., 2004)	5	46	0.0157	0.0151
OLS (Chen *et al*., 1991)	65	326	0.0288	N/A
RBF-AFS (Cho & Wang, 1996)	35	280	0.1384	N/A
SANFS (Leng *et al*., 2009)	8	N/A	0.0114	0.0107
Proposed Work – 2 By Zero-Order TS Fuzzy Model	8	26	0.029812	0.028627
Proposed Work – 2 By Affine TS Fuzzy Model	8	50	2.21E-04	0.000106

TABLE 4.15 COMPARISON OF RESULTS BASED ON (4.11)

4.6 Conclusion

The chapter proposes a similarity-based fuzzy learning algorithm with a two-layer optimisation scheme based on the work proposed in the previous chapter, a three-part input-output clustering-based approach for fuzzy system identification. The work proposed in this chapter can be regarded as an extension of the three-part input-output clustering-based fuzzy learning algorithm. In order to generate a compact fuzzy system, a pruning strategy based on similarity analysis between fuzzy sets has been applied for refining the rule base of the fuzzy model by measuring the similarity between fuzzy sets, as well as reducing the redundant rules of the fuzzy system. To avoid the problem of underfitting and that of overfitting, a two-layer optimisation scheme has been introduced for refining the fuzzy model. So far as the parameters optimisation is concerned, a two-layer optimisation scheme is more powerful than only one local minimum parameters optimisation. However, since a two-layer optimisation scheme is more complicated than a single-layer optimisation scheme, a two-layer optimisation scheme may lead to a higher computational cost and the problem of overfitting during validation. Due to the above reasons, the concept of uni-α-cuts of fuzzy set has been developed in the optimisation scheme. To validate the reliability of the proposed method, a number of classical simulations have been conducted using the proposed method and the comparisons of the simulation results with other methods have been discussed. Through the comparisons, the proposed method can achieve a positive reliability.

Further, the contribution brought by the extended work in this chapter can be described as follows. A two-layer optimisation scheme is developed for achieving a better local minimum with much less computational resource. Through discovering the optimal degree of the uni-α-cuts of fuzzy sets, a "better starting point" can be obtained for the gradient descent method. Also, the contribution is developed based on Occam's Razor, because it combines the general concept of the α-cuts of fuzzy sets and the existing gradient descent method to reach a better local minimum without much complicated mechanisms and computational resource.

CHAPTER 5 CONCLUSION AND FUTURE WORK

This thesis has concentrated on learning and identification of fuzzy systems, and a comprehensive introduction and analysis concerning research on learning and identification of fuzzy systems has been provided. Subsequently, two research topics on learning and identification of fuzzy systems have been proposed in this thesis to effectively cope with regression-type problems and function approximation problems. Each takes its assigned responsibility for the performance of fuzzy system identification. So far as the first work proposed in chapter 3 is concerned, a three-part input-output clustering-based approach to fuzzy system identification, this work aims at constructing a complete initial fuzzy model by discovering the proper number of clusters and the appropriate location of those clusters. The second work proposed in chapter 4, details a similarity-based learning algorithm for fuzzy system identification with a two-layer optimisation scheme, that aims at refining the fuzzy rule base by a similarity-based pruning strategy and by providing a more positive parameter optimisation scheme, so making the fuzzy system of chapter 4 more compact and precise.

So far as a perfect modular method for fuzzy system identification is concerned, it should not only positively cope with pattern recognition problems, but also primarily comprise of the following features, well-understanding interpretability, low-degree dimensionality, highly reliability, stable robustness, highly accuracy of the approximation, less computational costs, and maximum performance. However, it is extremely difficult to meet all of these conditions. Moreover, these two research areas can be integrated into one primary work, and this primary work has been designed to meet the above conditions as far as possible.

5.1 Contributions

In order to reach the optimal achievement, this research tries to perform the advantages of the proposed methods under various kinds of regression-type problems and function approximation problems, and overcome the weaknesses that occur during the computation of the overall procedure. Basically, the contributions of this thesis are brought by two proposed works respectively.

The first work, three-part input-output clustering-based approach for fuzzy system identification, can positively discover the proper number of clusters and their appropriate location by integrating a variety of clustering properties effectively. The contribution is developed based on Occam's Razor, because it applies the basic features of existing clustering algorithms to carry out the optimal performance without much complicated mechanisms and computational resource.

The second work, similarity-base learning algorithm for fuzzy system identification with a two-layer optimisation scheme, can achieve a better local minimum with much less computational resource. Through discovering the optimal degree of the uni-α-cuts of fuzzy sets, a "better starting point" can be obtained for the gradient descent method. Also, the contribution is developed based on Occam's Razor, because it combines the general concept of the α-cuts of fuzzy sets and the existing gradient descent method to reach a better local minimum without much complicated mechanisms and computational resource.

In addition to the above description, the research proposed in this thesis anticipates the discovery of the best compromise between the following features, the low degree of the dimensionality, high accuracy of the approximation, well-understood interpretability, maximum performance and minimum computational cost.

5.2 Future Work

As described earlier, a flexible, complete, consistent, compact and precise fuzzy system consists of well-understanding interpretability, low-degree dimensionality, highly reliability, stable robustness, highly accuracy of the approximation, less computational costs, and maximum performance. How to develop a flexible, complete, consistent and compact fuzzy system with a reliable performance that meets the conditions above is always an interesting challenge for future work on learning and identification of fuzzy systems.

Though the proposed learning algorithms for fuzzy system identification can acquire a positive performance by refining the rule base and reducing the useless subsets or clusters, it still cannot break the problem of the "curse of dimensionality". As described in the previous sections 1.4.3 and 4.1, the problem of the "curse of dimensionality" is still an important issue when dealing with large systems and designing a fuzzy learning algorithm, because the number of combinations is too huge to manage. In addition to the problem of the "curse of dimensionality" in learning and identification of the fuzzy system in high dimensional spaces, further research puts the focus on the advanced multi-overlaps similarity analysis in a similarity-based pruning strategy.

In order to understand the problems involved in future work, the problem of the "curse of dimensionality" and further research on multi-overlaps similarity analysis they will be described.

5.2.1 Curse of Dimensionality in Learning and Identification of fuzzy System in High Dimensional Spaces

The "curse of dimensionality" is a term coined by Richard Bellman (1961) and is applied to the problems caused by the rapid increase in volume associated with adding extra dimensions to a (mathematical) space. It is a significant obstacle in machine learning problems that involve learning from few data samples in a high dimensional feature space (Wikipedia, 2006). Simply, the "curse of dimensionality" means the size of a data-set grows exponentially with its dimension *N*. For example, there are ten inputs in a fuzzy system, and each input has two membership functions. Therefore, partitioning will lead to 2^{10} rules, and this is large for a learning algorithm, as illustrated in Fig. 5.1. The partition result from the "curse of dimensionality" can be calculated as follows:

$$\prod_{i=1}^{n} X_i \qquad \textbf{(5.1)}$$

where n is the number of input fuzzy sets of the input variable X, X_i is the input variable in the fuzzy system.

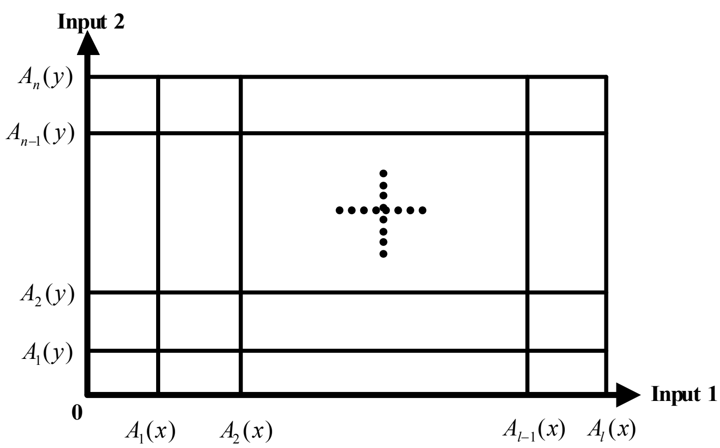

Fig. 5.1 "Cure of dimensionality" in MISO

The problem of the “curse of dimensionality” is that it spends too much time and cost for a learning algorithm to converge the entire system, especially in a high dimensional space. To break the “curse of dimensionality”, requires the refining and simplifying of partitioning or clustering and is a significant issue in solving this problem. Therefore, a good way to break the “curse of dimensionality” is by reducing the complexity of the clustering, and to find the best solution or optimal clustering will be a very significant area of research in the future. For the time being, the focus of research is on reducing the complexity of clustering as a short-term goal. Though simple clustering can possibly lift the problem of the “curse of dimensionality”, it also decreases the accuracy of a precise fuzzy system. Therefore, in addition to reducing the complexity of the clustering, how to keep the accuracy performance of precise fuzzy systems at the same time will be the mid-term goal of future work. So far, there are many theories and methods working on machine learning. Therefore, to discover useful theories and methods and to try to integrate these to make the best compromise between these conditions is a really significant topic, and which will be set as a long-term goal of future work.

5.2.2 Multi-Overlaps Similarity Analysis for A More Flexible Similarity-Based Pruning Strategy

Most of similarity analysis methods only discuss the degree of similarity between two fuzzy numbers, fuzzy sets, or fuzzy membership functions, but less discuss the similarity analysis between more than two. However, it is insufficient to only discuss the similarity analysis between two fuzzy numbers, fuzzy sets, or fuzzy membership functions, because there are a great diversity of similarity analysis and merging schemes worthy of consideration in order to design a more flexible similarity-based pruning strategy. For instance, similarity analysis between three fuzzy sets or fuzzy membership functions merges into two fuzzy sets or fuzzy membership functions, as illustrated in Fig. 5.2, or similarity analysis between four fuzzy sets or fuzzy membership functions merges into two or three fuzzy sets or fuzzy membership functions, as illustrated in Fig. 5.3.

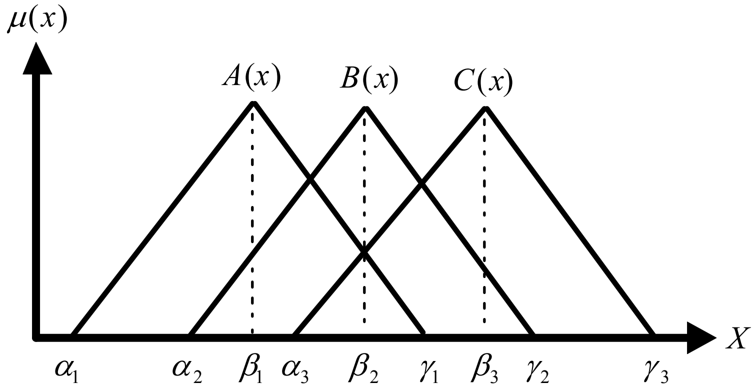

Fig. 5.2 Overlap between three fuzzy sets *A*, *B*, and *C*

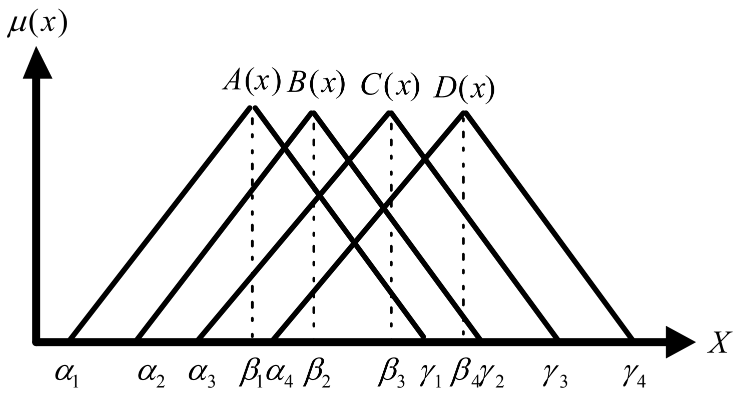

Figure 5.3 Overlap between four fuzzy sets *A*, *B*, and *C*

Similarity Priority:

To measure the degree of similarity between multiple fuzzy sets, the similarity priority based on similarity analysis can be regarded as a possible solution. Basically, the degree of similarity between fuzzy sets can be calculated based on section 2.6.6, afterwards the similarity priority can be acquired by sorting the degree of similarity between each fuzzy set. Since the similarity priority between each fuzzy set has been determined, the multi-merging scheme can merge fuzzy sets. For example, as illustrated in Fig. 5.2, the degree of similarity between fuzzy sets $A(\alpha_1, \beta_1, \gamma_1)$ $B(\alpha_2, \beta_2, \gamma_2)$ and $C(\alpha_3, \beta_3, \gamma_3)$ can be calculated by equations based on section 2.6.6, and which can be represented by $S(A, B)$, $S(B, C)$, as well as $S(A, C)$, respectively. Therefore, the similarity priority between fuzzy sets *A*, *B*, and *C* can be represented by $S_R[S(A,B), S(B,C), S(A,C)]$. In $S_E[S(A,B), S(B,C), S(A,C)]$, the similarity priority can be determined by the value of $S(A, B)$, $S(B, C)$, as well as $S(A, C)$. Moreover, a higher value of the degree of similarity indicates a high priority to be merged. Meanwhile, if $S(A, B) > S(B, C) > S(A, C)$, the similarity priority of $S(A, B)$ could be 1, that of $S(B, C)$ could be 2, and that of $S(A, C)$ could be 3. Therefore, in case two new fuzzy sets are planned to be merged from three fuzzy sets *A*, *B* and *C*, fuzzy sets *A* and *B* could be merged into a new fuzzy set. Also, if $S(A, B) = 0.7$, $S(B, C) = 0.5$, and $S(A, C) = 0.3$, then $S_R[S(A,B), S(B,C), S(A,C)] = [0.7, 0.5, 0.3]$. Therefore, the similarity priority between fuzzy sets *A*, *B* and *C* is fuzzy set *A*, fuzzy set *B*, and then fuzzy set *C*, so fuzzy sets *A* and *B* could be merged into a new fuzzy set.

The above is a simple example applying the concept of similarity priority to the problem of multi-overlaps similarity analysis. However, the more complicated the multi-overlaps similarity analysis, the more fuzzy sets are involved in a diverse way. How to apply the concept of similarity priority to a complicated multi-overlaps similarity analysis is also future work in constructing a complete and compact fuzzy system.

APPENDIX I: Similarity Measure Based on Four Cases Proposed by Chao's Work

In order to generate a simpler fuzzy inference system with fewer rules and parameters, Chao *et al*. (1996) simplified the fuzzy-neueral system with Gaussian membership functions by using similarity analysis to reduce redundant fuzzy rules. The calculation of the intersection of the nonlinear shape of Gaussian membership functions is very complex as a result of the nonlinear shape of Gaussian functions. To decrease the complex calculation of the intersection of the nonlinear shape of Gaussian membership functions, Chao *et al*. transformed the Gaussian membership function into a triangular membership function with an approximated value by:

$$\exp\left[-\frac{(x-m)^2}{\sigma^2}\right] \rightarrow \max\left[0, \frac{\sigma\sqrt{\pi}-|x-m|}{\sigma\sqrt{\pi}}\right] \qquad \textbf{(6.1)}$$

where m is the centre of the Gaussian membership function and the triangular membership function both, σ is the width of the Gaussian membership function and $\sigma\sqrt{\pi}$ is the width of the triangular membership function transformed from the Gaussian membership function.

Similarity Analysis

According to equations 2.54 and 6.1, Chao *et al.* considered the similarity measures in four different cases by recognising the relationships between two triangular fuzzy membership functions, including:

- Case 1: one membership function is the subset of the other membership function
- Case 2: two membership functions possess a single intersection point

- Case 3: two membership functions possess two intersection points
- Case 4: there is no intersection between membership functions

These four different cases are based on the various kinds of relationships between two triangular fuzzy membership functions as illustrated in Fig. 6.1-6.4.

Case 1:

In case 1, illustrated in Fig. 6.1, the membership function $M(B)$ has been included in the membership function $M(A)$, and $M(A)$ as well as $M(B)$ possess the same centre $m_1 = m_2$ with no intersection, where $M(A) = \Delta abe$, $M(B) = \Delta acd$, and Δ means the triangle area. Thus, the degree of similarity based on equation 2.54 for case 1 can be calculated by:

$$S(A,B) = \frac{M(A \cap B)}{M(A \cup B)} = \frac{M(A \cap B)}{M(A) + M(B) - M(A \cap B)} = \frac{M(B)}{M(A)} \qquad \textbf{(6.2)}$$

where $M(A \cap B) = M(B)$

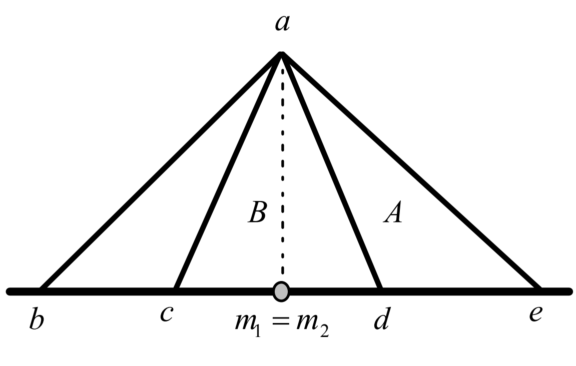

Figure 6.1 Case 1 of Chao's work, $M(B)$ is the subset of $M(A)$, $M(B) \subseteq M(A)$.

Case 2:

In case 2, illustrated in Fig. 6.2, the membership functions $M(A)$ and $M(B)$ possess one intersection point at (s_1, h_1), so the intersection of $M(A)$ and $M(B)$ can be calculated by:

$$M(A \cap B) = \frac{1}{2}(c_1 + c_2) \times h_1 \tag{6.3}$$

where $c_1 = \dfrac{\sigma_1(m_2 - m_1) + \sigma_1(\sigma_1 + \sigma_2)\sqrt{\pi}}{\sigma_1 + \sigma_2}$, $c_2 = \dfrac{\sigma_2(m_2 - m_1) + \sigma_2(\sigma_1 + \sigma_2)\sqrt{\pi}}{\sigma_1 + \sigma_2}$,

and $h_1 = \dfrac{(m_2 - m_1) + (\sigma_1 + \sigma_2)\sqrt{\pi}}{(\sigma_1 + \sigma_2)\sqrt{\pi}}$.

Integrating equation 6.3 into equation 2.54, the degree of similarity for case 2 can be calculated by:

$$S(A, B) = \frac{(c_1 + c_2)h_1}{2(\sigma_1 + \sigma_2)\sqrt{\pi} - (c_1 + c_2)h_1}. \tag{6.4}$$

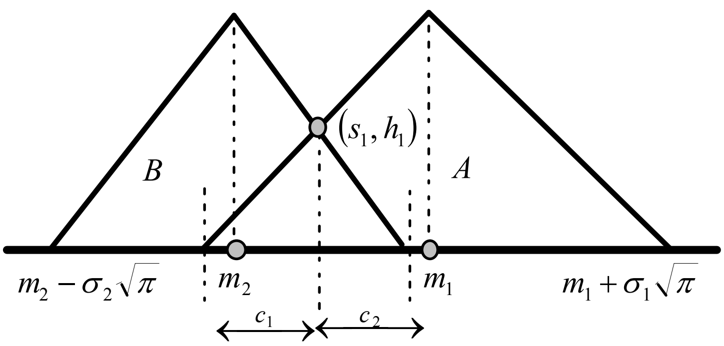

Fig. 6.2 Case 2 of Chao's work, $M(A)$ and $M(B)$ with one intersection point (s_1, h_1).

Case 3:

In case 3, illustrated in Fig. 6.3, the membership functions $M(A)$ and $M(B)$ possess two intersection point at (s_1, h_1) and (s_2, h_2), so the intersection of $M(A)$ and $M(B)$ can be calculated by:

$$M(A \cap B) = \frac{1}{2}\left[c_1 h_1 + c_2 h_2 + c_3 h_3\right] \tag{6.5}$$

where $c_1 = \dfrac{\sigma_1(m_2 - m_1) + \sigma_1(\sigma_2 + \sigma_1)\sqrt{\pi}}{\sigma_1 + \sigma_2}$, $c_2 = \dfrac{\sigma_1(m_2 - m_1) + \sigma_1(\sigma_2 - \sigma_1)\sqrt{\pi}}{\sigma_2 - \sigma_1}$,

$c_3 = 2\sigma_1\sqrt{\pi} - (c_1 + c_2)$, $h_1 = \dfrac{(m_2 - m_1) + (\sigma_2 + \sigma_1)\sqrt{\pi}}{(\sigma_2 + \sigma_1)\sqrt{\pi}}$,

$h_2 = \dfrac{(m_2 - m_1) + (\sigma_2 - \sigma_1)\sqrt{\pi}}{(\sigma_2 - \sigma_1)\sqrt{\pi}}$, and $h_3 = h_1 + h_2$.

Integrating equation 6.5 into equation 2.54, the degree of similarity for case 3 can be calculated by:

$$S(A,B) = \frac{c_1 h_1 + c_2 h_2 + c_3 h_3}{2(\sigma_1 + \sigma_2)\sqrt{\pi} - (c_1 h_1 + c_2 h_2 + c_3 h_3)}. \tag{6.6}$$

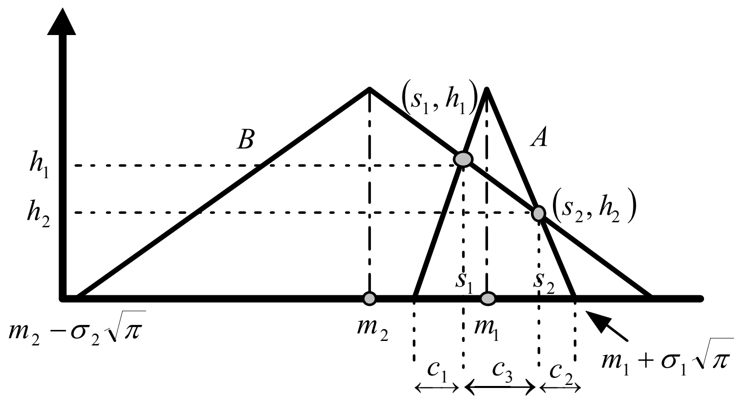

Fig. 6.3 Case 3 of Chao's work, $M(A)$ and $M(B)$ with two intersection points (s_1, h_1) and (s_2, h_2).

Case 4:

In case 4, illustrated in Fig. 6.4, there is no intersection between the membership functions $M(A)$ and $M(B)$, so the intersection of $M(A)$ and $M(B)$ is:

$$M(A \cap B) = 0 \tag{6.7}$$

Thus, integrating equation 6.7 into equation 2.54, the degree of similarity for case 4 is:

$$S(A,B) = 0 \tag{6.8}$$

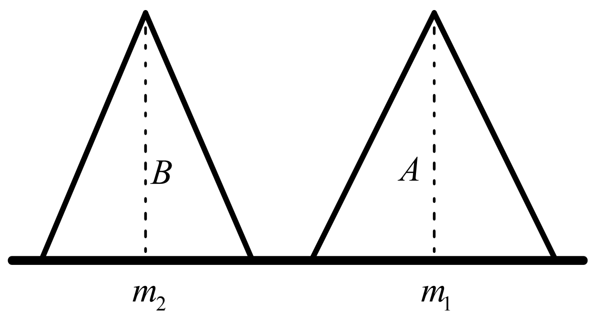

Fig. 6.4 Case 4 of Chao's work, $M(A)$ and $M(B)$ with no intersection.

In the above cases, $0 \le S(A,B) \le 1$, a larger value of $S(A,B)$ indicates a higher degree of similarity between fuzzy sets, fuzzy numbers, or fuzzy membership functions.

Merging Policy

According to the results calculated by Chao *et al*. (1996), two fuzzy membership functions $M(A)$ and $M(B)$ could be merged if the degree of similarity between the membership functions $M(A)$ and $M(B)$ is high enough or exceeds the reference value given by the expert system. Therefore, $M(A)$ and

$M(B)$ would be merged into a new membership function $M(new)$ if $S(A,B) \geq \varepsilon$, where ε is the reference value, and $0 < \varepsilon \leq 1$. Hence, the new centre m_{new} and width σ_{new} of the new membership function $M(new)$ can be determined as follows:

In case 1, illustrated in Fig. 6.1, $M(B)$ is the subset of $M(A)$, $M(B) \subseteq M(A)$, so the new centre m_{new} and a new width σ_{new} can be calculated by:

$$m_{new} = m_1 \ or \ m_2 \tag{6.9}$$

$$\sigma_{new} = \frac{\sigma_1 + \sigma_2}{2} \tag{6.10}$$

In case 2, illustrated in Fig. 6.2, $M(A)$ and $M(B)$ possess one intersection point (s_1, h_1), so the new centre m_{new} and a new width σ_{new} can be calculated by:

$$m_{new} = \frac{(m_1 + m_2) + (\sigma_1 - \sigma_2)\sqrt{\pi}}{2} \tag{6.11}$$

$$\sigma_{new} = \frac{(m_1 - m_2) + (\sigma_1 + \sigma_2)\sqrt{\pi}}{2\sqrt{\pi}} \tag{6.12}$$

In case 3, illustrated in Fig. 6.3, $M(A)$ and $M(B)$ possess two intersection points (s_1, h_1) and (s_2, h_2), so the new centre m_{new} and a new width σ_{new} can be calculated by:

$$m_{new} = \frac{m_1 + m_2}{2} \tag{6.13}$$

$$\sigma_{new} = \frac{\sigma_1 + \sigma_2}{2} \tag{6.14}$$

In case 4, illustrated in Fig. 6.4, since there is no intersection between the membership functions $M(A)$ and $M(B)$, $M(A) \cap M(B) = 0$, the merging behaviour is not necessarily required.

APPENDIX II: Similarity Measure Based on Four Cases Proposed by Jin's Work

Consequently, in order to generate flexible, complete, consistent and compact fuzzy rule systems, Jin *et al.* (1999) applied the fuzzy similarity measure as an index for accomplishing a compact fuzzy system. Jin *et al.* categorised nine different cases into four primary cases based on the triangular membership functions for the purpose of compactness:

- Case 1: two membership functions possess a single intersection point.
- Case 2: two membership functions possess two intersection points between the right side of membership function $M(A)$ and both sides of membership function $M(B)$.
- Case 3: two membership functions possess two intersection points between both sides of membership function $M(A)$ and the left side of membership function $M(B)$.
- Case 4: two membership functions possess three intersection points between both sides of membership functions $M(A)$ and $M(B)$.

According to equation 2.54, the size of fuzzy sets $M(A)$ and $M(B)$ based on Jin *et al.* can be calculated by:

$$M(A) = \int_{-\infty}^{+\infty} A(x)dx, \quad M(B) = \int_{-\infty}^{+\infty} B(x)dx. \tag{7.1}$$

The triplet $(\textit{left vertex}, \textit{centre}, \textit{right vertex})$ of the triangular membership functions $M(A)$ and $M(B)$ can be represented by (a_1, m_1, b_1) and

(a_2, m_2, b_2) respectively, and fuzzy sets $A(x)$ and $B(x)$ described with the triangular membership functions can be defined by the following form:

$$A(x): y = \begin{cases} \dfrac{x - a_1}{m_1 - a_1}, & if \ x < m_1 \\ \dfrac{x - b_1}{m_1 - b_1}, & if \ x > m_1 \end{cases}, \quad B(x): y = \begin{cases} \dfrac{x - a_2}{m_2 - a_2}, & if \ x < m_2 \\ \dfrac{x - b_2}{m_2 - b_2}, & if \ x > m_2 \end{cases} \quad \textbf{(7.2)}$$

These four different cases, based on the various kinds of relationships between two triangular fuzzy membership functions are illustrated in Fig. 7.1-7.4, and the intersection between membership functions $M(A)$ and $M(B)$, $M(A \cap B)$, can be calculated as follows:

Case 1:

In case 1, illustrated in Fig. 7.1, the membership functions $M(A)$ and $M(B)$ possess single intersection point at $P_{21}(x_{21}, y_{21})$, and x_{21} as well as y_{21} can be calculated by:

$$x_{21} = \frac{b_1 m_2 - a_2 m_1}{(b_1 - a_2) + (m_2 - m_1)}, \quad y_{21} = \frac{b_1 - a_1}{(b_1 - a_1) + (m_2 - m_1)}. \quad \textbf{(7.3)}$$

According to equation 7.3, the intersection of $M(A)$ and $M(B)$, $M(A \cap B)$ can be calculated by:

$$M(A \cap B) = \int_{a2}^{x21} \left(\frac{x - a_2}{m_2 - a_2} \right) dx + \int_{x21}^{b1} \left(\frac{x - b_1}{m_1 - b_1} \right) dx = \frac{(b_1 - a_1)^2}{2[(b_1 - a_2) + (m_2 - m_1)]} \quad \textbf{(7.4)}$$

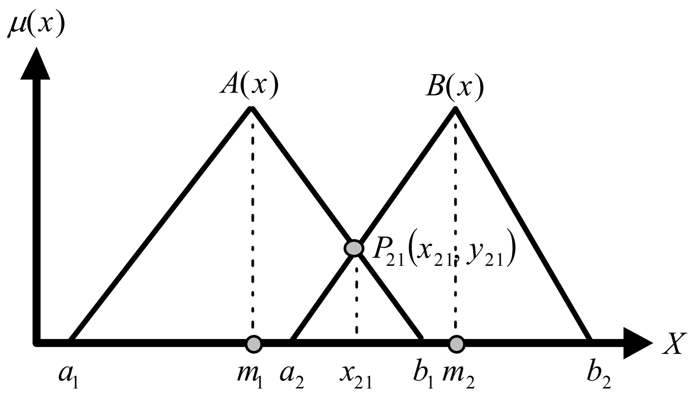

Fig. 7.1 Case 1 of Jin's work, $M(A)$ and $M(B)$ with one intersection point at $P_{21}(x_{21}, y_{21})$.

Case 2:

In case 2, illustrated in Fig. 7.2, the membership functions $M(A)$ and $M(B)$ possess two intersection points at $P_{21}(x_{21}, y_{21})$ and $P_{22}(x_{22}, y_{22})$ between the right side of membership function $M(A)$ and both sides of membership function $M(B)$. Also, x_{21} as well as y_{21} are the same as case 1 and x_{22} as well as y_{22} can be calculated by:

$$x_{22} = \frac{b_1 m_2 - m_1 b_2}{(b_1 + m_2) + (m_2 - m_1)}, \quad y_{22} = \frac{b_1 - b_2}{(b_1 - a_2) + (m_2 - m_1)}. \tag{7.5}$$

According to equation 7.5, the intersection of $M(A)$ and $M(B)$, $M(A \cap B)$ can be calculated by:

$$\begin{aligned} M(A \cap B) &= \int_{a2}^{x21} \left(\frac{x - a_2}{m_2 - a_2} \right) dx + \int_{x21}^{x22} \left(\frac{x - b_1}{m_1 - b_1} \right) dx + \int_{x22}^{b2} \left(\frac{x - b_2}{m_2 - b_2} \right) dx \\ &= \frac{(b_1 - a_1)^2}{2\left[(b_1 - a_2) + (m_2 - m_1)\right]} - \frac{(b_1 - b_2)^2}{2\left[(b_1 - b_2) + (m_2 - m_1)\right]}. \end{aligned} \tag{7.6}$$

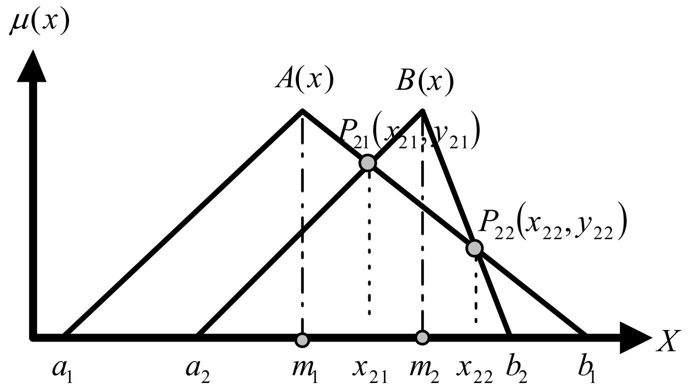

Fig. 7.2 Case 2 of Jin's work, $M(A)$ and $M(B)$ with two intersection points at $P_{21}(x_{21}, y_{21})$ and $P_{22}(x_{22}, y_{22})$.

Case 3:

In case 3, illustrated in Fig. 7.3, the membership functions $M(A)$ and $M(B)$ possess two intersection points at $P_{11}(x_{11}, y_{11})$ and $P_{21}(x_{21}, y_{21})$ between both sides of membership function $M(A)$ and the left side of membership function $M(B)$. Also, x_{21} as well as y_{21} are the same as case 1, and x_{11} as well as y_{11} can be calculated by:

$$x_{11} = \frac{a_1 m_2 - a_2 m_1}{(a_1 - a_2) + (m_2 - m_1)}, \quad y_{11} = \frac{a_1 - a_2}{(a_1 - a_2) + (m_2 - m_1)}. \tag{7.7}$$

According to equation 7.7, the intersection of $M(A)$ and $M(B)$, $M(A \cap B)$ can be calculated by:

$$\begin{aligned} M(A \cap B) &= \int_{a1}^{x11} \left(\frac{x - a_1}{m_1 - a_1} \right) dx + \int_{x11}^{x21} \left(\frac{x - a_2}{m_2 - a_2} \right) dx + \int_{x21}^{b1} \left(\frac{x - b_1}{m_1 - b_1} \right) dx \\ &= \frac{(b_1 - a_2)^2}{2\left[(b_1 - a_2) + (m_2 - m_1)\right]} - \frac{(a_1 - a_2)^2}{2\left[(a_1 - a_2) + (m_2 - m_1)\right]}. \end{aligned} \tag{7.8}$$

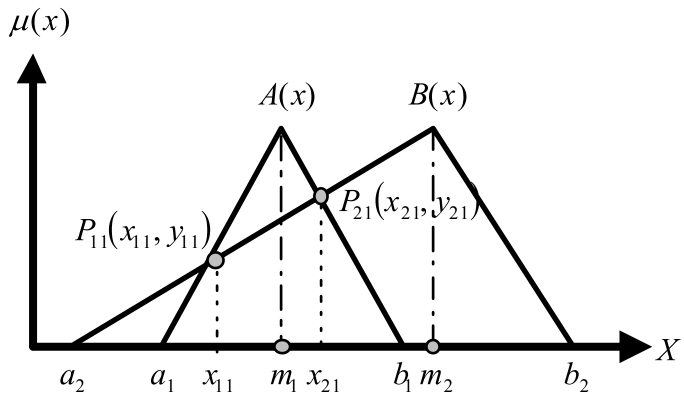

Fig. 7.3 Case 3 of Jin's work, $M(A)$ and $M(B)$ with two intersection points at $P_{11}(x_{11}, y_{11})$ and $P_{21}(x_{21}, y_{21})$.

Case 4:

In case 4, illustrated in Fig. 7.4, the membership functions $M(A)$ and $M(B)$ possess three intersection points at $P_{11}(x_{11}, y_{11})$, $P_{21}(x_{21}, y_{21})$ and $P_{22}(x_{22}, y_{22})$. The definition of $P_{11}(x_{11}, y_{11})$, $P_{21}(x_{21}, y_{21})$ and $P_{22}(x_{22}, y_{22})$ are the same as the above cases. According to equations 7.3, 7.5 and 7.7, the intersection of $M(A)$ and $M(B)$, $M(A \cap B)$ can be calculated by:

$$\begin{aligned} M(A \cap B) &= \int_{a1}^{x11} \left(\frac{x - a_1}{m_1 - a_1} \right) dx + \int_{x11}^{x21} \left(\frac{x - a_2}{m_2 - a_2} \right) dx + \int_{x21}^{x22} \left(\frac{x - b_1}{m_1 - b_1} \right) dx + \int_{x22}^{b2} \left(\frac{x - b_2}{m_2 - b_2} \right) dx \\ &= \frac{(b_1 - a_2)^2}{2[(b_1 - a_2) + (m_2 - m_1)]} - \frac{(a_1 - a_2)^2}{2[(a_1 - a_2) + (m_2 - m_1)]} - \frac{(b_1 - b_2)^2}{2[(b_1 - b_2) + (m_2 - m_1)]}. \end{aligned}$$

(7.9)

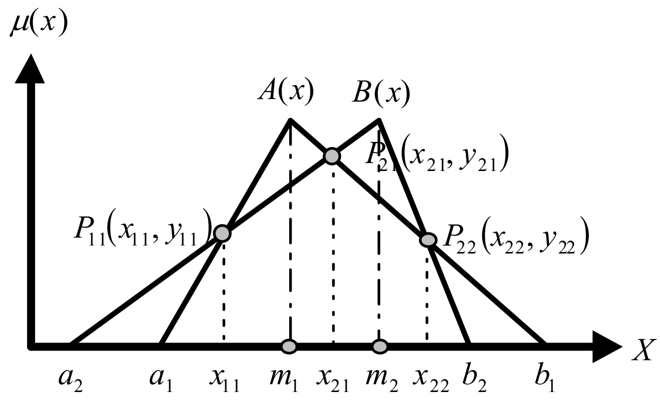

Fig. 7.4 Case 4 of Jin's work, $M(A)$ and $M(B)$ with three intersection points at $P_{11}(x_{11}, y_{11})$, $P_{21}(x_{21}, y_{21})$ and $P_{22}(x_{22}, y_{22})$.

In the above cases, $0 \le S(A,B) \le 1$, and a larger value of $S(A,B)$ indicates a higher degree of similarity between fuzzy sets, fuzzy numbers, or fuzzy membership functions.

REFERENCES

ABONYI, J., BABUSKA, R. & SZEIFERT, F. 1999. Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 32**,** 612-621.

ANGELOV, P. & FILEV, D. 2004. An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 34**,** 484-498.

ANGELOV, P. & FILEV, D. Year. Simpl_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models. *In:* Proceedings of the 14th IEEE International Conference of Fuzzy Systems, 2005. 1068-1073.

ANGELOV, P. & ZHOU, X. Year. Evolving Fuzzy Systems from Data Streams in Real-Time. *In:* Proceedings of 2006 International Sympsium on Evolving Fuzzy Systems, 2006 UK. IEEE Press, 26-32.

AOKI, S., KAWACHI, S. & SUGENO, M. 1990. Application of fuzzy control logic for dead time process in a glass melting furnace. *Fuzzy Sets and Systems,* 38**,** 251-265.

BABUSKA, R. & B., V. H. Year. Applied fuzzy modeling. *In:* IFAC Symposium on Artificial Intelligence in Real Time Control, 1994 Valencia, Spain. 61-66.

BABUSKA, R. & VERBRUGGEN, H. B. Year. A new identification method for linguistic fuzzy models. *In:* FUZZ-IEEE/IFES'95, 1995 Yokohama, Japan. 905-912.

BELLMAN, R. E. 1961. *Adaptive Control Processes,* Princeton, NJ, Princeton University Press.

BEZDEK, J. C. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms,* New York, Plenum Press.

BOX, G. E. P. & JENKINS, G. M. 1976. *Time series analysis: forecasting and control,* Oakland, Holden- Day.

CASILLAS, J., CORDON, O., HERRERA, F. & MAGDALENA, L. 2003. *Interpretability Issues in Fuzzy Modeling*, Springer.

CASTELLANO, G. & FANELLI, A. M. Year. A self-organizing neural fuzzy inference network. *In:* Proceedings of 2000 IEEE Internaternational J. Conference on Neural Networks, 2000 Italy. 14-19.

CAUCHY, A. 1847. Method generale pour la resolution des systems d'equations simultanees. *Comp. Rend. Acad. Sci.* Paris.

CHAO, C. T., CHEN, Y. J. & TENG, C. C. 1996. Simplification of fuzzy-neueral systems using similarity analysis. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 26**,** 344-354.

CHEN, J. Q., XI, Y. G. & ZHANG, Z. J. 1998. A clustering algorithm for fuzzy model identification. *Fuzzy sets and Systems,* 98**,** 319-329.

CHEN, S., COWAN, C. F. N. & GRANT, P. M. 1991. Orthogonal least squares learning algorithm for radial basis function network. *IEEE Transactions on Neural Networks,* 2**,** 302-309.

CHEN, S. H. 1985. Operations on fuzzy numbers with function principal. *Tamkang Journal of Management Sciences,* 6**,** 13-25.

CHEN, S. J. & CHEN, S. M. Year. A new simple center-of-gravity method for handling the fuzzy ranking and the defuzzification problems. *In:* Proceedings of the Eighth National Conference on Fuzzy Theory and Its Applications, 2000 Taipei, Taiwan, Republic of China.

CHEN, S. M. & HSIAO, H. R. 2005. A new method to estimate null vaues in relational database systems based on automatic clustering techniques. *IEEE Information Sciences: an International Journal,* 169**,** 47-60.

CHEN, S. M. & HUANG, C. M. 2003. Generating weighted fuzzy rules from relational database systems for estimating null values using genetic algorithms. *IEEE Transactions on Fuzzy System,* 11**,** 495-506.

CHEN, S. M. & LIN, S. Y. Year. A new method for fuzzy risk analysis. *In:* Artificial Intelligence Workshop, 1995 Taipei. 245-250.

CHENG, C. C. & HSIEH, W. H. 2001. The fuzzy crystallization algorithm: a new approach to complex systems modeling. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 31**,** 891-901.

CHIU, S. L. 1994. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems,* 2**,** 267-278.

CHO, K. B. & WANG, B. H. 1996. Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction. *Fuzzy Sets and Systems,* 83**,** 325-339.

CORDON, O. & HERRERA, F. 1995. *A general study on genetic fuzzy systems*, John Wiley and Sons.

CORDON, O. & HERRERA, F. 1996. *A hybrid genetic algorithm-evolution strategy process for learning fuzzy logic control knowledge bases*, Physica-Verlag.

CZOGOLA, E. & PEDRYCZ, W. 1981. On identification in fuzzy systems and its applications in control problems. *Fuzzy Sets and Systems,* 6**,** 73-83.

DELGADO, M., ANTONIO, G. F. & MARTIN, F. 1997. A fuzzy clustering-based rapid prototyping for fuzzy rule-based modeling. *IEEE Transactions on Fuzzy Systems,* 5**,** 223-232.

DELGADO, M., VILA, M. A. & GOMEZ-SKARMETA, A. F. Year. An unsupervised learning procedure to obtain a classification from a hierarchical clustering. *In:* EUFIT, 1994 Aachen, Germany. 528-533.

DENG, D. & KASABOV, N. Year. Evolving self-organizing maps for online learning, data analysis and modeling. *In:* AMARI, C., GILES, L., GORI, M. & V., P., eds. Proceedings of IJCNN'2000 Neural Networks, Neural Computing: New Challenges Perspectives New Millennium, 2000 New York, NY. 3-8.

DUBOIS, D. & PRADE, H. 1982. *A unifying view of comparison indices in a fuzzy set theoretic framework,* New York, Pergamon.

DUNN, J. C. 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics,* 3**,** 32-57.

EMANI, M. R. & GOLDENBERG, A. A. 1998. Development of a systematic methodology of fuzzy logic modeling. *IEEE Transactions on Fuzzy Systems,* 6**,** 346-361.

FRITZE, B. 1995. A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems,* 7**,** 1-8.

GLORENNEC, P. Y. 1999. *Algorithmes d'apprentissage pour systems d'inference floue,* Paris, France, Hermes.

GONZALEZ, J., ROJAS, I., H., P. & ORTEGA, J. 2001. A new clustering technique for function approximation. *IEEE Transactions on Neural Networks,* 13**,** 132-142.

GONZÁLEZ, J., ROJAS, I. & POMARES, H. 2002. Neural Networks, Clustering Techniques, and Function Approximation Problems. *In:* DORRONSORO, J. (ed.) *Artificial Neural Networks — ICANN 2002.* Springer Berlin / Heidelberg.

GONZALEZ, J., ROJAS, I., POMARES, H. & ORTEGA, J. 2002. A new clustering technique for function approximation. *IEEE Transactions on Neural Networks,* 13**,** 132-142.

GRAHAM, B. P. & NEWELL, R. B. 1988. Fuzzy identification and control of a liquid level rig. *Fuzzy Sets and Systems,* 26**,** 255-273.

GUELY, F. & SIARRY, P. 1993. Gradient descent method for optimizing various fuzzy rule bases. *Fuzzy Systems.* San Francisco, CA , USA.

GUILLAUME, S. 2001. Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review. *IEEE Transactions on Fuzzy Systems,* 9**,** 426-443.

GUPTA, M. M., KISZKA, J. B. & TROJAN, G. M. 1986. Multivariable structure of fuzzy control systems. *IEEE Transactions on Systems, Man and Cybernetics,* 16**,** 638-656.

HAN, J. & KAMBER, M. 2006. *Data Mining: Concepts and Techniques,* San Francisco, Morgan Kaufmann.

HELLENDOORM, H. & DRIANKOV, D. 1997. *Fuzzy Model Identification: Selected Approaches*, Springer.

HOLLAND, J. H. 1975. *Adaptation in Natural and Artificial Systems,* MA, MIT Press.

HSIEH, C. H. & CHEN, S. H. Year. Similarity of generalized fuzzy numbers with graded mean integration. *In:* Proceedings of the Eighth International Fuzzy Systems Association World Congress, 1999 Taipei, Taiwan, Republic of China. 551-555.

HUNG, T. W., FANG, S. C. & NUTTLE, H. L. W. 2003. A two-phase approach to fuzzy system identification. *Journal of Systems Science and Systems Engineering,* 12**,** 408-423.

ISHIBUCHI, H., NOZAKI, K., TANAKA, H., HOSAKA, Y. & MATSUDA, M. 1994. Empirical study on learning in fuzzy systems by rice test analysis. *Fuzzy Sets and Systems,* 64**,** 129-144.

JANG, J. S. R. 1993. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics,* 23**,** 665-684.

JANG, J. S. R. Year. Structure Determination in Fuzzy Modeling: A Fuzzy CART Approach. *In:* IEEE International Conference on Fuzzy Systems, 1994 Orlando. 480-485.

JIN, Y., SEELEN, W. V. & SENDHOFF, B. 1999. On generation FC3 fuzzy rule systems from data using evolution strategies. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 29**,** 829-845.

JUANG, C. F. 2002. A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms. *IEEE Transactions on Fuzzy Systems*.

KADIRKAMANATHAN, V. & NIRANJAN, M. 1993. A function estimation approach to sequential learning with neural networks. *Neural Computation,* 5**,** 954-975.

KARYANNIS, K. B. & MI, G. W. 1997. Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques,. *IEEE Transactions on Neural Networks,* 8**,** 1492-1506.

KASABOV, N. 1998. Evolving fuzzy neural networks - algorithms, applications and biological motivation. *In:* YAMAKAWA, T. & MATSUMOTO, G. (eds.) *Methodologies for the Conception, Design and Application of Soft Computing.* Singapore: World Scientific.

KASABOV, N. K. & SONG, Q. 2002. DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems,* 10, 144-154.

KIM, E., PARK, M., JI, S. & PARK, M. 1997. A new approach to fuzzy modeling. *IEEE Transactions on Fuzzy Systems,* 5, 328-337.

KIM, E., PARK, M., KIM, S. & PARK, M. 1998. A transformed input-domain approach to fuzzy modeling. *IEEE Transactions on Fuzzy Systems,* 6, 596-604.

LEE, C. C. 1990a. Fuzzy logic in control systems: Fuzzy logic controller-part I. *IEEE Transactions on Systems, Man and Cybernetics,* 20, 404-418.

LEE, C. C. 1990b. Fuzzy logic in control systems: Fuzzy logic controller-part II. *IEEE Transactions on Systems, Man and Cybernetics,* 20, 419-435.

LEE, H. S. Year. An optimal aggregation method for fuzzy opinions of group decision. *In:* Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999 Tokyo, Japan. 314-319.

LEE, S. J. & OUYANG, C. S. 2003. A neurofuzzy system modeling with self-constructing rule generation and hybrid SVD-based learning. *IEEE Transactions on Fuzzy Systems,* 11, 341-353.

LEE, S. J. & WANG, H. S. 2009. A dynamic modular method for estimating null values in relational database systems. *International Journal of Computer Information Systems and Industrial Management Applications,* 1, 249-257.

LEE, S. J. & ZENG, X. J. Year. A modular method for estimating null values in relational database systems. *In:* Proceedings of the Eighth International Conference on Intelligent Systems Design and Applications, 2008 Kaohsiung, Taiwan, Republic of China. 415-419.

LEE, S. J. & ZENG, X. J. 2010. A three-part input-output clustering-based approach to fuzzy system identification. *The Tenth International Conference on Intelligent Systems Design and Applications.* Cairo, Egypt, to appear.

LEE, S. J., ZENG, X. J. & WANG, H. S. 2009. Generating automatic fuzzy system from relational database system for estimating null values. *Cybernetics and Systems,* 40**,** 528-548.

LENG, G., MCGINNITY, T. M. & PRASAD, G. 2005. An Approach for On-line Extraction of Fuzzy Rules using a Sefl-organising Fuzzy Neural Network. *Fuzzy Sets and Systems,* 150**,** 211-243.

LENG, G., PRASAD, G. & MCGINNITY, T. M. 2004. An on-line algorithm for creating self-organising fuzzy neural networks. *Neural Networks,* 17**,** 1477-1493.

LENG, G., ZENG, X. J. & KEANE, J. A. 2009. A hybrid learning algorithm with a similarity-based pruning strategy for self-adaptive neuro-fuzzy systems. *Applied Soft Computing,* 9**,** 1354-1366.

LESKI, J. M. 2003. Generalized weighted conditional fuzzy clustering. *IEEE TRansactions on Fuzzy Systems,* 11**,** 709-715.

LI, R. P. & MUKAIDONO, M. 1995. A new approach to rule learning based on fusion of fuzzy logic and neural networks. *IEICE Transactions on Information Systems,* E78-d**,** 1509-1514.

LIN, C. T. 1995. A neural fuzzy control system with structure and parameter learning. *Fuzzy Sets and Systems,* 70**,** 183-212.

LIN, C. T. & LEE, C. S. G. Year. Real-time supervised structure/parameter learning for fuzzy neueral network. *In:* Proceedings of IEEE Conference of Fuzzy Systems, 1992. 1283-1290.

LINDSKOG, P. 1996. *Methods, Algorithms and Tools for System Identification Based on Prior Knowledge* PhD, Linkoping University.

LJUNG, L. 1987. *System Identification: Theory for the User*, Prentice Hall.

LJUNG, L. 1999. *System Identification: Theory for the User*, Prentice Hall.

LU, Y., SUNDARARAJAN, N. & SARATCHANDRAN, P. 1997. A sequential learning scheme for function approximation by using minimal radial basis function networks. *Neural Computation,* 8**,** 461-478.

MAMDANI, E. H. Year. Application of fuzzy algorithms for simple dynamic plant. *In:* IEE, 1974. 1585-1588.

MAMDANI, E. H. 1977. Application of fuzzy logic to approximate reasoning using linguistic systems. *Fuzzy Sets and Systems,* 26**,** 1182-1191.

MATTEUCCI, M. 2003. *Fuzzy C-Means Clustering* [Online]. Available: http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/cmeans.html [Accessed 30/04 2006].

MITCHELL, T. M. 1996. *An Introduction to Genetic Algorithms,* Cambridge, Mass, MIT Press.

MITCHELL, T. M. 1997. *Machine Learning*, McGraw-Hill.

MIYAMOTO, S., YASUNOBU, S. & IHARA, H. 1986. Predictive fuzzy control and application to automatic train operation system. *Analysis of fuzzy information,* 2**,** 59-72.

NAUCK, D. & KRUSE, R. 1999. Neuro-fuzzy systems for function approximation. *Fuzzy Sets and Systems,* 101**,** 261-271.

NOMURA, H., HAYASHI, I. & WAKAMI, N. Year. A self-tuning method of fuzzy control by descent method. *In:* 4th IFSA Congress, 1991. 155-158.

NOMURA, H., HAYASHI, I. & WAKAMI, N. Year. A learning method of fuzzy inference rules by descent method. *In:* FUZZ-IEEE 92, 1992. 203-210.

NOZAKI, K., ISHIBUCHI, H. & TANAKA, H. 1997. A simple but powerful method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems,* 86**,** 251-270.

OUYANG, C. S., LEE, W. J. & LEE, S. J. 2005. A TSK-type neurofuzzy network approach to system modeling problems. *IEEE Transactions on Systems, Man and Cybernetics,* 35**,** 751-767.

PAIVA, R. P. & DOURADO, A. 2004. Interpretability and learning in neuro-fuzzy systems. *Fuzzy Sets and Systems,* 147.

PAPPIS, C. P. & KARACATILIDIS, N. I. 1993. A comparative assessment of measures of similarity of fuzzy values. *Fuzzy Sets and Systems,* 56, 221-242.

PEDRYCZ, W. 1983. Numerical and applicational aspects of fuzzy relational equations. *Fuzzy Sets and Systems,* 11, 1-18.

PEDRYCZ, W. 1984. An identification algorithm in fuzzy relational systems. *Fuzzy Sets and Systems,* 13, 153-167.

PEDRYCZ, W. 1995. *Fuzzy Sets Engineering,* Boca Raton, CRC Press.

PEDRYCZ, W. 1996. Constriction fuzzy c-mean algorithm. *Pattern Recognition Letters,* 17, 625-631.

PEDRYCZ, W. 2002. Context fuzzy clustering in the design of radial basis function neural network. *IEEE Transactions on Neural Networks,* 9, 601-612.

PEDRYCZ, W. 2006. Linguistic models as a framework of user-centric system modeling. *IEEE Transactions on Systems, Man and Cybernetics - Part A,* 36, 727-745.

PLATT, J. 1991. A resource allocation network for function interpolation. *Neural Computation,* 3, 213-225.

QUINLAN, J. R. 1986. Induction of decision trees. *Machine Learning,* 1, 81-106.

RONG, H. J., SUNDARARAJAN, N., HUANG, G. B. & SARATCHANDRAN, P. 2006. Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction. *Fuzzy Sets and Systems,* 157, 1260-1275.

SETNES, M., BABUˇSKA, R., KAYMAK, U. & LEMKE, H. R. V. N. 1998. Similarity measures in fuzzy rules base simplification. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 28, 376-386.

SODERSTROM, T. & STOICA, P. 1989. *System Identification*, Prentice Hall International.

SUGENO, M. 1985. An introductory survey of fuzzy control. *Information Sciences,* 36, 59-83.

SUGENO, M. & KANG, G. T. 1988. Structure identification of fuzzy model. *Fuzzy Sets and Systems,* 28, 15-33.

SUGENO, M. & YASUKAWA, T. 1993. A fuzzy-logic based approach to qualitative modelling. *IEEE Transactions on Fuzzy Systems,* 1, 7-31.

SUGENO, S. & TANAKA, K. 1991. Successive identification of a fuzzy model and its applications to prediction of a complex system. *Fuzzy Sets and Systems,* 42, 315-334.

TAKAGI, T. & SUGENO, M. 1985. Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man and Cybernetics,* 15**,** 116-132.

TEKNOMO, K. 2004. *K-Means Clustering Tutorial* [Online]. Available: http://people.revoledu.com/kardi/tutorial/kMean/index.html [Accessed 30/05 2006].

TEKNOMO, K. 2006. *K-Means Clustering Tutorial, Kardi Teknomo's Tutorials* [Online]. Available: http://people.revoledu.com/kardi/tutorial/kMean/index.html [Accessed 30th April 2006].

TONG, R. M. 1979. The construction and evaluation of fuzzy models. *In:* GUPTA, M. M., RAGADE, R. K. & YAGER, R. R. (eds.) *Advances in fuzzy set theory and applications.* Amsterdam, The Netherlands: North-Holland.

TONG, R. M., BECK, M. B. & LATTEN, A. 1980. Fuzzy control of the activated sludge wastewater treatment process. *Automatica,* 16**,** 659-702.

TSEKOURAS, G. E. 2005. On the use of the weighted fuzzy C-means in fuzzy modeling. *International Journal of Advances in Enigeering Software,* 36**,** 287-300.

VIEIRA, J., DIAS, F. M. & MOTA, A. 2004. Neuro-Fuzzy Systems: A Survey. *The 5th WSEAS NNA International Conference on Neural Networks and Applications.* Udine, Italia.

WANG, D., ZENG, X. & KEANE, J. A. Year. An input-output clustering method for fuzzy system identification. *In:* Proceedings of 2007 IEEE International Conference on Fuzzy Systems, 2007 London, UK. 282-287.

WANG, D., ZENG, X. J. & KEANE, J. A. Year. An incremental construction learning algorithm for identification of TS Fuzzy Systems. *In:* Proceedings Of 2008 IEEE International Conference on Fuzzy Systems, 2008 Hong Kong. 1660-1666.

WANG, L. & LANGARI, R. 1996. Complex systems modeling via fuzzy logic. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 26**,** 100-106.

WANG, L. & YEN, J. 1999. Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter. *Fuzzy Sets and Systems,* 101**,** 353-362.

WANG, L. X. 1997. *A course in fuzzy systems and control,* Upper Saddle River, N.J., Prentice Hall PTR.

WANG, L. X. & MENDEL, J. M. 1992. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics,* 22**,** 1414-1427.

WIKIPEDIA. 2006a. *Curse of Dimensionality* [Online]. Wikipedia. Available: http://en.wikipedia.org/wiki/Curse_of_dimensionality [Accessed 30/04 2006].

WIKIPEDIA. 2006b. *Data Clustering* [Online]. Available: http://en.wikipedia.org/wiki/Data_clustering [Accessed 30th April 2006].

WU, S. & ER, M. J. 2000. Dynamic fuzzy neural networks-a novel approach to function approximation. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 30**,** 358-364.

WU, S., ER, M. J. & GAO, Y. 2001. A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. *IEEE Transactions on Fuzzy Systems,* 9**,** 578-594.

WU, T. P. & CHEN, S. M. 1999. A New Method for Constructing Membership Functions and Fuzzy Rules from Training Examples. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 29**,** 25-40.

XIE, X. & BENI, G. 1991. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13**,** 841-847.

XU, C. & LU, Y. 1987. Fuzzy model identification and self learning for dynamic systems. *IEEE Transactions on Systems, Man and Cybernetics,* 17**,** 683-389.

YEN, J. & WANG, L. 1998. Application of statistical information criteria for optimal fuzzy model construction. *IEEE Transactions on Fuzzy Systems,* 6**,** 362-371.

YEN, J. & WANG, L. 1999. Simplifying fuzzy rule-based models using orthogonal transformation methods. *IEEE Transactions on Systems, Man and Cybernetics - Part B,* 29**,** 13-24.

YEN, J., WANG, L. & GILLEPIE, C. W. 1998. Improving the interpretability of TSK fuzzy models by combining global learning and local learning. *IEEE Transactions on Fuzzy Systems,* 6**,** 530-537.

ZADEH, L. A. 1965. Fuzzy sets. *Inform and Control,* 8**,** 338-353.

ZADEH, L. A. 1971. Similarity relations and fuzzy orderings. *Information Sciences,* 3**,** 177-200.

ZADEH, L. A. 1973. Outline of a new approach to the analysis of complex systems and decision process. *IEEE Transactions on Systems, Man and Cybernetics,* 3**,** 28-44.

ZADEH, L. A. 1991. *From Circuit theory to system theory,* New York, Plenum Press.