Rule Model Simplification

Zhiheng Huang

Doctor of Philosophy

School of Informatics

University of Edinburgh

2006

Abstract

Due to its high performance and comprehensibility, fuzzy modelling is becoming more and more popular in dealing with nonlinear, uncertain and complex systems for tasks such as signal processing, medical diagnosis and financial investment. However, there are no principal routine methods to obtain the optimum fuzzy rule base which is not only compact but also retains high prediction (or classification) performance. In order to achieve this, two major problems need to be addressed. First, as the number of input variables increases, the number of possible rules grows exponentially (termed *curse of dimensionality*). It inevitably deteriorates the transparency of the rule model and can lead to over-fitting, with the model obtaining high performance on the training data but failing to predict the unknown data successfully. Second, gaps may occur in the rule base if the problem is too compact (termed *sparse rule base*). As a result, it cannot be handled by conventional fuzzy inference such as Mamdani.

This Ph.D. work proposes a rule base simplification method and a family of fuzzy interpolation methods to solve the aforementioned two problems. The proposed simplification method reduces the rule base complexity via *Retrieving Data from Rules (RDFR)*. It first retrieves a collection of new data from an original rule base. Then the new data is used for re-training to build a more compact rule model. This method has four advantages: 1) It can simplify rule bases without using the original training data, but is capable of dealing with combinations of rules and data. 2) It can integrate with any rule induction or reduction schemes. 3) It implements the similarity merging and inconsistency removal approaches. 4) It can make use of rule weights. Illustrative examples have been given to demonstrate the potential of this work.

The second part of the work concerns the development of a family of transformation based fuzzy interpolation methods (termed *HS methods*). These methods first introduce the general concept of representative values (RVs), and then use this to interpolate fuzzy rules involving arbitrary polygonal fuzzy sets, by means of scale and move transformations. This family consists of two sub-categories: namely, the *original* HS methods and the *enhanced* HS methods. The HS methods not only inherit the common advantages of fuzzy interpolative reasoning – helping reduce rule base complexity and allowing inferences to be performed within simple and sparse rule bases –

but also have two other advantages compared to the existing fuzzy interpolation methods. Firstly, they provide a degree of freedom to choose various RV definitions to meet different application requirements. Secondly, they can handle the interpolation of multiple rules, with each rule having multiple antecedent variables associated with arbitrary polygonal fuzzy membership functions. This makes the interpolation inference a practical solution for real world applications. The enhanced HS methods are the first proposed interpolation methods which preserve piece-wise linearity, which may provide a solution to solve the interpolation problem in a very high Cartesian space in the mathematics literature.

The RDFR-based simplification method has been applied to a variety of applications including nursery prediction, the Saturday morning problem and credit application. HS methods have been utilized in truck backer-upper control and computer hardware prediction. The former demonstrates the simplification potential of the HS methods, while the latter shows their capability in dealing with sparse rule bases. The RDFR-based simplification method and HS methods are further integrated into a novel model simplification framework, which has been applied to a scaled-up application (computer activity prediction). In the experimental studies, the proposed simplification framework leads to very good fuzzy rule base reductions whilst retaining, or improving, performance.

Acknowledgements

Many thanks to my supervisor, Prof. Qiang Shen, for his invaluable supervision and warm-hearted encouragement throughout this project. Without him, this Ph.D. and thesis would never have happened. Thanks must also go to Dr. Jacques Fleuriot, my second supervisor, for his supportive discussion during the research. In addition, many thanks to Prof. Laszlo Koczy, and all those in the Approximate and Qualitative Reasoning Group, especially Richard Jensen for lending his thesis to me to start, Michelle Galea and Khairul Rasmani for the useful discussions over the years. Finally, my sincere thanks to my beloved parents and my wife Yan for the unwavering support through all these years.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Zhiheng Huang*)

To my beloved wife Yan.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1965, Lotfi A. Zadeh of the University of California at Berkeley published "Fuzzy Sets," [Zad65, Zad73] which laid out the mathematics of fuzzy set theory and, by extension, fuzzy logic. Although, the technology was introduced in the U.S., U.S. and European scientists and researchers largely ignored it for years, perhaps because of its unconventional name. But fuzzy logic was readily accepted in Japan, China and other Asian countries.

Zadeh separated *hard computing* based on boolean logic, binary systems, numerical analysis and crisp software from *soft computing* based on fuzzy logic, neural nets and probabilistic reasoning. The former has the characteristics of precision and categoricity and the latter, approximation and dispositionality. Although in hard computing, imprecision and uncertainty are undesirable properties, in soft computing the tolerance for imprecision and uncertainty is exploited to achieve tractability, lower cost, high Machine Intelligence Quotient (MIQ) and economy of communication.

1.1 Soft Computing

The principal constituents of soft computing are fuzzy logic, artificial neural networks and probabilistic reasoning, with the latter subsuming belief networks, genetic algorithms etc. The principal contribution of fuzzy logic relates to its provision of a foundation for approximate reasoning, while neural network theory provides an effective methodology for learning from examples, and probabilistic reasoning systems furnish computationally effective techniques for representing and propagating probabilities and beliefs in complex inference networks.

Since the last decade, the research on fuzzy sets and systems has drawn more and more attention. In fact, fuzzy modelling [Zad65, Men95] is now one of the most famous ways in dealing with nonlinear, uncertain and complex systems such as signal processing and mechanical control [Sim00, RZK90, WM79, Jen04]. It has two important advantages: firstly, it imitates the human reasoning process using linguistic terms, which enables its comprehensibility and transparency; and secondly it is a universal modelling technique [WM92a, Buc93, Cas95, ZK04] that can approximate any nonlinear complex system with specified arbitrary accuracy.

Neural networks [Mit97] were developed as an attempt to realise simplified mathematical models of brain-like systems. The key advantage is their ability to learn from examples instead of requiring an algorithmic development from the designer. Compared to fuzzy logic, neural networks usually produce higher performance for classification or prediction tasks, however, they lack the transparency and comprehensibility which fuzzy logic has. Neural networks can be implemented as neuro-fuzzy networks which combine the advantages of both fuzzy reasoning and neural networks.

Belief networks (Bayes Nets, Bayesian Networks) [Pea88] are a vital tool in probabilistic modelling and Bayesian methods. They are one class of probabilistic graphical model. Genetic algorithms (GA) [Mit97] provide a technique useful for fining approximate solutions to optimization and search problems. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, natural selection, and recombination (or crossover).

1.2 Fuzzy Inference Systems

The first kind of fuzzy inference system (FIS) focused on the ability of fuzzy logic to model natural language [MA75]. These FISs contain fuzzy rules built from expert knowledge and they are called fuzzy expert systems or fuzzy controllers. These FISs offer a high semantic level and a good generalization capability. Unfortunately, the complexity of large systems may lead to an insufficient accuracy in the simulation results. Expert knowledge only based FIS may show poor performance.

The other class of FIS is a data-driven fuzzy system. The fuzzy rules are obtained from data rather than from the experts. Takagi-Sugeno-Kang (TSK) models [TS85, SK88] were the first attempt at this class of FIS. Since that piece of work, many methods [YS95, CLL01, WM92b, CZ97] have been designed to automatically generate rules from databases. A extensive discussion of classical fuzzy control and algorithms can be found in [DHR93, Ped92].

However, two major problems need to be addressed in order to obtain efficient and effective fuzzy models. First, as the number of input variables increases, the number of possible rules grows exponentially (termed *curse of dimensionality*). It inevitably deteriorates the transparency of the rule model and likely leads to over-fitting, with the model obtaining high performance on the training data but failing to predict the unknown data. Second, gaps may occur in the rule base if it is too compact (termed *sparse rule base*). As a result, it cannot be handled by conventional fuzzy inferences such as Mamdani.

1.3 Rule Base Simplification

The original motivation of rule base simplification, also called rule base optimization, is to conquer the curse of dimensionality [Gui01, KJS02]. If the induction methods are applied to simple systems with a few variables and/or a small quantity of data, there is no need for optimizing the rule base. The situation is different for large systems where many variables and/or tens of thousands of data are involved. The number of induced rules becomes enormous, resulting in a complex rule base. Obviously, the rule base will be easier to interpret if they are defined by the most influential variables and only consist of a small amount of rules. Feature selection and rule base reduction are thus two important issues of the rule generation process. They are usually referred to as structure optimization. Apart from that, many parameters such as membership functions parameters and rule conclusions can also be optimised, which is called parameter optimization [JSM97]. Unfortunately, parameter optimization inevitably leads to semantic loss if the fuzzy sets are predefined with particular physical meaning. This thesis focuses on a structure optimization. In particular, the rule base reduction methods are carefully reviewed and a new one is proposed towards the simplification goal.

On the other hand, attention is drawn to conquer the sparse rule base problem. When given observations have no overlap with the antecedent rule values, classical fuzzy inference methods have no rule to fire, but interpolative reasoning methods [KH93a] can still obtain certain conclusions. It thus facilitates fuzzy inferences when only limited knowledge is available. In addition, with interpolation, fuzzy rules which may be approximated from their neighboring rules can be omitted from the rule base [KH97]. This leads to the reduction of fuzzy models complexity.

1.4 Existing Simplification Approaches

The existing simplification approaches can be classified into categories using different criteria. In terms of the timing, they consist of three categories: the methods taking place before, within, and after the rule induction process. Traditionally, simplification methods are used after the induction process to refine the rule base to be more compact. However, due to the existence of noisy variables, or where the training schemes cannot handle a large quantity of variables, preprocessing of the data base must be done before the data is fed into the training schemes. This is usually called feature selection or feature transformation. Between the “after” and “before” stage simplification methods, there are some “within” stage simplifications [Qui87] which are integrated into the training schemes. They are, in fact, part of the training schemes. Once the training schemes are finished, the “reduced” rule bases are obtained without other processing. Despite the compact of the “within” stage simplification methods, they work depending on particular training schemes, thus cannot be reused between various training schemes. Therefore, “before” and “after” stage simplification methods are more desirable due to the generalization. This thesis focuses on the “after” stage simplifications. It studies the existing simplification methods and proposes a novel one.

Alternatively, in terms of the methodology, the existing fuzzy rule base simplification approaches are classified into five categories: 1) Feature based reduction, simplifying via preprocessing the original training data. 2) Similarity merging and inconsistency removal based reduction, merging similar rules and eliminating redundancy. 3) Orthogonal transformation based reduction, which includes the Orthogonal Least Square (OLS) method and Singular Value Decomposition (SVD). 4) Interpolative reasoning based reduction, which has the closest relevance to part of the work carried out in this project. 5) Hierarchical reasoning, which is based on the modification of rule base structure.

The main concern to choose a simplification method is the preservation of the semantic meaning. Otherwise, it is not worth using fuzzy modelling at all. Unfortunately, some simplification methods, such as the similarity merging and most of the transformation based methods, destroy the predefined fuzzy linguistic terms and hence result in loss of comprehensibility. The other concern to choose a proper simplification method is to avoid generating sparse rule bases. In fact it is highly likely that this will happen. Imagine that a fuzzy rule is eliminated in the reduction process, the data fired by this rule may no longer be fired by any other existing rules in the reduced rule base. Such sparse rule bases can be handled by fuzzy interpolative reasoning.

1.5 The Proposed Simplification Framework

This thesis proposes a rule base simplification method and a family of fuzzy interpolation methods to address the two concerns mentioned above. Firstly, in order to achieve a compact rule base under the conditions that no significant performance is sacrificed and no semantic meaning is destroyed, this thesis proposes a novel rule base simplification method via the technique of *retrieving data from rules* (RDFR). In particular, RDFR is carried out over the original rule sets to obtain new "training data". The new "training data" are then used for re-training to generate the final rule sets. Due to the flexibility of choosing the second rule induction algorithms, this in fact provides a general framed work to simplify the original given rule bases.

Secondly, in order to cope with the case that the simplified rule bases are sparse, a family of transformation based fuzzy interpolation methods has been proposed. These methods first introduce the general concept of representative values (RVs), and then uses them to interpolate fuzzy rules involving arbitrary polygonal fuzzy sets, by means of scale and move transformations. Compared to other existing fuzzy interpolation methods, this family offers a degree of freedom to provide a variety of unique, normal and valid results.

1.6 Thesis Structure

The rest of this thesis is structured as follows (with an indication of the publications produced as a result of this research):

- Chapter 2: *Background*. An overview of the existing simplification approaches, including feature based simplification, similarity merging, inconsistency removal, orthogonal based reduction, interpolative reasoning, and hierarchical reasoning are given. In particular, more detailed description is given to the interpolation methods which have the closest relation to this project.

- Chapter 3: *RDFR Based Simplification Method*. This chapter proposes a novel simplification method by means of *retrieving data from rules (RDFR)* procedure. It first retrieves a set of new data from an original rule base. Then the new data are re-trained to build a more compact rule model while maintaining a satisfactory performance. Illustrative examples are provided to demonstrate the success of this work. The contents of this chapter can be found in [HS05b].

- Chapter 4: *Transformation Based Interpolation: Specific Examples*. This chapter provides specific example studies of the proposed interpolative reasoning methods. In particular, the *representative values* are introduced and defined for the most widely used fuzzy terms (triangular, trapezoidal and hexagonal). This follows by the illustration of the interpolations via scale and move transformations. The contents of this chapter have been published in [HS03, HS04b].

- Chapter 5: *Transformation Based Interpolation: General Approach*. This chapter extends the work presented in chapter 4 so that the proposed family of interpolation methods can be applied to arbitrarily complex polygonal fuzzy sets with flexible RV definitions. A family of enhanced interpolation methods has been further developed which not only reduces the computation efforts but also preserves piecewise linearity (see chapter 6). The interpolation and extrapolation involving multiple antecedent variables and multiple rules have been extended. Partial contents of this chapter have been published in [HS05d, HS04a, HS06, HS05c].

- Chapter 6: *Transformation Based Interpolation: Evaluations*. This chapter compares the interpolative reasoning methods proposed in Chapter 5 to other existing approaches such as the first proposed fuzzy interpolation method (KH) [KH93a] and the general method, in terms of the dependency of the fuzziness of the conclusion on the observation, the preservation of the piecewise linearity and the computational complexity.

- Chapter 7: *Transformation Based Interpolation: Realistic Applications*. This chapter presents the interpolation based fuzzy rule base inference and demonstrates its usages on both simplifying fuzzy rule bases and facilitating fuzzy inferences. Partial contents of this chapter can be found in [HS05a].

- Chapter 8: *Scaled-up Applications*. This chapter demonstrates the combination of the RDFR-based rule base simplification and the proposed interpolation based inference in a real world database (computer activity). Despite the large quantity of variables and data, the proposed framework leads to very good reductions. The results between various interpolation methods are thoroughly compared.

- Chapter 9: *Conclusion*. The thesis is concluded in this chapter, and details of future work to be carried out in this area are presented.

Chapter 2

Background

It becomes difficult for conventional classifiers to handle massive databases. Therefore, fuzzy rule base simplification methods are desirable to resolve this problem. These methods usually consist of two categories: parameter and structure simplification methods. The former refer to the optimization of the membership functions (either the conditional or conclusion one). Although they are widely used to fine tune the fuzzy sets [JSM97], they are not included in this thesis as they inevitably destroy the semantic meaning. The latter consist of feature selection and rule base reduction (which is the concern of this thesis). This chapter reviews the existing fuzzy simplification techniques including feature-based reduction, merging and removal-based reduction, orthogonal transformation based methods, interpolative reasoning methods and hierarchical fuzzy reasoning. The comparisons between different simplification techniques are also summarised.

2.1 Feature Based Reduction

As machine learning tools become more and more important to help extract and manage knowledge, they must meet many challenges such as handling massive amounts of data. The situation becomes worse if each datum has many features (or variables). One way to resolve this is to choose a set of informative features before feeding data to machine learning tools. This technique is called feature based reduction, which has two sub-categories: feature transformation and feature selection [LM98]. Feature transformation constructs additional features from the given ones or extracts a set of new features to replace the old ones. The former does not help simplify the dataset while the latter does so by generating low dimensional data. This approach changes the physical meaning of features, and hence may be criticized as losing semantics. Feature selection on the other hand overcomes this shortcoming by selecting a subset of the most influential features.

Feature based reduction is no doubt an important component of fuzzy model simplification. However, it is not the focus of this PhD work which attempts to simplify fuzzy rule models after the rule induction process. Nevertheless, in order to provide a complete overview of fuzzy model simplification, a brief explanation of feature based reductions are presented in following subsections.

2.1.1 Feature Transformation

Feature transformation reduces the dimensionality of the data so that the analysis becomes less difficult. A typical method of feature transformation is Sammon's nonlinear projection [Sam69]. It maps high dimensional data to low dimensional ones while keeping the underlying data structure. In particular, suppose N vectors in an L-space $\mathbb{R}^L$ (which is Euclidean space of dimensionality L) are denoted as X_i, $i = 1, \ldots, N$. The goal is to construct N vectors Y_i, $i = 1, \ldots, N$ which correspond to X_i, but in d-space $\mathbb{R}^d$ ($d = 2$ or 3). Let the distance between the vectors X_i and X_j in $\mathbb{R}^L$ be denoted as $d^{\star}_{ij} = dist[X_i, X_j]$, and likewise, the distance between the corresponding vectors Y_i and Y_j in $\mathbb{R}^d$ as $d_{ij} = dist[Y_i, Y_j]$. The projection begins with randomly initialized N vectors in $\mathbb{R}^d$. A steepest descent procedure is then utilized to search for a minimum of error defined as

$$E = \frac{1}{\sum_{i<j}[d^{\star}_{ij}]} \sum_{i<j}^{N} \frac{[d^{\star}_{ij} - d_{ij}]^2}{d^{\star}_{ij}}. \tag{2.1}$$

Extensions to Sammon's method can be found in [MJ92, MJ95, PEM02, PE98]. There are many other methods such as Principal Components Analysis [Jol86] and Multidimensional Scaling [Tor52] which act the same as Sammon's method – determine the Euclidean structure of a dataset's internal relationships in a low dimension.

These methods effectively reduce the complexity of the training data with little or no information loss, but suffer from the loss of the models' physical meaning.

2.1.2 Feature Selection

Feature selection is defined as the problem of finding a minimum set of M features from N original ones ($M \leq N$). This is essential since in real life there are irrelevant or noisy features which do not significantly contribute to the systems considered. Elimination of these features will speed up the learning procedure or resolve the problem that many learning applications cannot work very well with a huge data base. In addition, it may lead to more general models. The generality here stands for that the outcoming model may obtain less performance in the training stage, but lead to higher performance when tested with unknown data. Because of these merits, feature selection has long been the focus of research in pattern recognition and statistics. A detailed review of feature selection techniques devised for classification tasks can be found in [DL97, BL97, KS96].

The basic idea of feature selection is to search an optimal set of useful features using some criteria (or evaluations). As it is not practical to carry out exhaustive searches for most datasets, heuristic is often used to guide the processes: the evaluation functions test if the selected features are sufficient to represent the underlying models. Feature selection algorithms may be classified into two categories based on their evaluation procedure. If an algorithm performs independently of any learning algorithm (i.e. it is a completely separate preprocessor), then it is a filter approach. RELIEF [KR92] and FOCUS [AD91] etc. fall in this category. In effect, irrelevant attributes are filtered out before induction. Filters tend to be applicable to most domains as they are not tied to any particular induction algorithm.

If the evaluation procedure is tied to the task (e.g. classification) of the learning algorithm, the feature selection algorithm employs the wrapper approach. For instance, the LVF [LS96b], LVW [LS96a], the neural network-based wrapper feature selector [SL97] and the rough and fuzzy sets based feature selector [JS04a, Jen04] belong to this family. These methods search through the feature subset space using the estimated accuracy from the induction algorithms as measures of subset suitability. Although wrappers may produce better results, they are expensive to run and can break down with a very large number of features.

For illustration, the information gain based feature selection [Qui86] is briefly explained. Given a classification problem with training example collection S which has c target classes, to measure the purity of this collection, *entropy* is defined as follows:

$$Entropy(S) = \sum_{i=1}^{c} -P_i \log_2 P_i, \tag{2.2}$$

where P_i is the proportion of S which belongs to class i.

A statistical property *information gain* is used to represent how well a given feature separates training examples into target classes. It is the expected reduction in entropy caused by partitioning the examples according to this feature. Information $Gain(S,A)$ can be defined as

$$Gain(S,A) = Entropy(S) - \sum_{\nu \in Values(A)} \frac{|S_\nu|}{|S|} \times Entropy(S_\nu), \tag{2.3}$$

Where:
$Value(A)$ is the set of all possible values of feature A,
S_ν is the subset of S for which feature A has value ν,
$|S_\nu|$ is the number of examples in S_ν,
$|S|$ is the number of examples in S.
This measurement can be used to determine which features should be retained, by keeping those whose information gains are greater than a predetermined threshold value.

2.2 Merging and Removal Based Reduction

In fuzzy rule-based models there may exist similar, inconsistent and inactive rules. Similar rules have almost the same meaning so they can be combined into one. Inconsistent rules are contrary to each other in reasoning and hence destroy the logical consistency of the models. Inactive rules contribute little to the models since they are not frequently used. All these rules inevitably result in unnecessary complexity and therefore make models harder to understand. To tackle these problems respectively, compatible cluster merging algorithms [KF92, KB95, BV95, BRV98] have been proposed for the cluster based rule induction issues. Sharing the same idea but not explicitly limited to clustering algorithms, similarity merging methods have been reported [CCT96, SBKL98, Jin99]. These methods eliminate the redundancy by combining similar rules into one. Also, methods for consistent checking [XL02] and inactive evaluation [Jin99] have been proposed. Consistent checking simplifies rule bases by removing conflicting rules, while inactive checking removes rules having lower firing strengths than a predetermined threshold.

2.2.1 Similarity Based Merge

The compatible cluster merging algorithm [KB95] is based on the work of Krishnapuram and Freg [KF92]. It first defines cluster i, $i = 1, \dots, m$, as eigenvalues $\lambda_{i1}, \dots, \lambda_{in}$ and eigenvectors $\phi_{i1}, \dots, \phi_{in}$, which stand for the axis lengths and axis directions respectively. Then every pair of clusters, say cluster i and cluster j, are examined by the following criteria:

$$|\phi_{in} \cdot \phi_{jn}| \geq k_1, k_1 \text{ close to } 1, \tag{2.4}$$

$$\|c_i - c_j\| \leq k_2, k_2 \text{ close to } 0. \tag{2.5}$$

Equation (2.4) states that the parallel hyper-plane clusters should be merged. Equation (2.5) states that the cluster centres should be sufficiently close for merging. According to these two criteria, two matrices $C1[c1_{ij}]$ and $C2[c2_{ij}]$ whose elements indicate the degree of similarity between the ith and jth clusters measured are obtained. By considering the combination of these two criteria, the geometric mean has been used as a decision operator:

$$\mu_{ij} = \sqrt{c1_{ij}c2_{ij}}. \tag{2.6}$$

At this point a similarity matrix S has been achieved. This matrix has a predetermined problem-dependent value γ as a threshold. That is to say, any two clusters with a similarity more than γ should be merged.

Sharing the same idea, Chao, Chen and Teng [CCT96] utilize fuzzy similarity-based merging. They first derive simple triangular approximate equations from Gaussian-shaped fuzzy sets. Then the measurement between two triangular fuzzy sets is proposed, and similar linguistic terms are merged into one. This indirectly results in decreasing of the number of rules. In particular, they use the following fuzzy similarity measure on two fuzzy sets:

$$E(A_1,A_2) = \frac{A_1 \cap A_2}{A_1 \cup A_2}, \tag{2.7}$$

where $\cap$ and $\cup$ denote the intersection and union of fuzzy sets A_1 and A_2, respectively. According to the definition, $0 \leq E(A_1,A_2) \leq 1$. To make computation simple, a triangular function is employed to approximate a Gaussian function. Thus the similarity measure of two Gaussian fuzzy sets can be directly applied by using the approximation equations. Note that in the event of complicated fuzzy shapes, this measurement may be computed as

$$S(A,B) = \frac{\sum_{j=1}^{m}[\mu_A(x_j) \wedge \mu_B(x_j)]}{\sum_{j=1}^{m}[\mu_A(x_j) \vee \mu_B(x_j)]}, \tag{2.8}$$

where $j = 1,\ldots,m$ are the intervals discretized in the variable domain.

Similarity measure on rules follows similarity measure on conditions and consequences. In conditions, the smallest similarity between a variable-pair is chosen as the similarity of conditions E_p, while in consequences, the similarity E_c is discretized to either 1 or 0 to indicate whether the conclusion is almost the same or not. If both $E_c = 1$ and $E_p \geq \gamma$ hold (γ is a reference value set by users), i.e., the two fuzzy rules have almost the same consequences and the degree of similarity on the conditions is high enough, these two rules are combined into one.

The work reported in [SBKL98] follows the same similarity merging procedure except that different fuzzy modelling techniques (Gustafson-Kessel and fuzzy c-means algorithms) are used. In addition, a similarity measure on trapezoid functions is used instead of triangular ones as described in [CCT96]. After all the fuzzy sets and fuzzy rules are merged, to improve the accuracy of the simplified model, a fine-tuning procedure for parameters that define fuzzy sets is executed using the gradient-descent algorithm.

The similarity measure can be divided into two main groups [SBKL98]: one is set-theoretic based and the other is geometric based. Set-theoretic based measurements are the most suitable for capturing similarity among overlapping fuzzy sets. The geometric based measurements represent fuzzy sets as points in a metric space and the similarity between the sets is regarded as an inverse of their distance in this metric space. Based on the set-theoretic operations of intersection and union, the similarity between fuzzy sets is defined as Eqn. (2.7). The work of [CCT96, SBKL98] implement set-theoretic based similarity measurements. As an example of geometric based similarity merging methods, Jin [Jin99] makes use of the distance concept rather than the set operations. Assuming a Gaussian fuzzy function is given as follows:

$$A_i(x) = e^{-\frac{(x-a_i)^2}{2b_i^2}}, \tag{2.9}$$

then the similarity of fuzzy subsets A_1 and A_2 is

$$S(A_1, A_2) = \frac{1}{1 + d(A_1, A_2)}, \tag{2.10}$$

where $d(A_1, A_2)$ is the distance between two fuzzy subsets A_1 and A_2:

$$d(A_1, A_2) = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}. \tag{2.11}$$

Another distinction to the previous methods [CCT96, SBKL98] is that, in Jin's work, the similarity measure makes use of the training data. In particular, [Jin99] refines a given rule model by means of the gradient learning algorithm. During the iteration an extra penalty term γ, which stands for the similarity of the fuzzy sets for all variables, is added to the conventional error function to drive learning. Therefore, the modification of the parameters depends not only on the system error but also on how the similar fuzzy subsets converge to the same one. The parameter γ plays a very important role in the refining stage. If γ is too large then the similar fuzzy subsets will be merged quickly but the system performance may become seriously worse. On the contrary, if γ is too small, then the system performance will be good but the similar fuzzy subsets may remain indistinguishable and the interpretability of the fuzzy model becomes poor.

2.2.2 Inconsistency Based Removal

Inconsistent rules have similar conditions but different consequences. It is essential for learning mechanisms to identify possible conflicts in rule bases and to obtain good logical coherence. For this purpose, Xiong and Litz [XL02] have introduced a numerical assessment named "consistency index", which helps establish the consistency/inconsistency of rule bases. This index is integrated into the fitness function of a

GA to search for a set of optimal rule conditions through two criteria: 1) the encoded fuzzy model has good accuracy; and 2) the rule base has little or no inconsistency.

2.3 Orthogonal Transformation Based Methods

Orthogonal transformation based methods simplify rule bases via matrix computation [GL83]. The first work in this field was proposed some ten years ago and research along this line has become considerably active. These methods either work on a *firing strength matrix* [CCG91, WM92a, WL95, NMM96, MM96, YW96] and employ some measure index to estimate the importance of rules, or work on the *fuzzy rule consequences matrix* [Yam97, YBY99] and construct new fuzzy rule bases in terms of newly constructed fuzzy sets. Briefly, the *firing strength matrix* based methods [YW97, YW99] include an orthogonal least squares (OLS) method [CCG91, WM92a, WL95], an eigenvalue decomposition (ED) method [NMM96], a singular value decomposition with column pivoting (SVD-QR) method [MM96], and a pure singular value decomposition (SVD) method [YW96]. The *rule consequences matrix* based methods have been recently attempted by Yam and his colleagues [Yam97, YBY99]. In order to give a flavour of these methods, this chapter outlines two typical methods from the above two categories: the orthogonal least square method and the rule consequences matrix based SVD method.

2.3.1 Orthogonal Least Square Method

The OLS algorithm is a one-pass regression procedure [CCG91]. It is able to generate a robust fuzzy model which is not sensitive to noisy inputs. Chen, Cowan and Grant [CCG91] have first provided an OLS method for the solution of radial basis function (RBF) networks.

The OLS algorithm can be used to select RBF centres so that adequate and parsimonious RBF networks can be obtained. In this algorithm, an RBF network is treated as a special case of the linear regression model:

$$d(t) = \sum_{i=1}^{M} p_i(t)\theta_i + \varepsilon(t) \tag{2.12}$$

where $d(t)$ is the desired output, θ_i are weight parameters, and the $p_i(t)$ is known as the regressor which is a certain fixed function of $x(t)$:

$$p_i(t) = x(t), \tag{2.13}$$

the error signal $\varepsilon(t)$ is assumed to be uncorrelated with the regressors $p_i(t)$. The problem of how to choose a suitable set of RBF centres from the dataset can be regarded as how to choose a subset of significant regressors (basis vectors) from a given candidate set.

The geometric interpretation of the LS method is best revealed by arranging (2.12) for $t = 1$ to N in the following matrix form:

$$d = P\theta + E, \tag{2.14}$$

where

$$d = [d(1), \ldots, d(N)]^T, \tag{2.15}$$

$$P = [p_1, \ldots, p_M],\ p_i = [p_i(1), \ldots, p_i(N)]^T,\ 1 \leq i \leq M, \tag{2.16}$$

$$\theta = [\theta_1, \ldots, \theta_M]^T, \tag{2.17}$$

$$E = [\varepsilon(1), \ldots, \varepsilon(N)]^T. \tag{2.18}$$

The regressor vectors p_i form a set of basis vectors and the LS solution θ satisfies the condition that θ be the projection of d onto the space spanned by these basis vectors p_i.

The OLS method involves the transformation of the set of p_i into a set of orthogonal basis vectors, and thus makes it possible to calculate the individual contribution to the desired output of each basis vector. The regression matrix P can be decomposed into

$$P = WA \tag{2.19}$$

where A is an $M \times M$ upper triangular matrix,

$$\mathbf{A} = \begin{pmatrix} 1 & \alpha_{12} & \alpha_{13} & \ldots & \alpha_{1M} \\ 0 & 1 & \alpha_{23} & \ldots & \alpha_{2M} \\ 0 & 0\ldots & & & \\ \vdots & & & & \vdots \\ & & \ldots & 1 & \alpha_{M-1M} \\ 0 & \ldots & 0 & 0 & 1 \end{pmatrix},$$

and W is an $N \times M$ matrix with orthogonal columns w_i. The space spanned by the set of orthogonal basis vectors w_i is the same space spanned by the set of p_i, and (2.14) can be rewritten as

$$d = Wg + E. \tag{2.20}$$

The orthogonal LS solution g is given by

$$g = \frac{w_i^T d}{w_i^T w_i} \; 1 \leq i \leq M \tag{2.21}$$

The quantities g and θ satisfy the triangular system

$$A\theta = g. \tag{2.22}$$

The classical Gram-Schmidt algorithm [GL83] can be used to derive the above equation and thus to compute the LS estimate θ. The OLS method further provides the regressors subset selection. In the case of RBF networks the number of data points $x(t)$, N, is often very large and the centres are chosen as a subset of the dataset. In general, the number of all the candidate regressors, M, can be very large and an adequate modelling may only require $M_s (\ll M)$ significant regressors. These significant regressors may be selected using the OLS algorithm by operating in a forward regression manner.

The geometric interpretation of this OLS procedure is obvious. Since the original basic vectors p_i are correlated, it is hard to calculate their individual contribution to the variance of the output variable. In order to solve this problem, orthogonal basic vectors are calculated, to reflect the independent contributions. During the computation, the dimension of the space spanned by the selected regressors is increased one by one. The newly added regressor maximises the increment to the expected variance of the output variable. Orthogonality ensures that these selected rules do not have similar conditions. Therefore, the reduced fuzzy model does not have the similarity or inconsistency problems. Essentially, the OLS attempts to select the important fuzzy rules based on their contributions to the variance of the output. This is quite similar to the strategy of selecting components in *principal component regression* [Jol86, Rog71], where those components with large variances are retained in the regression model.

Wang and Mendel [WM92a] have proposed OLS on fuzzy basis functions rather than radial basis functions [CCG91]. Although these two methods use the same technique, the work of [WM92a] is the first designed for the purpose of fuzzy rule base simplification. In order to understand how it works, the inference formulae are first defined as follows.

Definition 1 *The fuzzy systems with singleton fuzzifier, product inference, centroid defuzzifier, and Gaussian membership functions have the inference*

$$f(x) = \frac{\sum_{j=1}^{M} z^j (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))}{\sum_{j=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))}, \tag{2.23}$$

where $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R},\ x = (x_1, x_2, ..., x_n) \in U;\ \mu_{A_i^j}(x_i)$ *is the Gaussian membership function, defined by*

$$\mu_{A_i^j}(x_i) = a_i^j e^{-\frac{(x_i - x_i^j)^2}{2\sigma_i^{j2}}}, \tag{2.24}$$

where a_i^j, x_i^j, *and* σ_i^j *are real-valued parameters with* $0 \leq a_i^j \leq 1$, *and* z^j *is the point in the output space R at which* $\mu_{A^j}(z)$ *achieves its maximum value.*

Definition 2 *Define fuzzy basis functions (FBFs) as*

$$p_j(x) = \frac{\prod_{i=1}^{n} \mu_{A_i^j}(x_i)}{\sum_{j=1}^{M} \prod_{i=1}^{n} \mu_{A_i^j}(x_i)}, j = 1, 2, \ldots, M, \tag{2.25}$$

where $\mu_{A_i^j}(x_i)$ *are the Gaussian membership functions (2.24).*

The fuzzy system (2.23) is equivalent to an FBF expansion:

$$f(x) = \sum_{j=1}^{M} p_j(x)\theta_j \tag{2.26}$$

where $\theta_j \in R$ are constants. From (2.25), an FBF corresponds to a fuzzy IF-THEN rule. Note that this method is different from the work of [CCG91] as it uses (2.25) rather than (2.13).

As the numerator of (2.25) gives the degree to which a particular rule fires (the product implements the AND operation) and the denominator gives the sum of the degrees for all rules, (2.25) thus normalises the fire strength of one particular rule over one particular datum in the range $[0,1]$. Due to this normalization, each FBF is calculated upon the whole FBF base. The work of [WM92a] can successfully reduce to M_s rules from the original M rules ($M_s < M$). However, it is incorrect to believe that the fuzzy model can be maximally reduced because the denominator of each FBF in (2.25) contains all rules' contribution, including rules belonging to the $M - M_s$ non-selected FBF's. To tackle this problem Hohensohn and Mendel [HM95] have proposed a two-pass orthogonal least-square algorithm. The first run remains the same as in [WM92a]. After that, only those selected FBFs are kept together (with the respective antecedent parameters (x_i, σ_i) recorded). They are now only normalised by $\sum_{j=1}^{M_s} \prod_{i=1}^{n} \mu_{F_i^j}(x_i)$. That is, the first run of the OLS is used to choose the number of FBF's, but not the final θ parameters in (2.26). The next step runs OLS again to determine θ based on only M_s FBFs. This run is much faster than the first since usually $Ms \ll M$. Note that there is no need to use the same training samples in the second run of OLS as in the first run. In order to obtain a precise model, the second run may use a much larger training set than that used in the first run without requiring too much computation.

Sugeno-Type models [TS85] have also been attempted by means of the OLS method [WL95]. The only difference is that in the previous methods there are θ_i, $i = 1, 2, \ldots, M_s$, needing to be calculated, but in Sugeno-Type model, there are θ_i, $i = 1, 2, \ldots, (r+1) * M_s$ (r is the number of input variables) needing to be identified. The computation process is the same except for the size of identified parameters.

2.3.2 Consequence Matrix Based Singular Value Decomposition

Unlike fire strength matrix based methods, Yam and his colleagues [Yam97, YBY99] have applied SVD to the rule consequence matrix which describes the outputs of a rule set. The idea is to transform the original membership functions for each variable into a fewer number of membership functions. It amounts to the reduction of the fuzzy model since the rule number is determined by the possible combination of fuzzy sets of each variable. In particular, consider the rule sequence $\mathcal{F}$ in the SVD form

$$\mathcal{F} = U \Sigma V^T, \tag{2.27}$$

where $\mathcal{F}$ is $n_a \times n_b$, and U and V are $n_a \times n_a$ and $n_b \times n_b$, respectively. Similar to the matrix computation discussed above, a close approximation to $\mathcal{F}$ can be obtained by keeping those components having large singular values. Let n_r be the number of singular values to keep, the approximation becomes

$$\mathcal{F} = U^{(r)}\Sigma^{(r)}{V^{(r)}}^T, \tag{2.28}$$

where $U^{(r)}$ is $n_a \times n_r$ and $V^{(r)}$ is $n_r \times n_b$. The essential idea is to construct new fuzzy sets $f_j(x), j = 1, \ldots, r$ of variable x through the original fuzzy sets $f_i(x), i = 1, \cdots, n_a$ and matrix U

$$f_j(x) = \sum_{i=1}^{n_a} f_i(x) U_{i,j}. \tag{2.29}$$

The number of fuzzy sets of variable x decreases from n_a to n_r. Likewise, all other variables have a decreased number of fuzzy functions. The number of possible rules generated from these functions are hence significantly reduced.

However, from (2.29) the validity of the new fuzzy functions in terms of normality and nonnegativeness cannot be guaranteed. To support the discussion, the properties of Sum Normalization (SN), Nonnegativeness (NN) and Normality (NO) are introduced.

Definition 3 Sum Normalization (SN)*: A set of functions* $f_i(x),\ i = 1, \ldots, m$*, is SN if for any value of x within the domain of interest*

$$\sum_{i=1}^{m} f_i(x) = 1. \tag{2.30}$$

Without ambiguity, a matrix F *is SN if*

$$sum(F) = [1, \cdots, 1]^T, \tag{2.31}$$

where $sum(F)$ *denotes the column vector obtained by summing over the rows of matrix* F*.*

Definition 4 Nonnegativeness (NN)*: A set of functions* $f_i(x),\ i = 1, \ldots, m$*, is NN if for any value of x within the domain of interest*

$$f_i(x) \geq 0, \tag{2.32}$$

for each $i = 1, \ldots, m$*. Likewise, a matrix* F *is NN if every one of its elements* $F_{i,j}$ *is greater than or equal to zero.*

Definition 5 Normality (NO)*: A set of functions $f_i(x)$, $i = 1, \ldots, m$, is NO if it is SN and NN and each of the functions $f_i(x)$ attains the value of 1 at least on one point within the domain x. Correspondingly, a matrix F is NO if it is SN and NN and each of its column contains the value 1 as an element.*

A theorem follows these three definitions:

Theorem 1 *Given a set of function $f_i(x)$, $i = 1, \ldots, m$, and a matrix F of dimension m by q, and q new functions $f_j(x)$, $j = 1, \ldots, q$, such that*

$$f_j(x) = \sum_{i=1}^{m} f_i(x) F_{i,j}. \tag{2.33}$$

Then

1. *the set of functions $f_j(x)$ is SN if $f_i(x)$ and F are SN;*
2. *the set of functions $f_j(x)$ is NN if $f_i(x)$ and F are NN;*
3. *the set of functions $f_j(x)$ is NO if $f_i(x)$ and F are NO.*

Since the U in (2.29) is in general neither SN nor NN, the work presented in [Yam97, YBY99] gives the mathematical procedure for converting U into SN and NN matrices, and possibly, a NO matrix.

This method has been successfully implemented in [Sim00]. However, there are several points worth noting [Tao01]: 1) the approach for determining the number of singular values is not provided; 2) the performance is not always satisfactory; and 3) the computational load is increased for each input since the membership functions are modified.

From the semantic perspective, it is easy to see that the fire strength matrix based methods are semantic-keeping since they simplify the fuzzy model by selecting the important rules, while the consequence matrix based methods are semantic-losing since they construct new fuzzy sets, which have different physical meaning from those predefined.

2.4 Interpolative Reasoning Methods

Conventional fuzzy reasoning methods such as Mamdani [MA75] and TSK [TS85, SK88] require that the rule bases be *dense*. That is, the input universe of discourse is covered completely by the rule bases. When an observation occurs, a consequence can always be derived by using such dense rule bases. On the contrary, if fuzzy rule bases are *sparse*, that is, the input universes of discourses may not be covered completely by the rule bases, the conventional fuzzy reasoning methods encounter difficulties if an observation occurs in a gap, resulting in no rule fired and thus no consequence derived. This problem was initially proposed by Mizumoto and Zimmerman [MZ82] as the *tomato problem*, which is shown in Equation (2.34) and Fig. 2.1.

$$\begin{array}{l} \text{observation: This tomato is yellow} \\ \text{rules: if a tomato is red then the tomato is ripe} \\ \qquad\quad \text{if a tomato is green then the tomato is unripe} \\ \hline \text{conclusion: ???} \end{array} \tag{2.34}$$

The intuitive consequence by the human being would be that this tomato is half ripe.

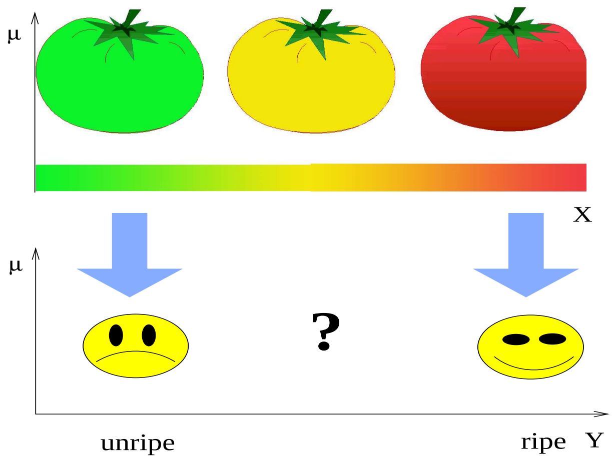

Figure 2.1: Fuzzy reasoning of tomato problem

However, none of the conventional fuzzy inferences is able to reach such a conclusion. Motivated by this, Kóczy and Hirota have proposed the first fuzzy interpolative reasoning method, termed the KH method [KH93a, KH93c, KH97, KHG97, KHM00, KHM91, KH93d].

In addition to support reasoning on sparse rule bases [BB92], fuzzy interpolation can be used to simplify the complexity of fuzzy rule bases [KH97] by eliminating the fuzzy rules which may be approximated from their neighboring rules. This potential opens a new door to tackle rule base simplification problems.

Despite these significant advantages, earlier work in fuzzy interpolative reasoning does not guarantee validity of the derived fuzzy sets [KK93, KK94b, KK94a, KK94c, YMQ95, SM95, KC96]. In fuzzy interpolation literature, the validity can be defined as follows.

Definition 6 Validity*: A fuzzy set described by the membership function* $f(x)$*, is valid if for any value of x within the domain of interest, it has only one corresponding fuzzy membership value* $f(x)$ *(with* $0 \leq f(x) \leq 1$ *normally assumed, which is always presumed throughout this thesis).*

Based on this definition, fuzzy set B^* as shown in Fig. 2.2 is invalid as it may have two different fuzzy membership values corresponding to one input value. In order to eliminate the drawback of invalidity, there has been considerable work reported in the literature. In terms of the methodology, this work is roughly divided into two categories: the α*-cut based interpolations* and the *intermediate rule based interpolations*.

The α-cut based interpolations infer the results based on the computation of each α-cut level. The KH method [KH93a, KH93c, KH97, KHG97, KHM00, KHM91, KH93d] is a typical α-cut based interpolation. Further development and modification has been carried out. For instance, Vas, Kalmar and Kóczy have proposed an algorithm [VKK92] that reduces the problem of invalid conclusions. Gedeon and Kóczy [GK96] have enhanced the original KH method. Dubois and Prade [DP92, DPG95, DP99] have operated all possible distances among the elements of fuzzy sets at each α-level and computed all conclusions for the same α-level. Tikk and Baranyi [TB00, Tik99, BTYK99a, BTYK99b, YBTK99, WGF00] have presented a modified α-cut based method which changes the coordinates when applying the KH method, and Tikk *et al.* have shown that the modified method inherited the approximation stability of the KH interpolation [TkM97, TBYk99, JKTV97]. This method was further extended in

[TBGM01]. One of the recent methods [WGF00, WGT00] have used the combination of different interpolation techniques. Despite the rapid development of the α-cut based fuzzy interpolations, there is a drawback in this group. The α-cut based interpolations should consider all possible α-cuts (an infinite number) in performing the interpolation. However, all the previously listed methods only take a finite number of α-cuts (usually 3 or 4) into consideration. The resulting points are then connected by linear pieces to yield an approximation of the accurate conclusion.

Intermediate rule based interpolations infer the results by reasoning an intermediate rule (together with the observation of course) rather than the given two original fuzzy rules. In particular, an intermediate fuzzy rule is generated by the given two rules before the interpolation process. The antecedent of the generated intermediate rule is expected to be very close to the given observation. Thus, the interpolation problem actually becomes the similarity reasoning [DP92, DSM89, DSM92]: *the more similar between the observation and an antecedent, the more similar conclusion must be concluded to the corresponding consequent set*. This semantic interpretation is in fact the extended version of the analogical inference which was proposed by Turksen [TZ88]. Within this category, Hsiao, Chen and Lee [HCL98] have introduced a new interpolative method which exploits the slopes of the fuzzy sets to obtain valid conclusions. Qiao, Mizumoto and Yan [QMY96] have published an improved method which uses similarity transfer reasoning to guarantee valid results. Baranyi *et al.* [BGK95, BK96a, BGK96, BG96, BK96b, BMK$^+$98, BKG04] have proposed general fuzzy interpolation and extrapolation techniques. Kawaguchi *et al.* [KMK97, KM98, KM00a, KM00b] have developed the B-spline based fuzzy interpolation from the semantic point of view.

Various further research has been reported in the fuzzy rule interpolation area. Kovás *et al.* have proposed an interpolation technique based on the approximation of the vague environment of fuzzy rules and applied it in the control of an automatic guided vehicle system [KK97c, KK97b, KK97a]. Bouchon, Marsala and Rifqi have created an interpolative method based on graduality [BMDM$^+$99, BM00, BMMR00, BMDM$^+$01]. Jenei [Jen01, JKK02] has suggested an axiomatic approach of fuzzy quantities interpolation and extrapolation. Bouchon-Meunier has proposed a comparative view of fuzzy interpolation methods in [BMDM$^+$01]. In addition, Yam *et al.* has introduced a Cartesian based interpolation in [YK98, YK00, YWB00, YKN00, YK01], where each fuzzy set is mapped into a point in high dimensional Cartesian space. This method can produce multiple interpolation results but it doesn't show how to choose a proper one. Also, this approach is restricted to a finite number of characteristic points. For a brief review of the available fuzzy interpolation techniques, interested users may refer to [MBKK99, MSK99, Miz01].

For the purpose of comparing the existing typical interpolation methods and the newly proposed one (see chapter 4 and 5), this section outlines the simplest case (the triangular case) of the KH interpolation [KH93a, KH97], the HCL interpolation [HCL98], the general interpolation [BGK95, BKG04] and the QMY interpolation [QMY96] methods. It is easy to spot that the KH method may lead to invalid fuzzy sets and the HCL is limited to the interpolation of triangular fuzzy sets. However, the drawbacks of the general interpolation and the QMY will only be identified in the real-life experiments (see chapter 8).

2.4.1 The KH Interpolation

The basic idea of interpolation is to get the fuzzy conclusion if two rules and the observation are given (see Fig. 2.2). An important notion in interpolative reasoning is the "less than" relation between two continuous, valid and normal fuzzy sets. Fuzzy set A_1 is said to be less than A_2, denoted by $A_1 \prec A_2$, if $\forall\alpha \in [0,1]$, the following conditions hold:

$$\inf\{A_{1\alpha}\} < \inf\{A_{2\alpha}\},\ \sup\{A_{1\alpha}\} < \sup\{A_{2\alpha}\}, \tag{2.35}$$

where $A_{1\alpha}$ and $A_{2\alpha}$ are respectively the α-cut of A_1 and that of A_2, $\inf\{A_{i\alpha}\}$ is the infimum of $A_{i\alpha}$, and $\sup\{A_{i\alpha}\}$ is the supremum of $A_{i\alpha}$, $i = 1,2$.

For simplicity, suppose that two fuzzy rules are given:

$$If\ X\ is\ A_1\ then\ Y\ is\ B_1,$$
$$If\ X\ is\ A_2\ then\ Y\ is\ B_2,$$

Also, suppose that these two rules are adjacent, i.e., there does not exist a rule such that the antecedent value A of that rule is between the region of A_1 and A_2. To entail the interpolation in the region between the antecedent values of these two rules, i.e., to determine a new conclusion B^* when an observation A^* located between fuzzy sets A_1 and A_2 is given, rules in a given rule base are arranged with respect to a partial ordering among the valid and normal fuzzy sets of the antecedents' variables. For the above two rules, this means that

$$A_1 \prec A^* \prec A_2. \tag{2.36}$$

To determine the fuzzy result B^*, the KH interpolation uses the interpolation equation

$$\frac{d(A^*, A_1)}{d(A^*, A_2)} = \frac{d(B^*, B_1)}{d(B^*, B_2)}, \tag{2.37}$$

where $d(.,.)$ is typically the Euclidean distance between two fuzzy sets (though other distance metrics may be used as alternatives). This is illustrated in Fig. 2.2, where the

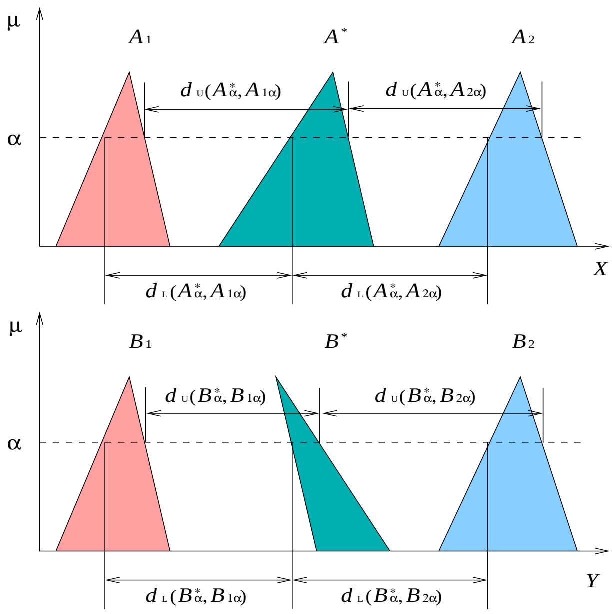

Figure 2.2: Fuzzy interpolative reasoning with an invalid conclusion on a sparse fuzzy rule base

lower and upper distances between α-cuts $A_{1\alpha}$ and $A_{2\alpha}$ are defined as follows:

$$d_L(A_{1\alpha}, A_{2\alpha}) = d(\inf\{A_{1\alpha}\}, \inf\{A_{2\alpha}\}), \tag{2.38}$$

$$d_U(A_{1\alpha}, A_{2\alpha}) = d(\sup\{A_{1\alpha}\}, \sup\{A_{2\alpha}\}). \tag{2.39}$$

From (2.38) and (2.39), (2.37) can be rewritten as:

$$\min\{B^*_\alpha\} = \frac{\frac{\inf\{B_{1\alpha}\}}{d_L(A^*_\alpha,A_{1\alpha})} + \frac{\inf\{B_{2\alpha}\}}{d_L(A^*_\alpha,A_{2\alpha})}}{\frac{1}{d_L(A^*_\alpha,A_{1\alpha})} + \frac{1}{d_L(A^*_\alpha,A_{2\alpha})}}, \tag{2.40}$$

$$\max\{B^*_\alpha\} = \frac{\frac{\sup\{B_{1\alpha}\}}{d_U(A^*_\alpha,A_{1\alpha})} + \frac{\sup\{B_{2\alpha}\}}{d_U(A^*_\alpha,A_{2\alpha})}}{\frac{1}{d_U(A^*_\alpha,A_{1\alpha})} + \frac{1}{d_U(A^*_\alpha,A_{2\alpha})}}. \tag{2.41}$$

From this, $B^*_\alpha = (\min\{B^*_\alpha\}, \max\{B^*_\alpha\})$ results. And the conclusion fuzzy set B^* can be constructed by the representation principle of fuzzy sets:

$$B^* = \bigcup_{\alpha\in[0,1]} \alpha B^*_\alpha. \tag{2.42}$$

Despite this method's capability of handling the tomato problem, it does not guarantee validity (although they may be normal, as B^* shown in Fig. 2.2).

2.4.2 The HCL Interpolation

The HCL interpolation method [HCL98] is an interpolative reasoning method based on the KH method. The difference is that it not only interpolates the bottoms of the fuzzy set, but also interpolates the highest point of fuzzy set. It can guarantee that "If fuzzy rules $A_1 \Rightarrow B_1$, $A_2 \Rightarrow B_2$ and the observation A^* are defined by triangular membership functions, the interpolated conclusion B^* will also be triangular-type". However, this method is specially designed for triangular cases, and thus the piecewise linearity property (see chapter 6 for more detailed discussion) is not preserved in general fuzzy sets (such as the trapezoidal).

The HCL interpolation method calculates the bottom of B^* in the same way as the KH method does, but calculates the top point in a different way. Fig. 2.3 shows the typical fuzzy interpolation problem, where k_1, t_1, k, t, k_2, t_2, h_1, m_1, h, m, h_2, and m_2 represent the slopes of corresponding fuzzy sets. The process to determine the top point of B^* is described as follows:

1. Deciding the slopes h and m of the triangular type membership function B^*. Let

$$k = k_1 x + k_2 y, \tag{2.43}$$

$$t = t_1 x + t_2 y, \tag{2.44}$$

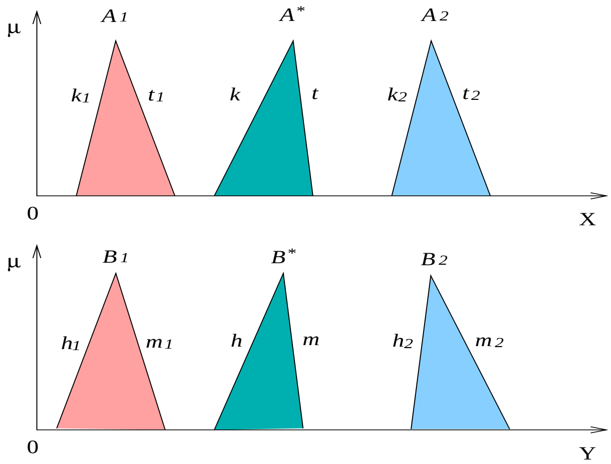

Figure 2.3: HCL fuzzy interpolation

where x and y are real numbers. If

$$\frac{k_1}{t_1} \neq \frac{k_2}{t_2},$$

then unique x and y are computed by solving (2.43) and (2.44) simultaneously. Let

$$h = |h_1 x + h_2 y| c, \tag{2.45}$$

$$m = -|m_1 x + m_2 y| c, \tag{2.46}$$

where c is a constant. Otherwise, let

$$h = kc, \tag{2.47}$$

$$m = tc, \tag{2.48}$$

where c is a constant.

2. Deciding the position of the top point b_1^* by solving the following equation,

$$\frac{1}{CP(B^*) - inf(B^*)} : \frac{-1}{sup(B^*) - CP(B^*)} = h : m, \tag{2.49}$$

where $CP(A)$ is the centre point of the specified fuzzy set A. It is defined as follows:

Definition 7 *The centre point of a given fuzzy set* $A \in F(X)$ *is:* $CP(A) = (\underline{A_\alpha} + \overline{A_\alpha})/2$*, where* $\alpha = height(A)$*.* A_α *denotes the* α*-cut of* A*.*

The centre point of a triangular fuzzy set is just its top point (of membership value of 1).

Equation (2.49) can be reformulated as

$$CP(B^*) = \frac{m \cdot sup(B^*) - h \cdot inf(B^*)}{m - h}. \tag{2.50}$$

2.4.3 The General Interpolation

As a member of the intermediate rule based interpolation family, the general interpolation [BGK96, BKG04] is capable of handling arbitrary membership functions, which is the main advantage of this approach. The general interpolation claims two groups of developed algorithms: one is based on the interpolation of fuzzy relations and the other is based on the interpolation of semantic relations. This subsection discusses the original and most typical method of this family, which consists of two key techniques: the *solid cutting* and the *revision principle*.

Solid cutting [BGK95, BK96a, BGK96, BG96, BK96b] is used to obtain the intermediate fuzzy set A' if the observation A^* and two fuzzy rules, $A_1 \Rightarrow B_1$ and $A_2 \Rightarrow B_2$ are given. A ratio of λ $(0 \leq \lambda \leq 1)$ is calculated to represent the important impact of A_2 upon the construction of intermediate rule antecedent A' with respect to A_1. The solid cutting method uses the centre point of the fuzzy set to represent its overall location. The λ thus can be computed as:

$$\lambda = \frac{d(A_1, A^*)}{d(A_1, A_2)} \tag{2.51}$$

$$= \frac{d(CP(A_1), CP(A^*))}{d(CP(A_1), CP(A_2))}, \tag{2.52}$$

where $d(.,.)$ stands for the distance between centre points of two fuzzy sets. In the extreme cases: if $\lambda = 0$, A_2 plays no part in constructing A', while if $\lambda = 1$, A_2 plays a full role in determining A'.

Fig. 2.4 shows how to calculate A' if A_1, A_2 and λ are given. Dimension S is orthogonal to plane $\mu \times X$. Let $g_k(s, t_k)$, $s \in S$, $t_k = CP(A_k)$, and $k = \{1, 2\}$, be the function that is obtained by rotating the membership function A_k: $\mu_{A_k}(x)$ by 90° around the axis t_k: $g_k(x - t_k, t_k) = \mu_{A_k}(x)$. Let a solid be constructed by fitting a surface on generatrices $g_k(s, t_k)$. Let $g'(s,t)$ be the cross-section of this imagined solid at position $t = CP(A')$, where $CP(A') = \Gamma(CP(A_1), CP(A_2), \lambda)$ and $\Gamma()$ stands for the *linear interpolation of two points*:

Definition 8 *The linear interpolation of two points x_1 and x_2 is*

$$x' = \Gamma(x_1, x_2, \lambda) = (1-\lambda)x_1 + \lambda x_2, \;\; \lambda \in [0,1].$$

Turning back $g'(s,t)$ into its original position, the interpolated fuzzy set A^i: $\mu_{A'}(x) = g'(x - cp(A'), cp(A'))$ is obtained.

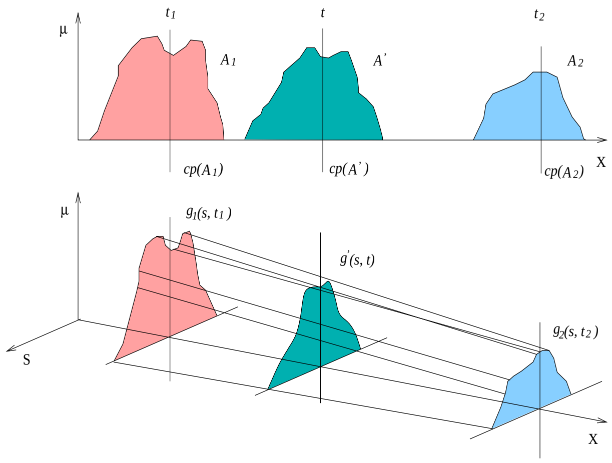

Figure 2.4: Interpolating fuzzy sets by solid cutting

For fuzzy interpolations only concerning triangular fuzzy sets, the solid cutting method works in the same way as the linear interpolation by using λ:

$$a'_0 = (1-\lambda)a_{10} + \lambda a_{20}, \tag{2.53}$$

$$a'_1 = (1-\lambda)a_{11} + \lambda a_{21}, \tag{2.54}$$

$$a'_2 = (1-\lambda)a_{12} + \lambda a_{22}, \tag{2.55}$$

which are collectively abbreviated to

$$A' = (1-\lambda)A_1 + \lambda A_2. \tag{2.56}$$

Similarly, the consequent fuzzy set B' can be obtained by

$$b'_0 = (1-\lambda)b_{10} + \lambda b_{20}, \tag{2.57}$$

$$b'_1 = (1-\lambda)b_{11} + \lambda b_{21}, \tag{2.58}$$

$$b'_2 = (1-\lambda)b_{12} + \lambda b_{22}, \tag{2.59}$$

with abbreviated notation:

$$B' = (1-\lambda)B_1 + \lambda B_2. \tag{2.60}$$

In so doing, the newly derived rule $A' \Rightarrow B'$ involves the use of only normal and valid fuzzy sets. The fuzzy set A' has the same centre of point as A^*. The revision principle based technique [SDM88, SDM93, MDS90, DSM89, DSM92] is used to infer the fuzzy conclusion by the new rule and the observation:

Definition 9 *The revision function $y = \Lambda(x, \mathbf{p1}, \mathbf{p2})$, where $x \in [\underline{x}, \overline{x}]$, $y \in [\underline{y}, \overline{y}]$, $\mathbf{p_1} = [p_{1,1} p_{1,2} \ldots p_{1,m}] \in \mathbb{R}^M$, where $p_{1,1} = a$ and $p_{1,M} = b$, and $\mathbf{p_2} = [p_{2,1} p_{2,2} \ldots p_{2,m}] \in \mathbb{R}^M$, where $p_{2,1} = c$ and $p_{2,M} = d$, subject to $p_{i,m} \leq p_{i,m+1}$, $i = 1, 2$.*

The revision function is a piecewise linear function where the linear pieces are defined by point-pairs $(p_{1,m}, p_{2,m})$. Fig. 2.5 shows a revision function with $M = 4$.

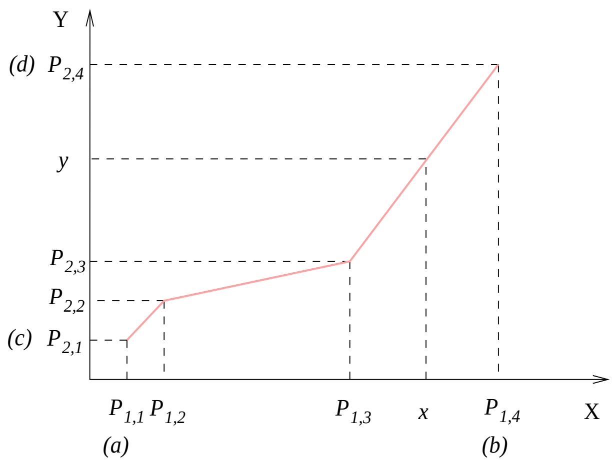

Figure 2.5: A revision function

In the triangular cases, the top point of the resulting fuzzy set B^* keeps the same position as that of B'. That is, $b^*_1 = b'_1$. The left and right points are determined by the revision principle:

$$b_0^* = \Lambda(a_0^*, \mathbf{p}_1, \mathbf{p}_2), \tag{2.61}$$

$$b_2^* = \Lambda(a_2^*, \mathbf{p}_1, \mathbf{p}_2), \tag{2.62}$$

where

$$\mathbf{p}_1 = [\underline{x}\ \underline{A'}\ CP(A')\ \overline{A'}\ \overline{x}],$$
$$\mathbf{p}_2 = [\underline{y}\ \underline{B'}\ CP(B')\ \overline{B'}\ \overline{y}].$$

2.4.4 The QMY Interpolation

As with the general fuzzy interpolation, Equations (2.56) and (2.60) are used to construct an intermediate rule $A' \Rightarrow B'$, where A' has the same centre point as A^*. To determine the left and right points of B^*, The QMY method [QMY96] suggests fuzzy reasoning in the following way:

1. Define a certain kind of similarity between two fuzzy sets.
2. Compare A^* and A' to get their similarity.
3. From B' reconstruct B^* according to the similarity transferred from the antecedent part.

This method is referred to as the *similarity transfer* (ST) reasoning method. The *similarities* between two fuzzy sets are defined as follows.

Definition 10 *Given two normal and valid fuzzy sets A and A' on the universe of discourse X, the lower similarity and the upper similarity between A and A' are respectively defined as follows.*

$$S_{L(A,A')}(\alpha) = \frac{d(inf(A_\alpha), CP(A))}{d(inf(A'_\alpha), CP(A))} \tag{2.63}$$

$$S_{U(A,A')}(\alpha) = \frac{d(sup(A_\alpha), CP(A))}{d(sup(A'_\alpha), CP(A))} \tag{2.64}$$

where $\alpha \in [0,1]$.

Then the consequence B^* is derived from the following equations:

$$CP(B^*) = CP(B'), \tag{2.65}$$

$$S_{L(A^*,A')}(\alpha) = S_{L(B^*,B')}(\alpha), \tag{2.66}$$

$$S_{U(A^*,A')}(\alpha) = S_{U(B^*,B')}(\alpha). \tag{2.67}$$

Combining (2.63)– (2.67) gives

$$inf(B^*_\alpha) = S_{L(A^*,A')}(\alpha)d(inf(B'_\alpha), CP(B')) + CP(B'), \tag{2.68}$$

$$sup(B^*_\alpha) = S_{U(A^*,A')}(\alpha)d(sup(B'_\alpha), CP(B')) + CP(B'). \tag{2.69}$$

Thus the consequence B^* can be calculated with the representation principle of fuzzy sets.

2.5 Hierarchical Fuzzy Reasoning

An alternative way of dealing with the "curse of dimensionality" is to use hierarchical fuzzy systems [RZK91, RZ93]. Such a system consists of a number of hierarchically connected low-dimensional fuzzy systems. Fig. 2.6 shows a typical example of hierarchical fuzzy systems. This n input hierarchical fuzzy system comprises $n-1$ low-dimensional fuzzy systems, with each low-dimensional fuzzy system having two inputs. If m fuzzy sets are defined for each variable, the total number of rules is $(n-1)m^2$ which is a linear function of the number of input variables n.

Earlier research work focuses on the proof of the availability of this approach. [Wan98, HB99, JKTV97, ZK04] show that any continuous function can be approximated by hierarchical fuzzy systems to achieve the universal approximation property. This enables the potential to build compact and efficient fuzzy models without the restraint of the curse of dimensionality. As a worked example, [SGB93] makes use of the hierarchical fuzzy system to control the unmanned helicopter. The work of [KHM00, KH93b] attempts the combination of hierarchical and sparse rule bases.

However, the main problem of the hierarchical reasoning is that it is often difficult to determine the low-dimensional fuzzy systems. Hierarchical fuzzy systems are

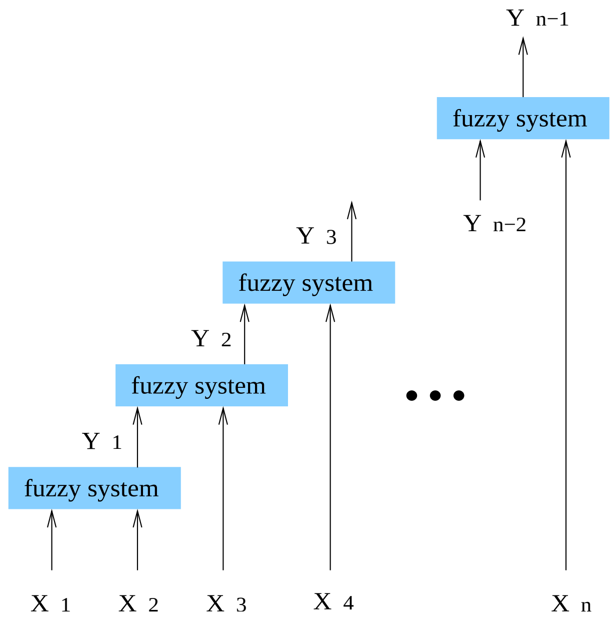

Figure 2.6: An example of an n input hierarchical fuzzy system

also criticized for transferring the complexity from the antecedent parts to the consequent parts. Nevertheless, the work of [Wan98] argued that the new structure no doubt does a better job in terms of distributing the burden somewhat "uniformly" over the antecedent and consequent parts. Further improvements to this work and the hybrid version of this system are desirable.

2.6 Summary

This chapter reviews the existing fuzzy simplification techniques which are vital to machine learning, pattern recognition and signal processing. In addition to overcoming the curse of dimensionality, simplification techniques are capable of enhancing the readability and transparency of reduced rule bases.

The outlined techniques consist of five categories: feature-based reduction, merging and removal-based reduction, orthogonal transformation based methods, interpolative reasoning methods and hierarchical fuzzy reasoning. Feature based reduction reduces the number of variables before the data is fed into machine learning tools. Merging and removal-based reduction simplify the rule bases by merging the similar rules or fuzzy sets, and removing the inconsistent or inactive rules. Orthogonal transformation based methods make use of matrix computation to optimise fuzzy rule bases. Interpolative reasoning methods not only simplify the rule base by eliminating the rules which can be approximated by their neighbors, but also provide a wise inference solution for sparse rule bases. Hierarchical fuzzy systems modify the structure of the conventional rule models and hence avoid the curse of dimensionality.

Three concerns are considered to choose a proper simplification technique for a question at hand. The first is about when the simplifications are to be applied: Feature-based reduction is a technique used before the rule induction; most orthogonal transformation based methods and hierarchical fuzzy systems are techniques used within the rule induction; and merging and removal-based reduction, some orthogonal transformation based methods and interpolative reasoning methods are techniques used after rule induction. Based on when a technique is applied, candidates can be identified for a given simplification task. For example, when a rule base is given and its associated fuzzy membership functions for each attribute are fixed, the candidates could only be chosen from the simplification techniques after the rule induction process.

The second concern to choose a simplification method is the preservation of the semantic meaning, as this is the major advantage of fuzzy modelling. Unfortunately, some simplification methods, such as the similarity merging and most of the transformation based methods, destroy the predefined fuzzy linguistic terms and hence result in loss of comprehensibility. In contrast, feature selection, interpolative reasoning methods and hierarchical fuzzy systems are good choices.

Finally, the concern is made to avoid generating sparse rule bases. When given observations have no overlap with the antecedent rule values, classical fuzzy inference methods have no rule to fire, but interpolative reasoning methods can still obtain certain conclusions. Thereby, this concern can be removed if interpolative reasoning based fuzzy inference is adopted.

Among all the existing approaches, the interpolative reasoning methods have been paid extra attention as they are closest to this Ph.D. project. Some typical interpolation methods which will be used in comparison (chapter 8) have been described in detail.

Although no significant difference can been seen through these discussions so far, these methods do make a lot of difference when they are used in a scaled-up application in chapter 8.

Chapter 3

RDFR Based Simplification Method

Rule model simplification techniques are desirable to alleviate the *curse of dimensionality* [Gui01, KJS02] so that models' effectiveness and transparency can be enhanced. This chapter proposes a novel simplification method by means of *retrieving data from rules*. It first retrieves a collection of new data from an original rule base. Then the new data is used for re-training to build a more compact rule model. This method has four advantages: 1) It can simplify rule bases without using the original training data, but is capable of dealing with combinations of rules and data. 2) It can integrate with any rule induction or reduction schemes. 3) It implements the similarity merging and inconsistency removal approaches. 4) It can make use of rule weights. Illustrative examples have been given to demonstrate the potential of this work.

The current rule simplification techniques are classified into three categories in terms of execution stages: the techniques executed before rule induction (RI) procedure, such as feature selection [DL97, JS04b]; the techniques integrated in the RI part, such as the orthogonal transformation based methods [CCG91, WM92a, YW99, YBY99]; and the techniques after the RI part, such as similarity merging [KB95, CCT96, SBKL98], inconsistency removal [XL02] and interpolative reasoning [KH93a, YK98]. The simplification methods in the first two categories make use of the original training data during their processes, while the methods in the third category are independent of the original training data (or they only need a small amount of data for test purposes). This difference highlights the advantages of the latter since the training data are not always available. In addition, most current simplification methods do not consider cases where both training data and rules, which are not necessarily obtained from any data but may be acquired directly from domain experts, are available. Another common disadvantage of current simplification methods is that they fail to make appropriate use of the rule weights (if applicable). This ignorance may destroy certain information of the underlying rule model. To overcome these two existing problems, a novel simplification method is proposed in this chapter.

The rest of the chapter is organised as follows. Section 3.1 gives a brief overview of the knowledge representation in IF-THEN production rules. Section 3.2 proposes the simplification method based on retrieving data from rules (RDFR). Section 3.3 gives realistic applications to illustrate the success of this method. Finally, Section 3.4 concludes the chapter and points out important further work.

3.1 Knowledge Representation

By means of human-like reasoning, production rule modelling becomes more and more popular in a variety of applications. The most outstanding advantage of this modelling is that it makes problem-solving systems understandable, unlike black-box techniques such as artificial neural networks. In particular, as an important part of rule modelling, fuzzy rule modelling is capable of handling perceptual uncertainties and imprecise information.

A typical fuzzy rule model consists of a set of IF-THEN rules, each of which takes certain crisp or fuzzy terms for input variables and output classes. Depending upon whether crisp or fuzzy terms are involved, the model is called a *crisp rule model* or *fuzzy rule model*. Since a multiple output rule can always be represented by several single output rules, without losing generality, only rules which have multiple input variables $X = (x_1, x_2, \ldots, x_n)$ and a single output class y are considered. Each input variable x_j, $j = 1, \ldots, n$, has $m(j)$ linguistic terms denoted as $A_{j1}, A_{j2}, \ldots, A_{j,m(j)}$. The whole linguistic terms of each input x_j can be defined by a vector $V_j = (A_{j1}, A_{j2}, \ldots, A_{j,m(j)})$. Similarly, the whole linguistic terms of the output is $V_{n+1} = (B_1, B_2, \ldots, B_{m(n+1)})$. Then a universal rule in a knowledge base has the following form:

$$if\ X_{p(1)} = SUB_1(V_{p(1)})\ and \cdots and\ X_{p(s)} = SUB_s(V_{p(s)})\ then\ y = B, \tag{3.1}$$

where $p(\cdot)$ is a mapping function from $\{1,2,...,s\}$, $(s \leq n)$, to $\{1,2,...,n\}$ satisfying $\forall x \neq y,\ p(x) \neq p(y)$, $SUB_k(k = 1,...,s)$ represents the subsethood operations and $B \in V_{n+1}$.

Consider the Saturday Morning Problem [YS95] as an example, with the attributes and their values shown in Table 3.1. A possible rule for this problem may be represented as

> IF Temperature is Hot AND Outlook is $\{$Sunny or Cloudy$\}$ THEN plan is Swimming.

Table 3.1: Saturday Morning Problem

Attribute name	Values
Outlook	Sunny, Cloudy, Rain
Temperature	Hot, Mild, Cool
Humidity	Humid, Normal
Wind	Windy, Normal
Plan	Volleyball, Swimming, Weight_lifting

If the first datum in Table.3.8 is given as an observation, the fuzzy inference is carried out as follows. First, as the fuzzy linguistic term "Sunny" takes on a fuzzy membership value of 0.9 and the "Cloudy" takes on 0.1, the logic union operator "or" calculates the maximal value (or other S-norm operators) of these two, that is, 0.9 as the firing strength (or confidence) of *Outlook is* $\{$*Sunny or Cloudy*$\}$. Then the logic intersection operator "and" calculates the minimal value (or other T-norm operators) of *Temperature is Hot* (1.0) and *Outlook is* $\{$*Sunny or Cloudy*$\}$ (0.9), resulting in a confidence of 0.9 to choose swimming as plan. Generally speaking, more than one rule may be fired for a given observation. All these rule results are aggregated to generate the final output.

3.2 The RDFR Based Simplification Method

3.2.1 Framework

The idea of the proposed method is inspired by the interchangeable usage of rules and data. In particular, the training data may be treated as specific rules. For instance in orthogonal transformation based methods [CCG91, WM92a, YW99], each training datum can be regarded as an individual "rule" (perhaps with fuzzification of attribute values) and only the "important rules" are retained to construct models. The reverse treatment that regards rules as data is attempted here. That is, the rules within an original rule base are used as training data to achieve a more compact rule model. However, some rules may not involve certain input variables, thus it is impossible for them to get re-trained directly. In order to solve this problem, a retrieving procedure is performed on each rule to assign vacant attributes (in that rule) with proper values, so that the retrieved data are ready for re-training.

The high level design (Fig. 3.1) of this method shows that the traditional rule induction and reduction procedures usually take place in the left dashed box. Rule induction algorithms (RIA) are generalization schemes which are used to learn from an original dataset (ODS) to derive an original rule set (ORS). Additionally, dimensionality reduction (DR) is applied before the training so that irrelevant or noisy input variables can be filtered out. Rule reductions (RR) such as similarity based rule merging [SBKL98] are applied to the original rule set to obtain a new one that is more compact. The main idea of the present simplification method is that it introduces a procedure of retrieving data from rules (RDFR) and a re-training procedure which is shown in the right dashed box. The RDFR based method builds a flexible and modular framework since any rule induction or reduction methods can be used in the right dashed box.

3.2.2 Retrieving Data From Rules (RDFR)

To formalise the description of the retrieving procedure, the following concepts and notations are introduced:

Definition 11 *The rule expressed in (3.1) is a* structure-complete rule *if all input vari-*

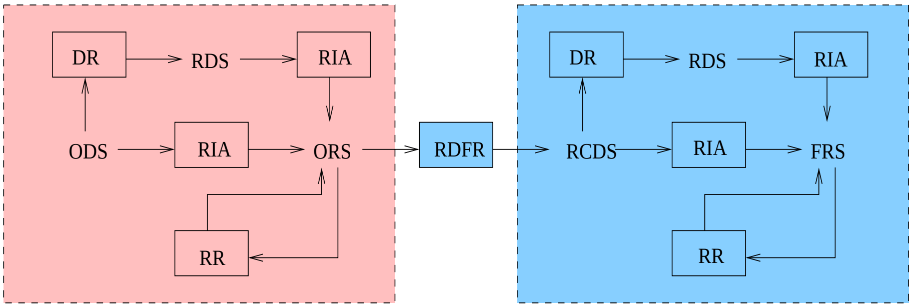

Figure 3.1: Simplification via Retrieving Data from Rules

ables are involved, i.e., $s = n$*. Other rules are termed* non-structure-complete rules.

Structure-complete rules are not preferred in modelling as they cannot represent as many data as non-structure-complete rules do.

Definition 12 *The rule of form (3.1) is a* multi-term rule *if at least one input* $X_{p(i)}$*,* $i = 1, \ldots, s$ *takes more than one linguistic or fuzzy term. i.e.,* $\exists k \in \{i = 1, \ldots, s\}$ *such that* $SUB_k(V_{p(k)})$ *involves more than one term.*

Multi-term rules represent logical union between alternative terms for certain variables. In contrast to multi-term rules, *single-term rules* contain variables which merely take one linguistic or fuzzy term. Obviously, a multi-term rule can be divided into many single-term rules without losing information. All the rules used later only concern single-term rules.

Definition 13 *A rule is a* complete-single-term rule *if it is a* single-term *rule as well as a* structure-complete *rule.*

In terms of the coverage of domain space, applying non-structure-complete rules is better than using structure-complete ones alone. This is because the former are more general than the latter. Let non-structure-complete rules be the "real rules" and the complete-single-term rules be “data”, the procedure of retrieving “data” from rules is implemented as retrieving complete-single-term rules from non-structure-complete rules. This implementation can be described as follows.

1: create a new datum in which each input variable is assigned null value and the class variable is assigned the linguistic class term of the given rule
2: **for** each input variable **do**
3:     **if** the variable is involved in the given rule **then**
4:         assign the linguistic fuzzy term of this variable in the given rule as the value of the new datum for this variable
5:     **else**
6:         assign an *appropriate* linguistic fuzzy term as the value of the new datum for this variable
7:     **end if**
8: **end for**

If the process retrieves all possible data from the given rule, this retrieval is called *exhaustive retrieval*. Otherwise, it is called *non-exhaustive retrieval*. Note that in the exhaustive case, there is no need to assign an *appropriate* linguistic fuzzy term to the newly constructed datum each time (as stated in line 6), as all the possible combinations of values of the vacant variables will be obtained anyway. However, for the non-exhaustive retrieval, such an assignment has to be considered. The next subsection gives examples to show how different assignments work.

3.2.3 Illustrative Examples

All the examples given in this subsection involve two inputs x_1 and x_2 and an output y. Assuming that each of them takes three linguistic (or fuzzy) terms which are denoted as A_1, A_2, A_3, B_1, B_2, B_3, and C_1, C_2, C_3 respectively.

3.2.3.1 Exhaustive Retrieval

Example 1. This example shows how to retrieve data from a rule model in a simple case. If a non-structure-complete rule is given as

$$if\ x_1 = A_1\ then\ y = C_1.$$

This rule is applied by RDFR to generate three data, which are shown in Table 3.2. As every datum has one fuzzy linguistic term associated with either x_1 or x_2, it can also be called a complete-single-term rule. Therefore, RDFR actually retrieves three complete-single-term rules from the given rule.

Table 3.2: Retrieve data from a non-structure-complete rule

No	x_1	x_2	y
1	A_1	B_1	C_1
2	A_1	B_2	C_1
3	A_1	B_3	C_1

Example 2. If two rules with different outputs are given:

$$if\ x_1 = A_1\ then\ y = C_1,$$
$$if\ x_2 = B_1\ then\ y = C_2.$$

A total of six data are retrieved and they are presented in Table 3.3. As can be seen, the first and the fourth data have the same inputs but derive different outputs, resulting in an inconsistency. In this case, if weights are being assigned to rules to reflect their importance, the data retrieved from the higher weighted rule ought to be retained and the others removed. This treatment makes use of rule weights and implements the method of inconsistency removal [XL02] at the data level rather than rule level. An experiment of such a treatment is given in section 3.3 to show its success in building compact and effective rule models. If however the weighting information is not available, voting can be used to choose the dominant datum from the retrieved inconsistent dataset.

Table 3.3: Retrieve data from two rules with different outputs

No	x_1	x_2	y
1	A_1	B_1	C_1
2	A_1	B_2	C_1
3	A_1	B_3	C_1
4	A_1	B_1	C_2
5	A_2	B_1	C_2
6	A_3	B_1	C_2

Example 3. If two rules with the same outputs are given:

$$if\ x_1 = A_1\ then\ y = C_1,$$
$$if\ x_2 = B_1\ then\ y = C_1.$$

A total of six data are retrieved as shown in Table 3.4. The first and the fourth data are identical. Clearly it is sufficient to keep one datum in this case. Such a treatment implements the similarity merging [KB95, CCT96, SBKL98] (identity merging in fact) at the data level rather than rule level. In the case of a scaled-up model, massive identical data may be retrieved. Such a process leads to less computation effort in re-training, thereby resulting in a more compact model.

Table 3.4: Retrieve data from two rules with the same output

No	x_1	x_2	y
1	A_1	B_1	C_1
2	A_1	B_2	C_1
3	A_1	B_3	C_1
4	A_1	B_1	C_1
5	A_2	B_1	C_1
6	A_3	B_1	C_1

3.2.3.2 Non-exhaustive Retrieval

The previous three examples illustrate retrieving all possible data from certain rules. Such exhaustive retrieval is hereafter referred to as ERDFR. Alternatively, *non-exhaustive retrieval* retrieves only partial of the whole possible data from given rules. The implementation of non-exhaustive retrieval is to assign an appropriate fuzzy linguistic term to each vacant variable (there is no need to assign to the variables which have already been associated with certain fuzzy linguistic terms). In particular, the implementations include assigning to each vacant variable:

- the most frequently used fuzzy linguistic term,
- the medium fuzzy linguistic term (if applicable),
- a randomly generated fuzzy linguistic term.

For later reference, the procedure of *randomly retrieving data from rules* is hereafter denoted as RRDFR.

Example 4. Given the same rules as in example 3,

$$if\ x_1 = A_1\ then\ y = C_1,$$
$$if\ x_2 = B_1\ then\ y = C_1.$$

Suppose that the first and second rules have different weights, say 1.0 and 0.5, obtained from certain training schemes (assuming the weights of rules are in the range of $[0,1]$). As the second rule is regarded to be not so confident as the first, one datum rather than three, may be retrieved from it, to reflect the lesser significance of this rule. One of the implementations is to choose the most frequently used term of x_1 (say, A_2) to generate the only data $x_1 = A_2 \wedge x_2 = B_1 \Rightarrow y = C_1$ from the second rule. The results are presented in Table 3.5. This retrieving strategy makes use of the rule weights, leading to a small amount of retrieved data which however may better represent the underlying model structure.

RDFR can be applied to simplify a model which has a combination of rules and data. In this case, the data retrieved from the given rule base are combined with the given data, to form a new training data set. Further processing will be carried out to the new training dataset to obtain the reduced models.

Table 3.5: Non-exhaustive retrieve data from two different weighted rules

No	x_1	x_2	y
1	A_1	B_1	C_1
2	A_1	B_2	C_1
3	A_1	B_3	C_1
4	A_2	B_1	C_1

3.3 Realistic Applications

The RDFR based method is applied in the same manner to either crisp or fuzzy rule models. Three examples concerned with both cases are given in this section to demonstrate the success of this work. In particular, the application for nursery prediction shows a crisp case, and the Saturday morning problem and credit applications show fuzzy cases.

3.3.1 Nursery Prediction

The *Nursery* database [HBM98] has eight nominal input variables and five output classes. It was derived from a hierarchical decision model originally developed to rank applications for nursery schools. Table 3.6 shows the variable names and values involved in the given database. A total of 12960 data in this database are divided evenly for training and test purposes. 55 rules are generated by the well-known decision tree algorithm (C4.5) [Qui86] with leaf objects set as 20 (a criterion to terminate C4.5 training) and the prediction accuracy on the test data is 92.48%. For further performance comparison, a simplified C4.5 tree (with minimal leaf objects set to 70) is obtained with 24 rules but having a lower prediction rate (89.77%).

The ERDFR and RRDFR are applied to retrieve all possible data (12960 in this case) and approximately 10% (1254) from the original rule set (produced by C4.5) respectively. The implementation of ERDFR and RRDFR in this example removes the inconsistency by following the first_in_first_kept principle. That is to say, during the retrieving process, if the previously retrieved dataset has a datum whose inputs are

Table 3.6: Nursery data base

Attribute name	Values
parents	usual, pretentious, great_pret
has_nurs	proper, less_proper, improper, critical, very_crit
form	complete, completed, incomplete, foster
children	1, 2, 3, more
housing	convenient, less_conv, critical
finance	convenient, inconv
social	non-prob, slightly_prob, problematic
health	recommended, priority, not_recom
class	not_recom, recommend, very_recom, priority, spec_prior

identical to the newly retrieved one, the newly retrieved will be dropped.

After ERDFR and RRDFR procedures, different classification schemes including decision trees (C4.5) [Qui86], PART [FW98, WF99] and Ridor [WF99] are applied to the retrieved data to generate more compact rule sets. PART is a classifier which generates a decision list rather than a collection of equally weighted rules. Ridor performs a tree-like expansion of exceptions with the leaf having only a default rule. The exceptions are a set of rules that predict a class different from the one that would be obtained if the default rule is fired. Using these classifiers the test results, in terms of rule number, average variable number (including the output class) and prediction rate on test data, are collectively presented in Table 3.7. Rule number stands for the number of rules for the rule sets obtained by classification schemes, average variable number stands for how many variables are averagely involved in a rule in the rule sets (used as an indicator to show how complex a rule set is), and the prediction rate on test data shows the prediction accuracy.

Table 3.7 shows that ERDFR + C4.5 is able to achieve the same performance as the original simplified C4.5 model. It is worth noting that the RDFR based method achieves this performance without using the original data. The ERDFR + PART scheme reduces the rule number to 39, while maintaining the same prediction accuracy

Table 3.7: Results comparison between C4.5 and RDFR simplification method

Schemes	Performance		
	Rule number	Average variable number	Prediction rate on test data
C4.5	55	4.81	92.48%
C4.5 (simplified)	24	3.83	89.77%
ERDFR + C4.5	24	3.83	89.77%
ERDFR + PART	39	3.26	92.48%
ERDFR + Ridor	25	3.28	91.27%
RRDFR + C4.5	21	3.67	89.38%
RRDFR + PART	34	3.04	92.61%
RRDFR + Ridor	26	3.32	91.16%

(92.48%) as produced by the original C4.5 model. The ERDFR + Ridor simplifies the rule model to 25 rules with a satisfactory prediction accuracy (91.27%), which is still higher than 89.77% produced by the original simplified C4.5 model. These experiments show that the ERDFR based simplification methods help simplify rule models while being capable of maintaining the same, or even improving, performance of the original rule set.

The experiments based on RRDFR produce more encouraging results. In particular, RRDFR + C4.5 achieves a model consisting of only 21 rules but with a lower prediction accuracy (89.38%), RRDFR + PART achieves the highest prediction rate (92.61%) with only 34 rules. RRDFR + Ridor generates a satisfactory result (91.16%) while significantly reducing the rule number from 55 to 26.

Considering the average number of the variables involved in a rule model, both ERDFR and RRDFR based methods obtain more compact models compared to the original C4.5 ones. As can be seen, the RRDFR based experiments outperform the ERDFR based ones. This is likely due to the fact that the randomly retrieved data generated from RRDFR may contain sufficient information to represent the underlying model structure. As it has much less data (1254 vs. 12960), it is more likely to result in effective and compact rule models.

3.3.2 Saturday Morning Problem

The Saturday morning problem [YS95] concerns the prediction of sports plan (volley ball, swimming and weight lifting) based on the status of outlook (sunny, cloudy and rain), temperature (hot, mild and cool), humidity (humid and normal) and wind (windy and not windy). Table 3.8 shows the given training set which includes 16 fuzzy data. A fuzzy decision tree generation method [YS95] has been applied to this dataset to generate six fuzzy rules which are presented below. The performance of the fuzzy decision tree over the training data is 81.25%.

```
Rule 1: IF Temperature is Hot AND Outlook is Sunny
THEN Swimming (S = 0.85)
Rule 2: IF Temperature is Hot AND Outlook is Cloudy
THEN Swimming (S = 0.72)
Rule 3: IF Temperature is Hot AND Outlook is Rain
THEN Weight_lifting (S = 0.73)
Rule 4: IF Temperature is Mild AND Wind is Windy
THEN Swimming (S = 0.81)
Rule 5: IF Temperature is Mild AND Wind is Not_windy
THEN Volleyball (S = 0.81)
Rule 6: IF Temperature is Cool THEN Weight_lifting (S = 0.88)
```

Note that Rule 3 can be simplified to Rule 3’:

```
Rule 3’: IF Outlook is Rain THEN Weight_lifting (S = 0.89)
```

Also, note that `S` is the classification truth level at the leaf.

Table 3.8: Saturday Morning Problem dataset

Case	Outlook			Temperature			Humidity		Wind		Plan		
	Sunny	Cloudy	Rain	Hot	Mild	Cool	Humid	Normal	Windy	Not_windy	Volleyball	Swimming	W_lifting
1	0.9	0.1	0.0	1.0	0.0	0.0	0.8	0.2	0.4	0.6	0.0	0.8	0.2
2	0.8	0.2	0.0	0.6	0.4	0.0	0.0	1.0	0.0	1.0	1.0	0.7	0.0
3	0.0	0.7	0.3	0.8	0.2	0.0	0.1	0.9	0.2	0.8	0.3	0.6	0.1
4	0.2	0.7	0.1	0.3	0.7	0.0	0.2	0.8	0.3	0.7	0.9	0.1	0.0
5	0.0	0.1	0.9	0.7	0.3	0.0	0.5	0.5	0.5	0.5	0.0	0.0	1.0
6	0.0	0.7	0.3	0.0	0.3	0.7	0.7	0.3	0.4	0.6	0.2	0.0	0.8
7	0.0	0.3	0.7	0.0	0.0	1.0	0.0	1.0	0.1	0.9	0.0	0.0	1.0
8	0.0	1.0	0.0	0.0	0.2	0.8	0.2	0.8	0.0	1.0	0.7	0.0	0.3
9	1.0	0.0	0.0	1.0	0.0	0.0	0.6	0.4	0.7	0.3	0.2	0.8	0.0
10	0.9	0.1	0.0	0.0	0.3	0.7	0.0	1.0	0.9	0.1	0.0	0.3	0.7
11	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.0	0.2	0.8	0.4	0.7	0.0
12	0.2	0.6	0.2	0.0	1.0	0.0	0.3	0.7	0.3	0.7	0.7	0.2	0.1
13	0.9	0.1	0.0	0.2	0.8	0.0	0.1	0.9	1.0	0.0	0.0	0.0	1.0
14	0.0	0.9	0.1	0.0	0.9	0.1	0.1	0.9	0.7	0.3	0.0	0.0	1.0
15	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.8	0.2	0.0	0.0	1.0
16	1.0	0.0	0.0	0.5	0.5	0.0	0.0	1.0	0.0	1.0	0.8	0.6	0.0

Retrieving is now applied to the these rules. As there are only 38 data retrieved by ERDFR, it is not necessary to apply RRDFR. The RRDFR procedure is thus omitted in this example. Within the retrieved 38 data by ERDFR, there are two pairs of inconsistent data as shown in Table 3.9. In particular, data 1 and 2 constitute an inconsistent pair, and data 3 and 4 form another. Data 1 and 3 are retrieved from rule 5 and data 2 and 4 are from rule $3'$. Since the weight (truth level) of rule 5 is 0.81 whilst that of rule $3'$ is 0.89, data 2 and 4 are of a higher confidence than data 1 and 3 respectively. Data 1 and 3 are hence removed from the retrieved 38 data. For comparison purposes, both of these two datasets are used to construct new fuzzy models and they are referred to as *data 38* and *data 36* hereafter. For each dataset, three classification schemes including C4.5, PART and JRip [WF99] are adopted. The final results are compared to the work of [YS95] in terms of rule number, average number of variables involved and prediction rate on the original training data. These results are presented in Table 3.10 and Table 3.11 (for data 38 and data 36 respectively). Note that the prediction rates on the *retrieved* data are also given in the tables.

Table 3.9: Two pairs of inconsistent data after ERDFR

No	Outlook	Temperature	Humidity	Wind	Plan
1	rain	mild	humid	not-windy	volleyball
2	rain	mild	humid	not-windy	weight_lifting
3	rain	mild	normal	not-windy	volleyball
4	rain	mild	normal	not-windy	weight_lifting

Table 3.10 shows that ERDFR + C4.5 produces the same prediction rate (81.25%) on the original data as the work of [YS95], despite the prediction rate on the retrieved data being much higher (94.74%). The ERDFR + PART achieves the same prediction accuracy. It however reduces the average number of variables involved per rule from 2.67 to 2.17, resulting in a more compact fuzzy rule base. ERDFR + JRip effectively reduces the rule number from six to three. Unfortunately, it brings down the prediction rate to 75%.

Table 3.10: Results comparison based on data 38

Schemes	Performance			
	Rule number	Average variable number	Prediction rate on retrieved data	Prediction rate on original data
Fuzzy Decision Trees	6	2.67	81.25%	81.25%
ERDFR + C4.5	6	2.67	94.74%	81.25%
ERDFR + PART	6	2.17	94.74%	81.25%
ERDFR + JRip	3	2.00	84.21%	75%

Table 3.11: Results comparison based on data 36

Schemes	Performance			
	Rule number	Average variable number	Prediction rate on retrieved data	Prediction rate on original data
Fuzzy Decision Trees	6	2.67	81.25%	81.25%
ERDFR + C4.5	8	2.88	100%	81.25%
ERDFR + PART	6	2.17	100%	93.75%
ERDFR + JRip	4	2.50	94.44%	81.25%

Table 3.11 shows a much improved performance. Although ERDFR + C4.5 in fact increases the rule number, ERDFR + PART reduces the average number of variables per rule from 2.67 to 2.17 while achieving a high prediction accuracy (93.75%). ERDFR + JRip reduces the number of rules from 6 to 4 while keeping the same prediction accuracy as the work of [YS95]. These two successful simplified fuzzy models are provided below. Note that PART and JRip generate ordered fuzzy rules, a firing threshold $\alpha = 0.7$ is imposed on both models to classify new data. That is, if any rule in the ordered list has a firing strength (for the given data) more than this threshold, the prediction will be determined by this rule.

```
ERDFR + PART on data 36:
Rule 1: IF Temperature is Cool THEN Weight_lifting
Rule 2: IF Temperature is Hot AND Outlook is Sunny THEN Swimming
Rule 3: IF Temperature is Mild AND Wind is Windy THEN Weight_lifting
```

```
Rule 4: IF Outlook is Rain THEN weight_lifting
Rule 5: IF Temperature is Hot THEN Swimming
Rule 6: Volleyball

ERDFR + JRip on data 36:
Rule 1: IF Temperature is Mild and Wind is not_windy THEN Volleyball
Rule 2: IF Temperature is Hot and Outlook is Cloudy THEN Swimming
Rule 3: IF Temperature is Hot and Outlook is Sunny THEN Swimming
Rule 4: Weight_lifting
```

The comparison between Table 3.10 and Table 3.11 indicates that inconsistency removal helps filter out noisy information, contributing to the construction of compact and effective models. This step is particularly useful when the performance of the original rule base is poor, as it is very likely there are inconsistent or conflicting information existing in that rule base.

From Tables 3.10 and 3.11, the performance of the rule bases tested on the retrieved data is not proportional to that on the original data for different classification schemes. However for one particular classification scheme, the higher the performance on retrieved data, the higher the performance is likely to be obtained on the original data. Table 3.12 and Table 3.13 show the results of different models with various training criteria for data 38 and data 36 respectively. For instance, if ERDFR + PART results in two models with a prediction rate of 94.74% and 89.47% on retrieved data 38 respectively, it is more likely that the first model also outperforms the second (81.25% vs. 68.75%) on the original data. This observation provides a useful guide to find an optimum rule base reduction without the use of original data.

3.3.3 Credit Applications

The credit applications data [HBM98], provided by a large bank, is a collection of individual applications for credit card facilities. Each application involves 9 discrete and 6 continuous attributes, with two decision classes (accept or reject). To make the comparison available to the results given in [Qui87], the 690 data (with 37 having one or more than one missing value) are randomly divided into a training set of 460 and a

Table 3.12: Results comparison based on data 38

Schemes	Performance			
	Rule number	Average variable number	Prediction rate on retrieved data	Prediction rate on original data
ERDFR + C4.5	6	2.67	94.74%	81.25%
	4	2.67	84.21%	75.00%
ERDFR + PART	6	2.17	94.74%	81.25%
	5	2.20	89.47%	68.75%

Table 3.13: Results comparison based on data 36

Schemes	Performance			
	Rule number	Average variable number	Prediction rate on retrieved data	Prediction rate on original data
ERDFR + C4.5	8	2.88	100%	81.25%
	4	2.50	83.83%	75.00%
ERDFR + PART	6	2.17	100%	93.75%
	5	2.20	88.89%	68.75%

test set of 230 (keeping each decision class the same proportion as that of the original dataset). As some discrete attributes have large collections of possible values (one of them has 14), this dataset results in broad, shallow decision trees. Also, since this data is both scanty and noisy, the generated decision trees are extremely complex and not very accurate on unseen cases.

The fuzzy decision tree algorithm [UOHT94] is applied to the training data. For simplicity, and not to give any bias towards any variable domains, each variable is evenly divided into n ($n > 0$) fuzzy partitions. The resulting rule number and prediction accuracy are shown in Table 3.14 (with respect to the number of evenly distributed fuzzy partitions and the number of leaf objects). As can be seen, the size of the rule set decreases and the accuracy increases while the number of leaf objects increases. This is because the increasing of leaf objects removes the less general rules which may cause model over-fitting, thereby resulting in more general fuzzy models. Further increasing the leaf objects to a large number, the number of rules (and the accuracy) tends to become independent of the size of the fuzzy partitions used. This is because all the numerical attributes are ruled out as important decision-making attributes, i.e., they are not involved in any of the rule sets. This indicates that the numerical attributes are less informative than the nominal ones in this application. Table 3.15 shows that in the extreme case (when the number of leaf objects is equal to or greater than 250), only two rules:

```
Rule 1: IF A9 is true, THEN +
Rule 2: if A9 is false, THEN -
```

are generated with an accuracy of 86.5%. The result is obviously better than the best result (which uses 11 rules with an average accuracy of 85.6%) produced in [Qui87].

Table 3.14: Fuzzy C4.5 results over Credit dataset

Number of fuzzy partitions	Objects = 2		Objects = 10		Objects = 20		Objects = 30	
	Rule No	Accuracy	Rule No	Accuracy	Rule No	Accuracy	Rule No	Accuracy
2	134	82.1%	72	85.6%	52	85.2%	45	85.2%
3	135	84.8%	74	85.2%	53	85.2%	45	85.2%
4	150	81.7%	84	86.0%	56	85.2%	47	85.2%
5	172	81.3%	82	85.2%	54	85.2%	47	85.2%
6	154	80.4%	82	84.4%	57	84.8%	48	85.2%
7	184	77.4%	89	83.5%	61	84.3%	49	85.2%
8	190	80.4%	95	84.3%	59	84.8%	50	85.2%
9	191	79.1%	102	83.4%	59	84.3%	50	85.2%
10	206	77.4%	110	83.5	62	84.8%	51	85.2%

Table 3.15: Fuzzy C4.5 results over Credit dataset

Objects	60	100	200	225	250	300
Rule number	41	28	15	3	2	2
Accuracy	85.2%	86.1%	86.1%	86.5%	86.5%	86.5%

For this dataset, such a compact rule set (consists of 2 rules) can effectively and efficiently predict the unknown data. Thus, there is no need for further simplification.

However, in many cases, such a satisfiable rule model may not be obtained (for instance, the work of [Qui87]). To test if the RDFR-based simplification can manage to find the model presented above, the case with 30 leaf objects and 3 fuzzy partitions is chosen as the original rule base. Since exhaustive retrieval implies the return of far too many data for this problem, a random retrieval is employed here to obtain 500 data. Then these 500 data are fed to the PART (with the minimum number of instances per rule is set to 2 as the training parameter) , resulting in a rule base with 16 rules and an accuracy of 85.7%. If the minimum number of instances per rule for PART is set to 50 rather than 2 (in order to retain more general fuzzy rules), the rule model with exactly the same two rules (as presented above) can be obtained.

Similarly, if the 500 randomly retrieved data are fed into the JRip algorithm (with the minimum instances per rule set to 2 as the training parameter), 7 rules are obtained with an accuracy of 85.7%. Again, if the minimum number of objects per rule is changed (from 2 to 30), the same two rules can be achieved.

In summary, the RDFR-based simplification method can use the size of the retrieved data to determine the complexity of the final rule base, and it is capable of finding good solutions in the presented examples.

3.4 Summary

This chapter proposes a novel rule model simplification method via Retrieving Data from Rules (RDFR). It first retrieves a collection of new data from an original rule base. Then the new data is used for re-training to build a more compact rule model. This method has four advantages: 1) It can simplify rule bases without using the original training data, but is capable of dealing with combinations of rules and data. 2) It can integrate with any rule induction or reduction schemes. 3) It implements the similarity merging and inconsistency removal approaches. 4) It can make use of rule weights. Illustrative examples including the nursery prediction, Saturday morning problem and credit applications are given to demonstrate the success of this work.

However, much more can be carried out to improve further the performance of this method. In particular, different retrieving methods with respect to the use of different weighted rules in a given rule set are worth further investigating. Also, this method only applies to *non-structure-complete* rules. The retrieving techniques that can deal with *structure-complete* rules require further research.

Chapter 4

Transformation Based Interpolation: Specific Examples

4.1 Motivation

As mentioned in section 2.4, fuzzy interpolative reasoning methods not only reduce the complexity of the fuzzy modelling, but also make inference in sparse rule bases possible. However, some of the existing methods may include complex computation. It becomes more difficult when they are extended to multiple variables interpolation. Others may only apply to simple fuzzy membership functions limited to triangular or trapezoidal. Almost all generate unique results while the work of [YK00, YWB00] obtains more than one result; the former lack the flexibility whilst the latter does not show how to decide the final result. This chapter proposes a novel interpolative reasoning method which avoids the problems mentioned above. It is a method in the category of intermediate rule based interpolations. Firstly an intermediate fuzzy rule is constructed by its two adjacent rules. Then it together with the observation are converted into the final results by proposed scale and move transformations, which ensure unique as well as normal and valid fuzzy (NVF) sets.

The rest of the chapter is organised as follows. Section 4.2, 4.3 and 4.4 describe the proposed scale and move transformations with single antecedent variable having triangular, trapezoidal and hexagonal fuzzy sets respectively. Section 4.5 gives the outline of the interpolation method based on the triangular, trapezoidal and hexagonal examples. Section 4.6 summarises the chapter.

4.2 Single Antecedent Variable with Triangular Fuzzy Sets

Triangular fuzzy membership functions are considered first to demonstrate the basic ideas of the present work, due to its simplicity and popularity. This is to be followed by more complex functions such as trapezoidal and hexagonal in the next subsections. For presentational simplicity, only rules involving one antecedent variable are dealt with here, with a generalised case to be given later.

To facilitate this discussion, the *representative value* of a triangular membership function is defined as the average of the x coordinates of its three key points: the left and right extreme points (whose membership values are 0) and the normal point (whose membership value is 1). Without losing generality, given a fuzzy set A, denoted as (a_0, a_1, a_2), as shown in Fig. 4.1, its representative value is

$$Rep(A) = \frac{a_0 + a_1 + a_2}{3}. \tag{4.1}$$

This representative value happens to be the x coordinate of the centre of gravity of

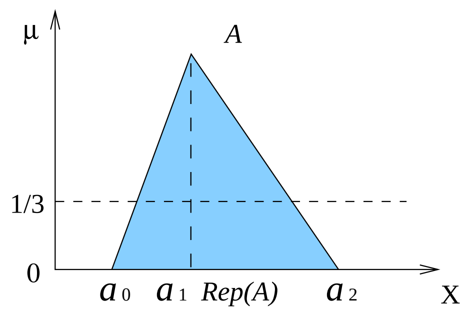

Figure 4.1: Representative value of a triangular fuzzy set

such a triangular fuzzy set [HS03].

Suppose that two adjacent fuzzy rules $A_1 \Rightarrow B_1$, $A_2 \Rightarrow B_2$ and the observation A^*, which is located between fuzzy sets A_1 and A_2, are given. The case of interpolative fuzzy reasoning concerning two variables X and Y can be described through the modus ponens interpretation (4.2), as illustrated in Fig. 4.2.

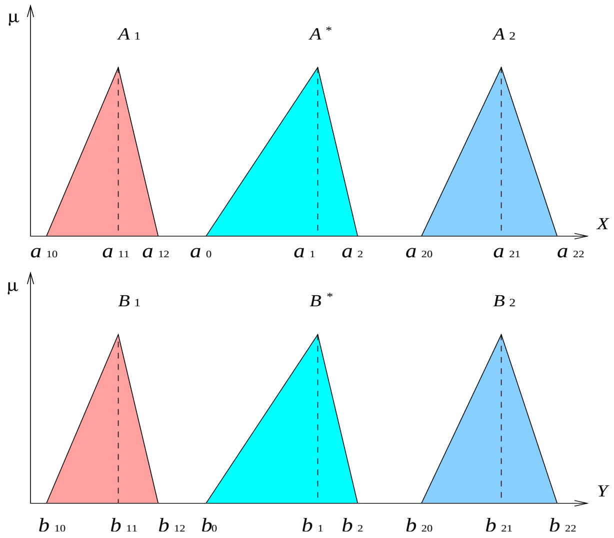

Figure 4.2: Interpolation with triangular membership functions

$$\begin{array}{l} \text{observation: } X \text{ is } A^* \\ \text{rules: if } X \text{ is } A_1 \text{, then } Y \text{ is } B_1 \\ \qquad\quad \text{if } X \text{ is } A_2 \text{, then } Y \text{ is } B_2 \\ \hline \text{conclusion: } Y \text{ is } B^* ? \end{array} \tag{4.2}$$

Here, $A_i = (a_{i0}, a_{i1}, a_{i2})$, $B_i = (b_{i0}, b_{i1}, b_{i2})$, $i = 1, 2$, and $A^* = (a_0, a_1, a_2)$, $B^* = (b_0, b_1, b_2)$.

To perform interpolation, the first step is to construct a new fuzzy set A' which has the same representative value as A^*. For this, the following is created first:

$$\begin{aligned} \lambda_{Rep} &= \frac{d(A_1, A^*)}{d(A_1, A_2)} \\ &= \frac{d(Rep(A_1), Rep(A^*))}{d(Rep(A_1), Rep(A_2))} \\ &= \frac{\frac{a_0+a_1+a_2}{3} - \frac{a_{10}+a_{11}+a_{12}}{3}}{\frac{a_{20}+a_{21}+a_{22}}{3} - \frac{a_{10}+a_{11}+a_{12}}{3}}, \end{aligned} \tag{4.3}$$

where $d(A_1, A_2) = d(Rep(A_1), Rep(A_2))$ represents the distance between two fuzzy sets A_1 and A_2.

From this, a'_0, a'_1 and a'_2 of A' are calculated as follows:

$$a'_0 = (1-\lambda_{Rep})a_{10} + \lambda_{Rep}a_{20}, \tag{4.4}$$

$$a'_1 = (1-\lambda_{Rep})a_{11} + \lambda_{Rep}a_{21}, \tag{4.5}$$

$$a'_2 = (1-\lambda_{Rep})a_{12} + \lambda_{Rep}a_{22}, \tag{4.6}$$

which are collectively abbreviated to

$$A' = (1-\lambda_{Rep})A_1 + \lambda_{Rep}A_2. \tag{4.7}$$

Now, A' has the same representative value as A^*.

proof 1

$$Rep(A') = \frac{a'_0 + a'_1 + a'_2}{3}.$$

With (4.4)–(4.6) and (4.3),

$$\begin{aligned} Rep(A') &= (1-\lambda_{Rep})\frac{a_{10}+a_{11}+a_{12}}{3} + \lambda_{Rep}\frac{a_{20}+a_{21}+a_{22}}{3} \\ &= (1-\lambda_{Rep})Rep(A_1) + \lambda_{Rep}Rep(A_2) \\ &= Rep(A^*). \end{aligned}$$

Importantly, in so doing, A' is generated to be a valid fuzzy set as the following holds given $a_{10} \le a_{11} \le a_{12}$, $a_{20} \le a_{21} \le a_{22}$ and $0 \le \lambda_{Rep} \le 1$:

$$\begin{aligned} a'_1 - a'_0 &= (1-\lambda_{Rep})(a_{11}-a_{10}) + \lambda_{Rep}(a_{21}-a_{20}) \ge 0, \\ a'_2 - a'_1 &= (1-\lambda_{Rep})(a_{12}-a_{11}) + \lambda_{Rep}(a_{22}-a_{21}) \ge 0. \end{aligned}$$

The second step of performing interpolation is carried out in a similar way to the first, such that the consequent fuzzy set B' can be obtained as follows:

$$b'_0 = (1-\lambda_{Rep})b_{10} + \lambda_{Rep}b_{20}, \tag{4.8}$$

$$b'_1 = (1-\lambda_{Rep})b_{11} + \lambda_{Rep}b_{21}, \tag{4.9}$$

$$b'_2 = (1-\lambda_{Rep})b_{12} + \lambda_{Rep}b_{22}, \tag{4.10}$$

with abbreviated notation:

$$B' = (1-\lambda_{Rep})B_1 + \lambda_{Rep}B_2. \tag{4.11}$$

As a result, the newly derived rule $A' \Rightarrow B'$ involves the use of only NVF sets.

As $A' \Rightarrow B'$ is derived from $A_1 \Rightarrow B_1$ and $A_2 \Rightarrow B_2$, it is feasible to perform fuzzy reasoning with this new rule without further reference to its originals. The interpolative reasoning problem is therefore changed from expression (4.2) to the new modus ponens interpretation:

$$\begin{array}{l} \text{observation: } X \text{ is } A^* \\ \text{rule: if } X \text{ is } A', \text{ then } Y \text{ is } B' \\ \hline \text{conclusion: } Y \text{ is } B^*? \end{array} \tag{4.12}$$

This interpretation retains the same results as (4.2) in dealing with the extreme cases: If $A^* = A_1$, then it follows from (4.3) that $\lambda_{Rep} = 0$, and according to (4.7) and (4.11), $A' = A_1$ and $B' = B_1$, so the conclusion $B^* = B_1$. Similarly, if $A^* = A_2$, then $B^* = B_2$.

Other than the extreme cases, *similarity* measures are used to support the application of this new modus ponens as done in [QMY96]. In particular, (4.12) can be interpreted as

$$\textit{The more similar } X \textit{ to } A', \textit{ the more similar } Y \textit{ to } B'. \tag{4.13}$$

Suppose that a certain degree of similarity between A' and A^* is established, it is intuitive to require that the consequent parts B' and B^* attain the same similarity degree. The question is now how to obtain an operator which can represent the similarity degree between fuzzy sets A' and A^*, and to allow transforming B' to B^* with the desired degree of similarity. In this respect, two transformations are proposed as follows.

Scale Transformation Given a *scale rate* s ($s \geq 0$), in order to transform the current support $(a_2 - a_0)$, of fuzzy set $A = (a_0, a_1, a_2)$, into a new support $(s*(a_2 - a_0))$ while keeping the same representative value and ratio of left-support $(a_1' - a_0')$ to right-support $(a_2' - a_1')$ of the transformed fuzzy set, $A' = (a'_0, a'_1, a'_2)$, as those of its original, that is, $Rep(A') = Rep(A)$ and $\frac{a'_1 - a'_0}{a'_2 - a'_1} = \frac{a_1 - a_0}{a_2 - a_1}$, the new a'_0, a'_1 and a'_2 must satisfy (as illustrated in Fig. 4.3):

$$a'_0 = \frac{a_0(1+2s)+a_1(1-s)+a_2(1-s)}{3}, \tag{4.14}$$

$$a'_1 = \frac{a_0(1-s)+a_1(1+2s)+a_2(1-s)}{3}, \tag{4.15}$$

$$a'_2 = \frac{a_0(1-s)+a_1(1-s)+a_2(1+2s)}{3}. \tag{4.16}$$

In fact, to satisfy the conditions imposed over the transformation, the linear equations

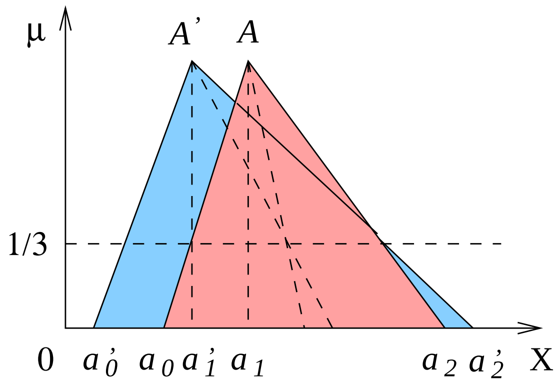

Figure 4.3: Triangular scale transformation

below must hold simultaneously:

$$\begin{cases} \frac{a'_0+a'_1+a'_2}{3} = \frac{a_0+a_1+a_2}{3} \\ \frac{a'_1-a'_0}{a'_2-a'_1} = \frac{a_1-a_0}{a_2-a_1} \\ a'_2 - a'_0 = s(a_2-a_0) \end{cases}$$

Solving these equations leads to the solutions as given in (4.14)–(4.16). Note that this scale transformation guarantees that the transformed fuzzy sets are valid as the following holds given $a_0 \leq a_1 \leq a_2$ and $s \geq 0$:

$$a'_1 - a'_0 = s(a_1 - a_0) \geq 0$$

$$a'_2 - a'_1 = s(a_2 - a_1) \geq 0$$

The above shows how to obtain the resultant fuzzy set A' when the original fuzzy set A and a scale rate s are given. Conversely, in the case where two fuzzy sets $A =$

(a_0, a_1, a_2) and $A' = (a'_0, a'_1, a'_2)$ which have the same representative value are given, the scale rate is calculated as follows:

$$s = \frac{a'_2 - a'_0}{a_2 - a_0} \geq 0. \tag{4.17}$$

This measure reflects the similarity degree between A and A': the closer is s to 1, the more similar is A to A'. It is therefore used to act as, or to contribute to, the desirable similarity degree in order to transform B' to B^*.

Move Transformation Given a moving distance l, in order to transform the current fuzzy support $(a_2 - a_0)$ from the starting location a_0 to a new starting position $a_0 + l$ while keeping the same representative value and length of support of the transformed fuzzy set as its original, i.e., $Rep(A') = Rep(A)$ and $a'_2 - a'_0 = a_2 - a_0$, the new a'_0, a'_1 and a'_2 must be (as shown in Fig. 4.4):

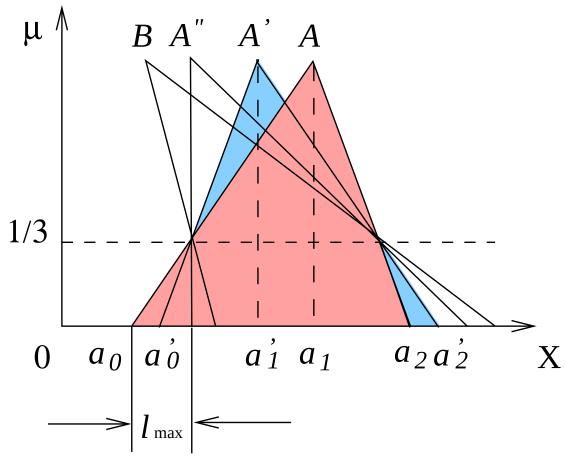

Figure 4.4: Triangular move transformation

$$a'_0 = a_0 + l, \tag{4.18}$$
$$a'_1 = a_1 - 2l, \tag{4.19}$$
$$a'_2 = a_2 + l. \tag{4.20}$$

These can be obtained by solving the equations which are imposed to the transformation:

$$\begin{cases} \frac{a'_0+a'_1+a'_2}{3} = \frac{a_0+a_1+a_2}{3} \\ a'_0 = a_0 + l \\ a'_2 - a'_0 = a_2 - a_0 \end{cases}$$

To ensure A' to be valid, the condition of $0 \leq l \leq l_{max} = (a_1 - a_0)/3$ must hold. If $l > l_{max}$, the transformation will generate invalid fuzzy sets. For instance, consider the extreme case in which A is transformed to A'', where the left slope of A'' becomes vertical (i.e. $a'_0 = a'_1$) as shown in Fig. 4.4. Here, $l = l_{max}$. Any further increase in l will lead to the resulting transformed fuzzy set being a non-NVF set. To avoid this, the *move ratio* $\mathbb{M}$ is introduced:

$$\mathbb{M} = \frac{l}{(a_1 - a_0)/3}. \tag{4.21}$$

The closer is $\mathbb{M}$ to 0, the less move (in terms of moving displacement l) is being made, and the closer is $\mathbb{M}$ to 1, the more move is being made. If move ratio $\mathbb{M} \in [0,1]$, then $l \leq l_{max}$ holds. This ensures that the transformed fuzzy set A' to be normal and valid if A is itself an NVF set.

Note that the move transformation has two possible moving directions, the above discusses the right-direction case (from the viewpoint of a_0) with $l \geq 0$, the left direction with $l \leq 0$ should hold by symmetry:

$$\mathbb{M} = \frac{l}{(a_2 - a_1)/3} \in [-1, 0]. \tag{4.22}$$

As with the description for scale transformation, the above describes how to calculate resultant fuzzy set A' given the original fuzzy set A and a moving distance l (or move ratio $\mathbb{M}$). Now, consider the case where two valid triangular sets $A = (a_0, a_1, a_2)$ and $A' = (a'_0, a'_1, a'_2)$ which have the same representative value and have the same support lengths are given, the move ratio $\mathbb{M}$ can be calculated as follows:

$$\mathbb{M} = \begin{cases} \frac{3(a'_0 - a_0)}{a_1 - a_0} & if\ a'_0 \geq a_0 \\ \frac{3(a'_0 - a_0)}{a_2 - a_1} & if\ a'_0 \leq a_0 \end{cases} \tag{4.23}$$

This reflects the similarity degree between A and A': the closer is $\mathbb{M}$ to 0, the more similar is A to A'. As A and A' both are valid, $\mathbb{M} \in [0,1]$ (when $a'_0 \geq a_0$) or $\mathbb{M} \in [-1,0]$ (when $a'_0 \leq a_0$) must hold.

Thus, in general, the third step of the interpolation process is to calculate the similarity degree in terms of scale rate and move ratio between A' and A^*, and then obtain the resultant fuzzy set B^* by transforming B' with the same scale rate and move ratio.

Through interpolation steps 1 - 3, given a normal and valid triangular fuzzy set as the observation, a new normal and valid fuzzy set can be derived using two adjacent rules.

4.3 Single Antecedent Variable with Trapezoidal Fuzzy Sets

It is potentially very useful to extend the above interpolative reasoning method to apply to rules involving more complex fuzzy membership functions. This subsection describes the interpolation involving trapezoidal membership functions.

Consider a trapezoidal fuzzy set A, denoted as (a_0, a_1, a_2, a_3), as shown in Fig. 4.5, for notation convenience, the *bottom support*, *left slope*, *right slope* and *top support* of A are defined as $a_3 - a_0$, $a_1 - a_0$, $a_3 - a_2$ and $a_2 - a_1$, respectively. The representative value of A is defined as:

$$Rep(A) = \frac{1}{3}(a_0 + \frac{a_1 + a_2}{2} + a_3). \tag{4.24}$$

This definition subsumes the representative value of a triangular set as its specific case.

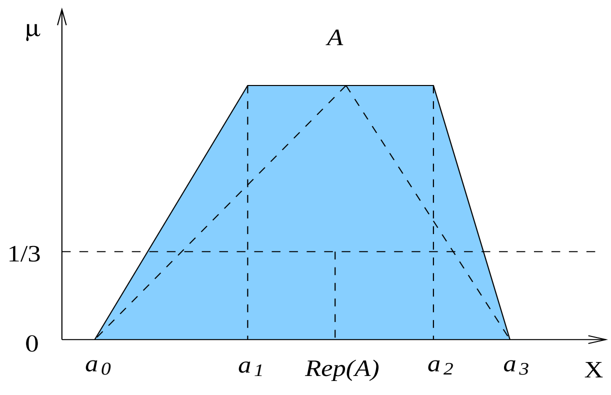

Figure 4.5: Representative value of a trapezoidal fuzzy set

This is because when a_1 and a_2 in a trapezoid are collapsed into a single value a_1, it degenerates into a triangle. In this case, the representative value definitions for trapezoidals (4.24) and triangles (4.1) remain the same. Of course, alternative definitions (e.g., $Rep(A) = \frac{a_0+a_1+a_2+a_3}{4}$) may be used, but this will destroy its compatibility with the triangular representation.

The calculation of the intermediate fuzzy rule $A' \Rightarrow B'$ follows a similar process as applying to triangular membership functions except that A' and B' here are trapezoidals rather than triangulars. It is straightforward to verify the extreme cases (such as if $A^* = A_1$ then $B^* = B_1$) in the same way as with triangular cases. To adapt the proposed method to be suitable for trapezoidal fuzzy sets, attention is only drawn to the two transformations.

Scale Transformation Given two *scale rates* s_b and s_t ($s_b \geq 0$ and $s_t \geq 0$) for bottom support scale and top support scale respectively, in order to transform the current bottom support $(a_3 - a_0)$ to a new bottom support $(s_b * (a_3 - a_0))$, and the top support $(a_2 - a_1)$ to a new top support $(s_t * (a_2 - a_1))$ while keeping the representative value and the ratio of left slope $(a'_1 - a'_0)$ to right slope $(a'_3 - a'_2)$ of the transformed fuzzy set the same as those of its original, that is, $Rep(A') = Rep(A)$ and $\frac{a'_1 - a'_0}{a'_3 - a'_2} = \frac{a_1 - a_0}{a_3 - a_2}$, the new a'_0, a'_1, a'_2 and a'_3 must satisfy (as illustrated in Fig. 4.6):

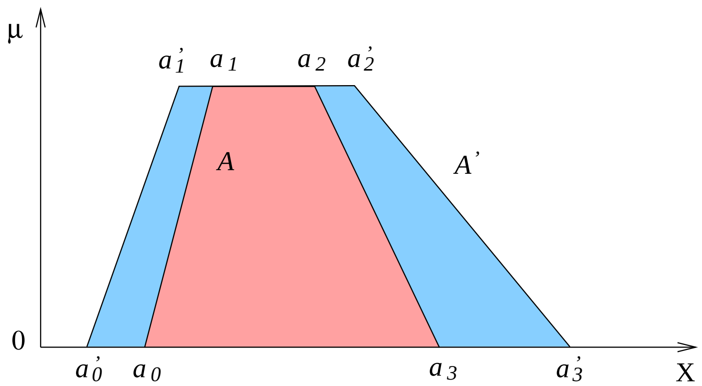

Figure 4.6: Trapezoidal scale transformation

$$a'_0 = A - \frac{C(2a_1 + a_3 - 2a_0 - a_2) - D(a_1 + a_2 - a_0 - a_3)}{B}, \tag{4.25}$$

$$a'_1 = A - \frac{C(a_0 + a_3 - a_1 - a_2) - D(5a_0 + a_2 - 5a_1 - a_3)}{B}, \tag{4.26}$$

$$a'_2 = A - \frac{C(a_0 + a_3 - a_1 - a_2) - D(a_1 + 5a_3 - a_0 - 5a_2)}{B}, \tag{4.27}$$

$$a'_3 = A - \frac{C(a_0 + 2a_2 - a_1 - 2a_3) - D(a_1 + a_2 - a_0 - a_3)}{B}, \tag{4.28}$$

where $A = \frac{2a_0+a_1+a_2+2a_3}{6}$, $B = 6(a_1 + a_3 - a_0 - a_2)$, $C = 2s_b(a_3 - a_0)$ and $D = s_t(a_2 - a_1)$. These results can be achieved by solving conditions below, imposed over the transformation:

$$\begin{cases} \frac{1}{3}(a'_0 + \frac{a'_1+a'_2}{2} + a'_3) = \frac{1}{3}(a_0 + \frac{a_1+a_2}{2} + a_3) \\ \frac{a'_1-a'_0}{a'_3-a'_2} = \frac{a_1-a_0}{a_3-a_2} \\ a'_3 - a'_0 = s_b(a_3 - a_0) \\ a'_2 - a'_1 = s_t(a_2 - a_1) \end{cases}$$

Note that the scale transformation guarantees that the transformed fuzzy sets are valid given that s_b and s_t ensure the bottom support of the resultant fuzzy set is wider than the top support and both left and right slopes are non-negative. This can be shown by

$$\begin{aligned} a'_1 - a'_0 &= \frac{(a_1 - a_0)(bot(A') - top(A'))}{a_1 + a_3 - a_0 - a_2} \geq 0, \\ a'_2 - a'_1 &= s_t(a_2 - a_1) \geq 0, \\ a'_3 - a'_2 &= \frac{(a_3 - a_2)(bot(A') - top(A'))}{a_1 + a_3 - a_0 - a_2} \geq 0, \end{aligned}$$

where $bot(A')$ and $top(A')$ stand for the bottom and top supports' lengths of transformed fuzzy set A', respectively. However, arbitrarily choosing s_t when s_b is fixed may lead to the top support of the resultant fuzzy set becoming wider than the bottom support. To avoid this, the *scale ratio* $\mathbb{S}_t$, which represents the actual increase of the ratios between the top supports and the bottom supports, before and after the transformation, normalised over the maximal possible such increase (in the sense that it does not lead to invalidity), is introduced to restrict s_t with respect to s_b:

$$\mathbb{S}_t = \begin{cases} \dfrac{\frac{s_t(a_2-a_1)}{s_b(a_3-a_0)} - \frac{a_2-a_1}{a_3-a_0}}{1 - \frac{a_2-a_1}{a_3-a_0}} & if\ s_t \geq s_b \geq 0 \\ \dfrac{\frac{s_t(a_2-a_1)}{s_b(a_3-a_0)} - \frac{a_2-a_1}{a_3-a_0}}{\frac{a_2-a_1}{a_3-a_0}} & if\ s_b \geq s_t \geq 0 \end{cases} \tag{4.29}$$

If $\mathbb{S}_t \in [0,1]$ (when $s_t \geq s_b > 0$) or $\mathbb{S}_t \in [-1,0]$ (when $s_b \geq s_t > 0$), $s_b(a_3 - a_0) \geq s_t(a_2 - a_1)$, i.e., $bot(A') \geq top(A')$. This can be shown as follows.

proof 2 *When* $s_t \geq s_b \geq 0$, *assume* $s_t(a_2 - a_1) > s_b(a_3 - a_0)$,

$$\therefore \frac{s_t(a_2 - a_1)}{s_b(a_3 - a_0)} > 1,$$

also

$$1 \geq \frac{a_2 - a_1}{a_3 - a_0} \geq 0,$$
$$\therefore \mathbb{S}_t > 1.$$

This conflicts with $\mathbb{S}_t \in [0,1]$, *and hence the assumption is wrong. So*

$$s_b(a_3 - a_0) \geq s_t(a_2 - a_1).$$

When $s_b \geq s_t \geq 0$,

$$\because a_3 - a_0 \geq a_2 - a_1,$$
$$\therefore s_b(a_3 - a_0) \geq s_t(a_2 - a_1).$$

If, however, two valid trapezoidal fuzzy sets $A = (a_0, a_1, a_2, a_3)$ and $A' = (a'_0, a'_1, a'_2, a'_3)$ happen to have the same representative value, the bottom scale rate of A, s_b, and the top scale ratio of A, $\mathbb{S}_t$, can be calculated as:

$$s_b = \frac{a'_3 - a'_0}{a_3 - a_0} \geq 0, \tag{4.30}$$

$$\mathbb{S}_t = \begin{cases} \dfrac{\frac{a'_2 - a'_1}{a'_3 - a'_0} - \frac{a_2 - a_1}{a_3 - a_0}}{1 - \frac{a_2 - a_1}{a_3 - a_0}} \in [0,1] & if\ \frac{a'_2 - a'_1}{a_2 - a_1} \geq \frac{a'_3 - a'_0}{a_3 - a_0} \geq 0 \\ \dfrac{\frac{a'_2 - a'_1}{a'_3 - a'_0} - \frac{a_2 - a_1}{a_3 - a_0}}{\frac{a_2 - a_1}{a_3 - a_0}} \in [-1,0] & if\ \frac{a'_3 - a'_0}{a_3 - a_0} \geq \frac{a'_2 - a'_1}{a_2 - a_1} \geq 0 \end{cases} \tag{4.31}$$

Thus, in this case, s_b is free to take on any positive value while $\mathbb{S}_t \in [0,1]$ or $\mathbb{S}_t \in [-1,0]$ (depending on whether $\frac{a'_2 - a'_1}{a_2 - a_1} \geq \frac{a'_3 - a'_0}{a_3 - a_0}$ or not) must hold given that A and A' are both valid. The closer is $\mathbb{S}_t$ to 0, the closer is the ratio between $top(A')$ and $bot(A')$ to that between $top(A)$ and $bot(A)$. Correspondingly, the closer is $\mathbb{S}_t$ to 1, the closer is the ratio between $top(A')$ and $bot(A')$ to 1. Similarly, the closer is $\mathbb{S}_t$ to -1, the closer is the ratio between $top(A')$ and $bot(A')$ to 0. The ranges of $\mathbb{S}_t$ values (as shown in (4.31)) are proven as follows:

proof 3 *When* $\frac{a_2'-a_1'}{a_2-a_1} \geq \frac{a_3'-a_0'}{a_3-a_0} \geq 0$,

$$\because 1 \geq \frac{a_2'-a_1'}{a_3'-a_0'} \geq \frac{a_2-a_1}{a_3-a_0} \geq 0,$$
$$\therefore 1 \geq \mathbb{S}_t \geq 0.$$

When $\frac{a_3'-a_0'}{a_3-a_0} \geq \frac{a_2'-a_1'}{a_2-a_1} \geq 0$,

$$\because 1 \geq \frac{a_2-a_1}{a_3-a_0} \geq \frac{a_2'-a_1'}{a_3'-a_0'} \geq 0,$$
$$\therefore 0 \geq \mathbb{S}_t \geq -1.$$

Move Transformation Given a moving distance l, in order to transform the current fuzzy set from the starting location a_0 to a new starting position a_0+l while keeping the same representative value, the length of support (a_3-a_0) and the length of the top support (a_2-a_1), i.e., $Rep(A') = Rep(A)$, $a_3'-a_0' = a_3-a_0$ and $a_2'-a_1' = a_2-a_1$, the new a_0', a_1', a_2' and a_3' must be (as shown in Fig. 4.7):

$$a_0' = a_0+l, \tag{4.32}$$
$$a_1' = a_1-2l, \tag{4.33}$$
$$a_2' = a_2-2l, \tag{4.34}$$
$$a_3' = a_3+l. \tag{4.35}$$

These can be obtained by solving the equations which are imposed to the transforma-

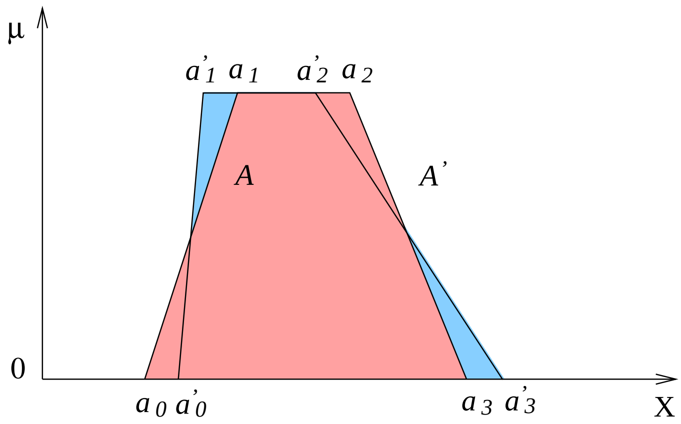

Figure 4.7: Trapezoidal move transformation

tion:

$$\begin{cases} \frac{1}{3}(a'_0 + \frac{a'_1+a'_2}{2} + a'_3) = \frac{1}{3}(a_0 + \frac{a_1+a_2}{2} + a_3) \\ a'_0 = a_0 + l \\ a'_3 - a'_0 = a_3 - a_0 \\ a'_2 - a'_1 = a_2 - a_1 \end{cases}$$

To ensure A' to be valid, the condition of $0 \leq l \leq l_{max} = (a_1 - a_0)/3$ must hold. If $l > l_{max}$, the transformation will generate invalid fuzzy sets. As with the triangular case, the *move ratio* $\mathbb{M}$ is introduced to avoid invalidity:

$$\mathbb{M} = \frac{l}{(a_1 - a_0)/3}. \tag{4.36}$$

If the move ratio $\mathbb{M} \in [0,1]$, then $l \leq l_{max}$ holds. Similar to triangular move transformation, there is another moving direction with $l \leq 0$. In that case the condition

$$\mathbb{M} = \frac{l}{(a_3 - a_2)/3} \in [-1,0] \tag{4.37}$$

is imposed to ensure the validity of the transformed fuzzy sets.

As with the scale transformation, if two valid trapezoidal sets $A = (a_0, a_1, a_2, a_3)$ and $A' = (a'_0, a'_1, a'_2, a'_3)$ which have the same representative value and have the same support lengths are given, the move ratio $\mathbb{M}$ can be calculated as follows:

$$\mathbb{M} = \begin{cases} \frac{3(a'_0 - a_0)}{a_1 - a_0} & if\ a'_0 \geq a_0 \\ \frac{3(a'_0 - a_0)}{a_3 - a_2} & if\ a'_0 \leq a_0 \end{cases} \tag{4.38}$$

As A and A' both are valid, $\mathbb{M} \in [0,1]$ (if $a'_0 \geq a_0$) or $\mathbb{M} \in [-1,0]$ (if $a'_0 \leq a_0$) must hold.

It is easy to see that trapezoidal transformations are a generalization of the triangular ones. In fact, if $a_1 = a_2$ the trapezoidal fuzzy set becomes a triangular one. Substituting $a_1 = a_2$ and $s_t = 0$ in the trapezoidal transformation formulae (4.25)-(4.28) and (4.32)-(4.35) leads to the same results by the triangular transformation formulae (4.14)-(4.16) and (4.18)-(4.20).

4.4 Single Antecedent Variable with Hexagonal Fuzzy Sets

A fairly general case, the interpolation of the hexagonal fuzzy sets, is described in this subsection. This is to be followed by dealing with the interpolation of any complex polygonal fuzzy membership functions in the next chapter. One open issue for such an extension is to determine the representative value for a given complex, asymmetrical polygonal fuzzy set. For computational simplicity, the average of the x coordinate values of all characteristic points is defined as the representative value for more complex polygonal fuzzy sets than trapezoidals.

Consider a generalised hexagonal fuzzy set A, denoted as $(a_0, a_1, a_2, a_3, a_4, a_5)$, as shown in Fig. 4.8, a_2 and a_3 are two normal, characteristic points (whose membership values are 1), a_0 and a_5 are two extreme, characteristic points (whose membership values are 0), and a_1 and a_4 are the two intermediate, characteristic points (whose membership values are the same and both are between 0 and 1 exclusively). For notational convenience, three *supports* (the horizontal intervals between a pair of characteristic points which involve the same membership value) are denoted as the *bottom support* $(a_5 - a_0)$, *middle support* $(a_4 - a_1)$ and *top support* $(a_3 - a_2)$, and four *slopes* (non-horizontal intervals between two consecutive characteristic points) are denoted as $a_1 - a_0$, $a_2 - a_1$, $a_4 - a_3$ and $a_5 - a_4$. Also, as indicated above, for computational simplicity, the representative value of A is defined as:

$$Rep(A) = \frac{a_0 + a_1 + a_2 + a_3 + a_4 + a_5}{6}. \tag{4.39}$$

Alternative definitions may be used to apply the transformations. For example, below

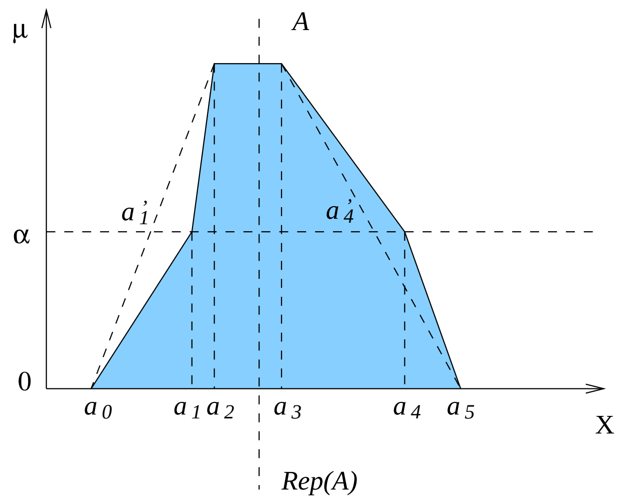

Figure 4.8: Representative value of a hexagonal fuzzy set

shows a definition making use of fuzzy membership values:

$$Rep(A) = \frac{1}{5-\alpha}[(a_0 + a_5) + (1 - \frac{\alpha}{2})(a_1 + a_4) + \frac{1}{2}(a_2 + a_3)], \tag{4.40}$$

where α is the membership value of both a_1 and a_4. This definition assigns different weights to different pairs of points. The weighted average is then taken as the representative value of such a fuzzy set. Another alternative definition, which is compatible to the less complex fuzzy sets (including triangular, trapezoidal and pentagonal fuzzy sets), can be defined as:

$$Rep(A) = \frac{1}{3}[a_0 + (1 - \frac{\alpha}{2})(a_1 - a_1') + \frac{1}{2}(a_2 + a_3) + (1 - \frac{\alpha}{2})(a_4 - a_4') + a_5], \tag{4.41}$$

where $a_1' = \alpha a_2 + (1-\alpha)a_0$ and $a_4' = \alpha a_3 + (1-\alpha)a_5$ (see Fig 4.8). Note that the interpolation by using either of these alternative definitions follows the same procedure as the one employing the simple definition (4.39).

The calculation of intermediate fuzzy rule $A' \Rightarrow B'$ follows the triangular or trapezoidal cases. Attention is again drawn to the scale and move transformations as described below.

Scale Transformation Given three *scale rates* s_b, s_m and s_t ($s_b \geq 0$, $s_m \geq 0$ and $s_t \geq 0$) representing the bottom support, middle support and top support scale respectively, the fuzzy set $A = (a_0, a_1, a_2, a_3, a_4, a_5)$ can be transformed to $A' = (a'_0, a'_1, a'_2, a'_3, a'_4, a'_5)$ by solving

$$\begin{cases} \frac{a'_0+a'_1+a'_2+a'_3+a'_4+a'_5}{6} = \frac{a_0+a_1+a_2+a_3+a_4+a_5}{6} \\ \frac{a'_1-a'_0}{a'_5-a'_4} = \frac{a_1-a_0}{a_5-a_4} \\ \frac{a'_2-a'_1}{a'_4-a'_3} = \frac{a_2-a_1}{a_4-a_3} \\ a'_5 - a'_0 = s_b(a_5 - a_0) \\ a'_4 - a'_1 = s_m(a_4 - a_1) \\ a'_3 - a'_2 = s_t(a_3 - a_2) \end{cases}$$

The solution of this is omitted here. As with the trapezoidal case, the resultant fuzzy set A' must have property that $a'_0 \leq a'_1 \leq a'_2 \leq a'_3 \leq a'_4 \leq a'_5$, given that the desired top support is narrower than the middle support and the middle support is narrower than the bottom support. Therefore, certain constraints should be imposed over s_m if s_b is fixed, and over s_t if s_m is fixed. For this reason, the scale ratios of middle and top supports of A, denoted as $\mathbb{S}_m$ and $\mathbb{S}_t$, are introduced to constrain the scale rates s_m and s_t respectively:

$$\mathbb{S}_m = \begin{cases} \dfrac{\frac{s_m(a_4-a_1)}{s_b(a_5-a_0)} - \frac{a_4-a_1}{a_5-a_0}}{1-\frac{a_4-a_1}{a_5-a_0}} & if\ s_m \geq s_b \geq 0 \\ \dfrac{\frac{s_m(a_4-a_1)}{s_b(a_5-a_0)} - \frac{a_4-a_1}{a_5-a_0}}{\frac{a_4-a_1}{a_5-a_0}} & if\ s_b \geq s_m \geq 0 \end{cases} \tag{4.42}$$

$$\mathbb{S}_t = \begin{cases} \dfrac{\frac{s_t(a_3-a_2)}{s_m(a_4-a_1)} - \frac{a_3-a_2}{a_4-a_1}}{1-\frac{a_3-a_2}{a_4-a_1}} & if\ s_t \geq s_m \geq 0 \\ \dfrac{\frac{s_t(a_3-a_2)}{s_m(a_4-a_1)} - \frac{a_3-a_2}{a_4-a_1}}{\frac{a_3-a_2}{a_4-a_1}} & if\ s_m \geq s_t \geq 0 \end{cases} \tag{4.43}$$

If $\mathbb{S}_m \in [0,1]$ (when $s_m \geq s_b \geq 0$) or $\mathbb{S}_m \in [-1,0]$ (when $s_b \geq s_m \geq 0$) whilst $\mathbb{S}_t \in [0,1]$ (when $s_t \geq s_m \geq 0$) or $\mathbb{S}_t \in [-1,0]$ (when $s_m \geq s_t \geq 0$), then $a'_5 - a'_0 \geq a'_4 - a'_1 \geq a'_3 - a'_2$. Interested readers may refer to proof 6 in subsection 5.2.2 for the discussion of the general polygonal fuzzy membership function case. The constraints of $\mathbb{S}_m$ and $\mathbb{S}_t$ along with the scale transformation thus lead to a unique and CNF set A'.

Conversely, if two valid hexagonal fuzzy sets $A = (a_0, a_1, a_2, a_3, a_4, a_5)$ and $A' = (a'_0, a'_1, a'_2, a'_3, a'_4, a'_5)$ which have the same representative value are given, the scale rate of the bottom support, s_b, and the scale ratios of the middle and top supports, $\mathbb{S}_m$ and $\mathbb{S}_t$, are calculated as:

$$s_b = \frac{a'_5 - a'_0}{a_5 - a_0} \geq 0 \tag{4.44}$$

$$\mathbb{S}_m = \begin{cases} \frac{\frac{a'_4-a'_1}{a'_5-a'_0} - \frac{a_4-a_1}{a_5-a_0}}{1-\frac{a_4-a_1}{a_5-a_0}} \in [0,1] & if\ \frac{a'_4-a'_1}{a_4-a_1} \geq \frac{a'_5-a'_0}{a_5-a_0} \geq 0 \\ \frac{\frac{a'_4-a'_1}{a'_5-a'_0} - \frac{a_4-a_1}{a_5-a_0}}{\frac{a_4-a_1}{a_5-a_0}} \in [-1,0] & if\ \frac{a'_5-a'_0}{a_5-a_0} \geq \frac{a'_4-a'_1}{a_4-a_1} \geq 0 \end{cases} \tag{4.45}$$

$$\mathbb{S}_t = \begin{cases} \frac{\frac{a'_3-a'_2}{a'_4-a'_1} - \frac{a_3-a_2}{a_4-a_1}}{1-\frac{a_3-a_2}{a_4-a_1}} \in [0,1] & if\ \frac{a'_3-a'_2}{a_3-a_2} \geq \frac{a'_4-a'_1}{a_4-a_1} \geq 0 \\ \frac{\frac{a'_3-a'_2}{a'_4-a'_1} - \frac{a_3-a_2}{a_4-a_1}}{\frac{a_3-a_2}{a_4-a_1}} \in [-1,0] & if\ \frac{a'_4-a'_1}{a_4-a_1} \geq \frac{a'_3-a'_2}{a_3-a_2} \geq 0 \end{cases} \tag{4.46}$$

Again, the proof of $\mathbb{S}_m \in [-1,1]$ and $\mathbb{S}_t \in [-1,1]$ given that A and A' are both valid is referred to proof 7 in subsection 5.2.2.

Move Transformation It is slightly more complicated to apply move transformations to hexagonal fuzzy sets although they still follow the same principle. Compared to the cases of triangular and trapezoidal fuzzy sets, where only one move transformation is carried out in order to obtain the resultant fuzzy set, this case needs two moves (referred to as *sub-moves* hereafter) to achieve the resultant fuzzy set.

Given two moving distances l_b and l_m, in order to transform the bottom support of the fuzzy set $A = (a_0, a_1, a_2, a_3, a_4, a_5)$ from the starting location a_0 to a new starting position $a'_0 = a_0 + l_b$, and to transform the middle support from a_1 to $a'_1 = a_1 + l_m$ while keeping the representative value, the lengths of three supports to remain the same (as shown in Fig. 4.9), two sub-moves are carried out.

First, a sub-move to the desired bottom support position is attempted. If it moves a_0 to the right position, $0 \leq l_b \leq l_{bmax} = (\frac{a_0+a_1+a_2}{3} - a_0)$ must hold. In the extreme position where $l_b = l_{bmax}$, the resultant fuzzy set $A'' = (a''_0, a''_1, a''_2, a''_3, a''_4, a''_5)$, i.e., the dotted hexagonal set in Fig. 4.9, has $a''_0 = a''_1 = a''_2$. If $l_b > l_{bmax}$, it will lead to an invalid fuzzy set. As with the triangular and trapezoidal cases, the bottom move ratio

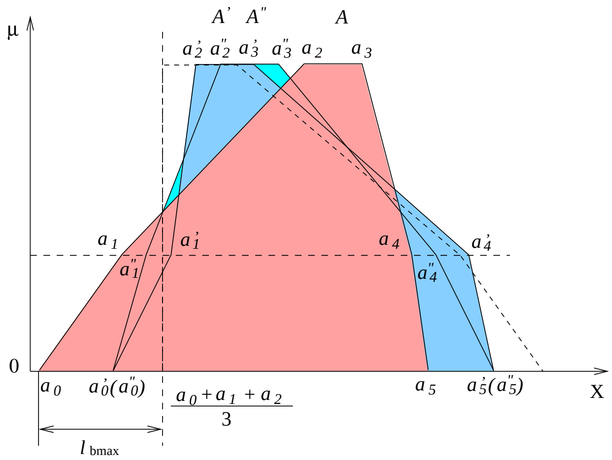

Figure 4.9: Hexagonal bottom move transformation

is introduced to avoid this potential invalidity:

$$\mathbb{M}_b = \frac{l_b}{\frac{a_0+a_1+a_2}{3} - a_0}. \tag{4.47}$$

If the move ratio $\mathbb{M}_b \in [0,1]$, then $l_b \leq l_{max}$ holds. The moving distance of the point a_i ($i = 0,1,2$) is calculated by multiplying $\mathbb{M}_b$ with the distance between the extreme position ($\frac{a_0+a_1+a_2}{3}$) and itself. In so doing, a_0, a_1 and a_2 will move the same proportion of their respective distances to the extreme positions. The other three points a_3, a_4 and a_5 can therefore be determined by attaining the same lengths of the three supports, respectively. The fuzzy set A' after this sub-move is thus calculated by:

$$a''_0 = a_0 + \mathbb{M}_b(\frac{a_0+a_1+a_2}{3} - a_0), \tag{4.48}$$

$$a''_1 = a_1 + \mathbb{M}_b(\frac{a_0+a_1+a_2}{3} - a_1), \tag{4.49}$$

$$a''_2 = a_2 + \mathbb{M}_b(\frac{a_0+a_1+a_2}{3} - a_2), \tag{4.50}$$

$$a''_3 = a_3 + \mathbb{M}_b(\frac{a_0+a_1+a_2}{3} - a_2), \tag{4.51}$$

$$a''_4 = a_4 + \mathbb{M}_b(\frac{a_0+a_1+a_2}{3} - a_1), \tag{4.52}$$

$$a''_5 = a_5 + \mathbb{M}_b(\frac{a_0+a_1+a_2}{3} - a_0). \tag{4.53}$$

From (4.48)-(4.53), it is clear that A'' is valid as the following holds given $\mathbb{M}_b \in [0,1]$:

$$a_1'' - a_0'' = (a_1 - a_0)(1 - \mathbb{M}_b) \geq 0,$$
$$a_2'' - a_1'' = (a_2 - a_1)(1 - \mathbb{M}_b) \geq 0,$$
$$a_3'' - a_2'' = a_3 - a_2 \geq 0,$$
$$a_4'' - a_3'' = a_4 - a_3 + \mathbb{M}_b(a_2 - a_1) \geq 0,$$
$$a_5'' - a_4'' = a_5 - a_4 + \mathbb{M}_b(a_1 - a_0) \geq 0.$$

It can be verified that A'' has the same representative value as A. This is because, according to equations (4.48)-(4.53),

$$\begin{aligned} Rep(A'') &= \frac{a_0'' + a_1'' + a_2'' + a_3'' + a_4'' + a_5''}{6} \\ &= \frac{a_0 + a_1 + a_2 + a_3 + a_4 + a_5}{6} \\ &= Rep(A). \end{aligned}$$

For the opposite moving direction where $l_b \leq 0$, the condition

$$\mathbb{M}_b = \frac{l_b}{a_5 - \frac{a_3 + a_4 + a_5}{3}} \in [-1, 0] \tag{4.54}$$

is imposed to ensure the validity of the transformed fuzzy set. The results of A'' can similarly be written as:

$$a_0'' = a_0 + \mathbb{M}_b(a_5 - \frac{a_3 + a_4 + a_5}{3}), \tag{4.55}$$
$$a_1'' = a_1 + \mathbb{M}_b(a_4 - \frac{a_3 + a_4 + a_5}{3}), \tag{4.56}$$
$$a_2'' = a_2 + \mathbb{M}_b(a_3 - \frac{a_3 + a_4 + a_5}{3}), \tag{4.57}$$
$$a_3'' = a_3 + \mathbb{M}_b(a_3 - \frac{a_3 + a_4 + a_5}{3}), \tag{4.58}$$
$$a_4'' = a_4 + \mathbb{M}_b(a_4 - \frac{a_3 + a_4 + a_5}{3}), \tag{4.59}$$
$$a_5'' = a_5 + \mathbb{M}_b(a_5 - \frac{a_3 + a_4 + a_5}{3}). \tag{4.60}$$

Of course, it can be proved from (4.55)-(4.60) that this resultant fuzzy set is indeed valid given $\mathbb{M}_b \in [-1, 0]$:

$$\begin{aligned}
a''_1 - a''_0 &= a_1 - a_0 + \mathbb{M}_b(a_4 - a_5) \geq 0, \\
a''_2 - a''_1 &= a_2 - a_1 + \mathbb{M}_b(a_3 - a_4) \geq 0, \\
a''_3 - a''_2 &= a_3 - a_2 \geq 0, \\
a''_4 - a''_3 &= (a_4 - a_3)(1 + \mathbb{M}_b) \geq 0, \\
a''_5 - a''_4 &= (a_5 - a_4)(1 + \mathbb{M}_b) \geq 0.
\end{aligned}$$

Again, A'' and A have the same representative value, ensured by (4.55)-(4.60).

In both cases ($l_b \geq 0$ and $l_b \leq 0$), $a''_0 = a_0 + l_b$ holds. This means the bottom support of A is moved to the desired place after the first sub-move. So the second sub-move is aimed to move the middle and the top supports to the desired places from A'' to A' as shown in Fig. 4.9. This sub-move does not affect the place of the bottom support as it is already in the right place. Considering moving the middle support to the right direction (i.e., the new move displacement $l'_m = l_m - (a''_1 - a_1) \geq 0$), this move is almost the same as the move proposed for a trapezoidal fuzzy set except that the maximal moving distance (in the sense that it does not lead to invalidity) should be less than, or at most equal to $\frac{a''_2 - a''_1}{2}$ (not $\frac{a''_2 - a''_1}{3}$ as in the trapezoidal case due to the difference in the representative definition for hexagonal fuzzy sets). This is because the maximal moving distance is also constrained to the bottom support (i.e., $l'_m \leq a''_5 - a''_4$) as it may move a''_4 exceeding a''_5. It is intuitive to pick the minimal value of the two distances as the maximal moving distance. The move ratio therefore can be defined as:

$$\mathbb{M}_m = \frac{l_m - (a''_1 - a_1)}{min\{\frac{a''_2 - a''_1}{2}, a''_5 - a''_4\}}. \tag{4.61}$$

When applying the second sub-move, considering both upper and lower invalidity may arise, the *applied move ratio* $\mathbb{M}'_m$ is introduced as:

$$\mathbb{M}'_m = \mathbb{M}_m \frac{min\{\frac{a''_2 - a''_1}{2}, a''_5 - a''_4\}}{\frac{a''_2 - a''_1}{2}}. \tag{4.62}$$

If $\mathbb{M}_m \in [0, 1]$, $\mathbb{M}'_m \in [0, \mathbb{M}_m]$. The introduction of applied move ratio avoids the potential lower invalidity when applying the sub-move as follows:

$$
\begin{aligned}
a'_0 &= a''_0, && (4.63)\\
a'_1 &= a''_1 + \mathbb{M}'_m(\frac{a''_1+a''_2}{2} - a''_1), && (4.64)\\
a'_2 &= a''_2 + \mathbb{M}'_m(\frac{a''_1+a''_2}{2} - a''_2), && (4.65)\\
a'_1 &= a''_3 + \mathbb{M}'_m(\frac{a''_1+a''_2}{2} - a''_2), && (4.66)\\
a'_1 &= a''_4 + \mathbb{M}'_m(\frac{a''_1+a''_2}{2} - a''_1), && (4.67)\\
a'_5 &= a''_5. && (4.68)
\end{aligned}
$$

Merging (4.61) and (4.62) into (4.64) and (4.65) leads to $a'_1 = a_1 + l_m$ and $a'_2 = a_2 - l_b - l_m$, which are the desired positions for a_1 and a_2 to be moved on to, respectively. It can also be shown that A' is an NVF fuzzy set and $Rep(A') = Rep(A'') = Rep(A)$. All these properties are maintained if in the opposite case where $l'_m \leq 0$.

As discussed above, if given two move ratios $\mathbb{M}_b \in [-1,1]$ and $\mathbb{M}_m \in [-1,1]$, the two sub-moves transform the given NVF set $A = (a_0,a_1,a_2,a_3,a_4,a_5)$ to a new NVF set $A' = (a'_0,a'_1,a'_2,a'_3,a'_4,a'_5)$ while keeping the representative values and the lengths or supports to be the same.

Conversely, if two valid hexagonal fuzzy sets $A = (a_0,a_1,a_2,a_3,a_4,a_5)$ and $A' = (a'_0,a'_1,a'_2,a'_3,a'_4,a'_5)$ which have the same representative value and have the same support lengths are given, the move ratios which are calculated in an order from bottom to top must lie between $[-1,1]$. First, the bottom move ratio is computed by:

$$
\mathbb{M}_b = \begin{cases} \frac{a'_0 - a_0}{\frac{a_0+a_1+a_2}{3} - a_0} \in [0,1] & if\ a'_0 \geq a_0 \\ \frac{a'_0 - a_0}{a_5 - \frac{a_3+a_4+a_5}{3}} \in [-1,0] & if\ a'_0 \leq a_0 \end{cases} \tag{4.69}
$$

It is used to carry out the first sub-move of A to generate $A'' = (a''_0,a''_1,a''_2,a''_3,a''_4,a''_5)$ according to (4.48) - (4.53) or (4.55) - (4.60). Then, the middle move ratio can be calculated by:

$$
\mathbb{M}_m = \begin{cases} \frac{a'_1 - a''_1}{min\{\frac{a''_2 - a''_1}{2}, a''_5 - a''_4\}} \in [0,1] & if\ a'_1 \geq a''_1 \\ \frac{a'_1 - a''_1}{min\{\frac{a''_4 - a''_3}{2}, a''_1 - a''_0\}} \in [-1,0] & if\ a'_1 \leq a''_1 \end{cases} \tag{4.70}
$$

4.5 Outline of the Method

On top of the *scale* and *move transformations*, an *integrated transformation*, denoted $T(A,A')$, between two fuzzy sets A and A' can be introduced such that A' is the derived NVF set of A by applying both transformation components. Obviously, two integrated transformations are said to be identical if and only if both of their *scale rate*, *scale ratios* (for polygonal fuzzy sets more complex than triangular) and *move ratios* are equal.

As indicated earlier in (4.13), it is intuitive to maintain the similarity degree between the consequent parts B' and B^* to be the same as that between the antecedent parts A' and A^*, in performing interpolative reasoning. Now that the integrated transformation allows the similarity degree between two fuzzy sets to be measured by the *scale rate*, *scale ratios* (for fuzzy polygonal sets more complex than triangular) and *move ratios*, the desired conclusion B^* can be obtained by satisfying the following (as shown in Fig. 4.10 for an interpolation involving triangular fuzzy sets):

$$T(B',B^*) = T(A',A^*). \tag{4.71}$$

That is, the parameters of *scale rate*, *scale ratios* and *move ratios* calculated from A'

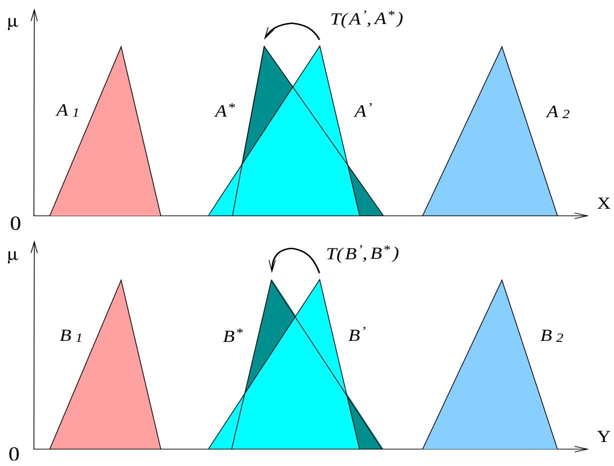

Figure 4.10: Proposed interpolative reasoning method

to A^* are used to compute B^* from B'. Clearly, B^* will then retain the same similarity degree as that between the antecedent parts A' and A^*.

4.6 Summary

This chapter proposes a novel interpolative reasoning method based on specific examples (triangular, trapezoidal and hexagonal fuzzy sets). First an intermediate fuzzy rule is constructed by its two adjacent rules. Then it together with the observation are converted into the final results by proposed scale and move transformations, which ensure unique as well as normal and valid results. The generalization of this work will be discussed in the next chapter.

Chapter 5

Transformation Based Interpolation: General Approach

This chapter extends the work presented in Chapter 4 in four aspects: 1) the representative value definitions are generalised, which provides a degree of freedom to meet particular application requirements; 2) the interpolation method is extended to deal with arbitrarily complex polygonal fuzzy sets; 3) further development has been made on the scale and move transformations; and 4) the interpolation (and extrapolation) method is extended to deal with multiple antecedent variables and/or multiple rules. Numerical examples have been illustrated to show the use of the interpolation methods.

5.1 General Representative Value (RV) Definition

To facilitate the discussion of the transformation based interpolation method, the *representative value* of the polygonal fuzzy sets involved must be defined beforehand. This value represents the overall location of the fuzzy set and it guides the transformations as presented in the next section. As different RV definitions lead to different interpolation results (although the transformations apply in the same manner), it provides the flexibility to choose proper RV definitions to suit different application requirements.

The RV definitions deployed in the previous chapter are firstly reviewed. Considering a triangular fuzzy set A, denoted as (a_0, a_1, a_2), as shown in Fig. 5.1, the RV definition is written as follows:

$$Rep(A) = \frac{a_0 + a_1 + a_2}{3}. \tag{5.1}$$

This happens to be the centre of gravity of the triangular fuzzy set [HS03].

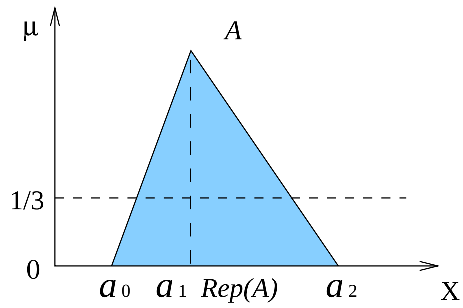

Figure 5.1: The RV of a triangular fuzzy set

To be compatible to this definition, the definition of RV for a trapezoidal fuzzy set $A = (a_0, a_1, a_2, a_3)$ (as shown in Fig. 5.2) is calculated as:

$$Rep(A) = \frac{1}{3}(a_0 + \frac{a_1 + a_2}{2} + a_3). \tag{5.2}$$

This definition subsumes the RV of a triangular fuzzy set as its specific case. This is

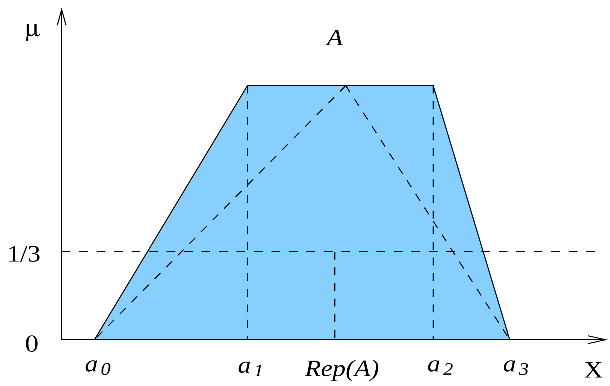

Figure 5.2: The RV of a trapezoidal fuzzy set

because when a_1 and a_2 in a trapezoidal fuzzy set are collapsed into a single value a_1, it degenerates into a triangular one. In this case, the representative value definitions for trapezoidals (5.2) and triangles (5.1) remain the same.

It becomes more complicated to deal with more complex fuzzy sets such as hexagonal fuzzy sets (as shown in Fig. 5.3). The simplest solution is calculating the average of values of all points as the RV of that fuzzy set. The work of [HS04a] also suggested the possible RV definitions as:

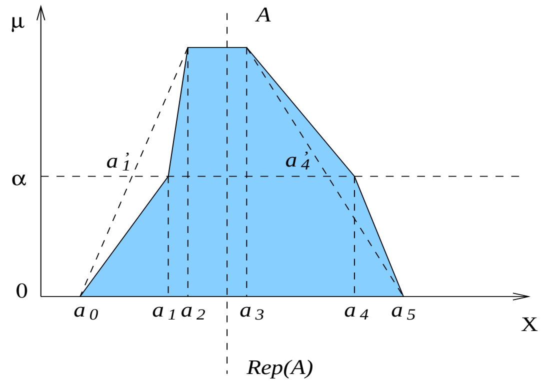

Figure 5.3: The RV of a hexagonal fuzzy set

$$Rep(A) = \frac{(a_0 + a_5) + (1 - \frac{\alpha}{2})(a_1 + a_4) + \frac{1}{2}(a_2 + a_3)}{5 - \alpha}, \tag{5.3}$$

where α is the membership value of both a_1 and a_4. This definition assigns different pairs of points with different weights. The weighted average is then taken as the representative value.

Another alternative definition for the hexagonal fuzzy sets is compatible to the less complex fuzzy sets including triangular, trapezoidal and pentagonal fuzzy sets. For example, if a_1 and a_4 happen to be on the lines between a_0, a_2 and a_3, a_5, respectively, such a hexagonal fuzzy set becomes a trapezoidal set, the definition is thus equal to (5.2). Such a compatible definition can be written as:

$$\begin{aligned} Rep(A) = & \frac{1}{3}[a_0 + (1 - \frac{\alpha}{2})(a_1 - a'_1) + \frac{1}{2}(a_2 + a_3) \\ & + (1 - \frac{\alpha}{2})(a_4 - a'_4) + a_5], \end{aligned} \tag{5.4}$$

where $a'_1 = \alpha a_2 + (1 - \alpha) a_0$ and $a'_4 = \alpha a_3 + (1 - \alpha) a_5$ (see Fig 5.3).

After the review of the previously adopted RV definitions, now considering the general RV definition for an arbitrary polygonal fuzzy set with n characteristic points, $A = (a_0, \ldots, a_{n-1})$, as shown in (5.4). Note that the two top points (of membership value 1) do not need to be different. If they happen to have the same value, they are collapsed into one. Also, although the figure explicitly assumes that evenly paired characteristic points are on each α-cut, this doesn't affect the generality of the fuzzy set as artificial characteristic points can be created to form evenly paired characteristic points. Clearly, a general fuzzy membership function with n characteristic points has $\lfloor \frac{n}{2} \rfloor$ *supports* (horizontal intervals between a pair of characteristic points which have the same membership value) and $2(\lceil \frac{n}{2} \rceil - 1)$ *slopes* (non-horizontal intervals between two consecutive characteristic points). A general RV definition of such an arbitrary

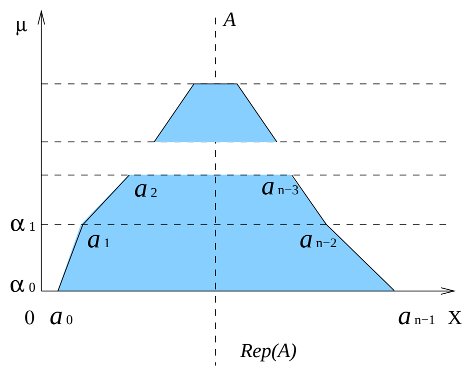

Figure 5.4: The RV of an arbitrarily complex fuzzy set

polygonal fuzzy set can be written as:

$$Rep(A) = \sum_{i=0}^{n-1} w_i a_i, \tag{5.5}$$

where w_i is the weight assigned to point a_i.

The simplest case (which is denoted as the *average RV* definition hereafter) is that all points take the same weight value, i.e., $w_i = \frac{1}{n}$. The RV is therefore written as:

$$Rep(A) = \frac{1}{n} \sum_{i=0}^{n-1} a_i. \tag{5.6}$$

Note that (5.1) belongs to this definition.

Given a RV definition, if the RV of a fuzzy set by using more characteristic points keeps the same value as that of the same fuzzy set but by using less characteristic points, such a definition is called a *compatible RV* definition. One such solution can be specified by the following rules:

1. Artificial characteristic points are assigned weights of 0.
2. Bottom points (of membership value 0) are assigned weights of $\frac{1}{3}$.
3. Top points (of membership value 1) are assigned weights of $\frac{1}{3}$ if the fuzzy sets have odd characteristic points (e.g., triangular sets), $\frac{1}{6}$ otherwise (e.g., trapezoidal).
4. Intermediate characteristic points are assigned weights of $\frac{1}{3}$ if the fuzzy sets have odd characteristic points, $\frac{1}{3}(1-\frac{\alpha_i}{2})$ otherwise, where α_i is the fuzzy membership value of characteristic point a_i, $i = \{0, \ldots, n-1\}$.

Another alternative definition (denoted as the *weighted average RV* definition) assumes that the weights increase (or decrease) upwardly from the bottom support to the top support. This weight assignment strategy is inspired by the assumption that different characteristic points may have varied weights, and the weights may have something to do with the fuzzy membership values. For instance, assuming the weights increase upwardly from $\frac{1}{2}$ to 1, the weight w_i can thus be calculated by $w_i = \frac{1+\alpha_i}{2}$ (where α_i is the fuzzy membership value of a_i, $i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 1\}$), and then be normalised by the summary of w_i, $i = \{0, \ldots, n-1\}$. The RV therefore can be written as:

$$Rep(A) = \frac{\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \frac{1+\alpha_i}{2}(a_i + a_{n-1-i})}{\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \frac{1+\alpha_i}{2}}. \tag{5.7}$$

Also, the definition (5.3) is another particular case of the weight average definition although the weight assignment is different : the weights decrease upwardly from 1 to $\frac{1}{2}$.

One of the most widely used defuzzification methods – the centre of core can also be used to define the *centre of core RV*. In this case, the RV is solely determined by those points with a fuzzy membership value of 1:

$$Rep(A) = \frac{1}{2}(a_{\lceil \frac{n}{2} \rceil - 1} + a_{n - \lceil \frac{n}{2} \rceil}). \tag{5.8}$$

The general RV definition can be simplified if the lengths of $\lfloor \frac{n}{2} \rfloor$ supports $S_0, \ldots, S_{\lfloor \frac{n}{2} \rfloor - 1}$ (the index in ascending order from the bottom to the top) are known. As $a_{n-1-i} = a_i + S_i$, $i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$, the general RV definition (5.5) can thus be re-written as:

$$\begin{aligned} Rep(A) &= a_0(w_0 + w_{n-1}) + S_0 w_{n-1} + \ldots \\ &\quad + a_{\lceil \frac{n}{2} \rceil - 1}(w_{\lceil \frac{n}{2} \rceil - 1} + w_{n - \lceil \frac{n}{2} \rceil}) + S_{\lceil \frac{n}{2} \rceil - 1} w_{n - \lceil \frac{n}{2} \rceil} \\ &= \sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} a_i(w_i + w_{n-1-i}) + C, \end{aligned} \tag{5.9}$$

where $C = S_0 w_{n-1} + \ldots + S_{\lceil \frac{n}{2} \rceil - 1} w_{n - \lceil \frac{n}{2} \rceil}$ is a constant. From this definition, the representative value acts as a function with respect to the values of the points on the left side of the fuzzy set.

The general RV definition (5.5) subsumes all the RV definitions used earlier on in [HS03, HS04b, HS05d]. It provides more room to define suitable RVs for different applications. In fact the general definition is the linear combination of values of all characteristic points. Beyond this, non-linear combination of such values, such as the one including the product of two or more points' values, is not valid as the interpolation is itself linear.

5.2 Base Case

5.2.1 Construct the Intermediate Rule

In fuzzy interpolation, the simplest case is commonly used to demonstrate the underlying techniques without losing any generality. That is, given two adjacent rules as follows

$$If\ X\ is\ A_1\ then\ Y\ is\ B_1,$$
$$If\ X\ is\ A_2\ then\ Y\ is\ B_2,$$

which are denoted as $A_1 \Rightarrow B_1$, $A_2 \Rightarrow B_2$ respectively, together with the observation A^* which is located between fuzzy sets A_1 and A_2, the interpolation is supposed to achieve the fuzzy result B^*. In another form this simplest case can be represented through the modus ponens interpretation (5.10), and as illustrated in Fig. 5.5.

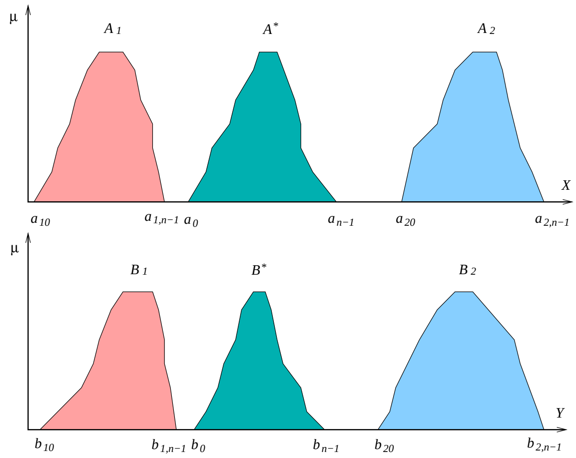

Figure 5.5: Interpolation with arbitrary polygonal fuzzy membership functions

$$
\begin{array}{l}
\text{observation: } X \text{ is } A^* \\
\text{rules: if } X \text{ is } A_1 \text{, then } Y \text{ is } B_1 \\
\qquad\quad \text{if } X \text{ is } A_2 \text{, then } Y \text{ is } B_2 \\
\hline
\text{conclusion: } Y \text{ is } B^*?
\end{array}
\tag{5.10}
$$

Here, $A_i = (a_{i0}, \ldots, a_{i,n-1})$, $B_i = (b_{i0}, \ldots, b_{i,n-1})$, $i = \{1,2\}$, and $A^* = (a_0, \ldots, a_{n-1})$, $B^* = (b_0, \ldots, b_{n-1})$.

The transformation based interpolation method begins with constructing a new fuzzy set A' which has the same RV as A^*. To support this work, the distance between A_1 and A_2 is herein re-represented by the following:

$$d(A_1, A_2) = d(Rep(A_1), Rep(A_2)). \tag{5.11}$$

An interpolative ratio λ_{Rep} $(0 \leq \lambda_{Rep} \leq 1)$ is introduced to represent the important impact of A_2 (with respect to A_1) when constructing A':

$$
\begin{aligned}
\lambda_{Rep} &= \frac{d(A_1, A^*)}{d(A_1, A_2)} \\
&= \frac{d(Rep(A_1), Rep(A^*))}{d(Rep(A_1), Rep(A_2))}.
\end{aligned}
\tag{5.12}
$$

That is to say, if $\lambda_{Rep} = 0$, A_2 plays no part in the construction of A'. While if $\lambda_{Rep} = 1$, A_2 plays a full role in determining A'. Then by using the simplest linear interpolation,

a'_i, $i = \{0, \ldots, n-1\}$, of A' are calculated as follows:

$$a'_i = (1-\lambda_{Rep})a_{1i} + \lambda_{Rep}a_{2i}, \tag{5.13}$$

which are collectively abbreviated to

$$A' = (1-\lambda_{Rep})A_1 + \lambda_{Rep}A_2. \tag{5.14}$$

Now, A' has the same representative value as A^*.

proof 4 *As* $Rep(A') = \sum_{i=0}^{n-1} w_i a'_i$. *With (5.13) and (5.12),*

$$\begin{aligned} & Rep(A') \\ & = \sum_{i=0}^{n-1} w_i[(1-\lambda_{Rep})a_{1i} + \lambda_{Rep}a_{2i}] \\ & = (1-\lambda_{Rep})\sum_{i=0}^{n-1} w_i a_{1i} + \lambda_{Rep}\sum_{i=0}^{n-1} w_i a_{2i} \\ & = (1-\lambda_{Rep})Rep(A_1) + \lambda_{Rep}Rep(A_2) \\ & = Rep(A^*) \end{aligned} \tag{5.15}$$

Also, it is worth noting that A' is a valid fuzzy set as the following holds given $a_{1i} \leq a_{1,i+1}$, $a_{2i} \leq a_{2,i+1}$, where $i = \{0, \ldots, n-2\}$, and $0 \leq \lambda_{Rep} \leq 1$:

$$\begin{aligned} & a'_{i+1} - a'_i \\ & = (1-\lambda_{Rep})(a_{1,i+1} - a_{1i}) + \lambda_{Rep}(a_{2,i+1} - a_{2i}) \geq 0. \end{aligned}$$

Similarly, the consequent fuzzy set B' can be obtained by

$$B' = (1-\lambda_{Rep})B_1 + \lambda_{Rep}B_2. \tag{5.16}$$

In so doing, the newly derived rule $A' \Rightarrow B'$ involves the use of only normal and valid fuzzy sets.

As $A' \Rightarrow B'$ is derived from $A_1 \Rightarrow B_1$ and $A_2 \Rightarrow B_2$, it is feasible to perform fuzzy reasoning with this new rule without further reference to its originals. The interpolative reasoning problem is therefore changed from (5.10) to the new modus ponens interpretation:

$$\begin{array}{l} \text{observation: } X \text{ is } A^* \\ \text{rule: if } X \text{ is } A' \text{, then } Y \text{ is } B' \\ \hline \text{conclusion: } Y \text{ is } B^*? \end{array} \tag{5.17}$$

This interpretation retains the same results as (5.10) in dealing with the extreme cases: If $A^* = A_1$, then from (5.12) $\lambda_{Rep} = 0$, and according to (5.14) and (5.16), $A' = A_1$ and $B' = B_1$, so the conclusion $B^* = B_1$. Similarly, if $A^* = A_2$, then $B^* = B_2$.

Other than the extreme cases, *similarity* measures are used to support the application of this new modus ponens. In particular, (5.17) can be interpreted as

$$The\ more\ similar\ X\ to\ A',\ the\ more\ similar\ Y\ to\ B'. \tag{5.18}$$

Suppose that a certain degree of similarity between A' and A^* is established, it is intuitive to require that the consequent parts B' and B^* attain the same similarity degree. The question is now how to obtain an operator which can represent the similarity degree between A' and A^*, and to allow transforming B' to B^* with the desired degree of similarity. To this end, the following two component transformations are proposed as follows.

5.2.2 Scale Transformation for General RV Definition

Consider applying scale transformation to an arbitrary polygonal fuzzy membership function $A = (a_0, \ldots, a_{n-1})$ (as shown in Fig. 5.6) to generate $A' = (a'_0, \ldots, a'_{n-1})$ such that they have the same RV, and $a'_{n-1-i} - a'_i = s_i(a_{n-1-i} - a_i)$, where s_i are scale rates and $i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$. In order to achieve this, $\lfloor \frac{n}{2} \rfloor$ equations $a'_{n-1-i} - a'_i = s_i(a_{n-1-i} - a_i)$, $i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$, are imposed to obtain the supports with desired lengths, and $(\lceil \frac{n}{2} \rceil - 1)$ equations $\frac{a'_{i+1} - a'_i}{a'_{n-1-i} - a'_{n-2-i}} = \frac{a_{i+1} - a_i}{a_{n-1-i} - a_{n-2-i}}$, $i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}$ are imposed to equalize the ratios between the left $(\lceil \frac{n}{2} \rceil - 1)$ slopes' lengths and the right $(\lceil \frac{n}{2} \rceil - 1)$ slopes' lengths of A' to those counterparts of the original fuzzy set A. The equation $\sum_{i=0}^{n-1} w_i a'_i = \sum_{i=0}^{n-1} w_i a_i$ which ensures the same representative values before and after the transformation is added to make up of $\lfloor \frac{n}{2} \rfloor + (\lceil \frac{n}{2} \rceil - 1) + 1 = n$ equations.

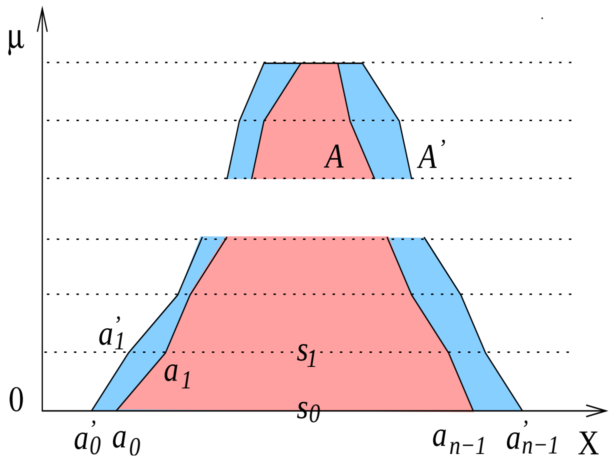

Figure 5.6: Scale transformation

All these n equations are collectively written as:

$$\begin{cases} a'_{n-1-i} - a'_i = s_i(a_{n-1-i} - a_i) = S_i \\ (i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}) \\ \frac{a'_{i+1} - a'_i}{a'_{n-1-i} - a'_{n-2-i}} = \frac{a_{i+1} - a_i}{a_{n-1-i} - a_{n-2-i}} = R_i \\ (i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}) \\ \sum_{i=0}^{n-1} w_i a'_i = \sum_{i=0}^{n-1} w_i a_i \end{cases} \tag{5.19}$$

where S_i is the ith support length of the resultant fuzzy set and R_i is the ratio between the left ith slope length and the right ith slope length. Solving these n equations simultaneously results in a unique and valid fuzzy set A' given that the resultant set has a descending order of the support lengths from the bottom to the top. This can be proved as follows.

proof 5 *As* $R_i \geq 0$ $(i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\})$ *and* $S_i \geq S_{i+1}$ $(i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 2\})$, *from*

(5.19), the conclusions below can be drawn:

$$\begin{cases} a'_{i+1} - a'_i = \frac{R_i}{1+R_i}(S_i - S_{i+1}) \geq 0 \\ i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\} \\ \\ a'_{n-\lceil \frac{n}{2} \rceil} - a'_{\lceil \frac{n}{2} \rceil - 1} = S_{\lceil \frac{n}{2} \rceil - 1} \geq 0 \\ \\ a'_{i+1} - a'_i = \frac{1}{1+R_{n-i-2}}(S_{n-i-2} - S_{n-i-1}) \geq 0 \\ i = \{n - \lceil \frac{n}{2} \rceil, \ldots, n-2\} \end{cases}$$

It can be concluded from this proof that, if a fuzzy set A and the support scale rates s_i are given, the RV definition doesn't affect the geometrical shape of the resultant fuzzy set after the scale transformation. Instead, it only affects the position of this fuzzy set.

However, arbitrarily choosing the ith support scale rate when the $(i-1)$th scale rate is fixed may lead the ith support to becoming wider than the $(i-1)$th support, i.e., $S_i > S_{i-1}$. To avoid this, the ith *scale ratio* $\mathbb{S}_i$, which represents the actual increase of the ratios between the ith supports and the $(i-1)$th supports, before and after the transformation, normalised over the maximal possible such increase (in the sense it does not lead to invalidity), is introduced to restrict s_i with respect to s_{i-1}:

$$\mathbb{S}_i = \begin{cases} \frac{\frac{s_i(a_{n-i-1}-a_i)}{s_{i-1}(a_{n-i}-a_{i-1})} - \frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}}{1 - \frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}} & if\ s_i \geq s_{i-1} \geq 0 \\ \frac{\frac{s_i(a_{n-i-1}-a_i)}{s_{i-1}(a_{n-i}-a_{i-1})} - \frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}}{\frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}}} & if\ s_{i-1} \geq s_i \geq 0 \end{cases} \quad (5.20)$$

If $\mathbb{S}_i \in [0,1]$ (when $s_i \geq s_{i-1} \geq 0$) or $\mathbb{S}_i \in [-1,0]$ (when $s_{i-1} \geq s_i \geq 0$), $S_{i-1} \geq S_i$. This can be shown as follows.

proof 6 *When* $s_i \geq s_{i-1} \geq 0$*, assume* $S_i > S_{i-1}$*, i.e,* $s_i(a_{n-i-1} - a_i) > s_{i-1}(a_{n-i} - a_{i-1})$,

$$\therefore \frac{s_i(a_{n-i-1} - a_i)}{s_{i-1}(a_{n-i} - a_{i-1})} > 1.$$

Also,

$$\because 1 \geq \frac{a_{n-i-1} - a_i}{a_{n-i} - a_{i-1}} \geq 0,$$
$$\therefore \mathbb{S}_i > 1.$$

This conflicts with $\mathbb{S}_i \in [0,1]$*. The assumption is therefore wrong. So* $S_{i-1} \geq S_i$*.*
When $s_{i-1} \geq s_i \geq 0$*,*

$$\because a_{n-i} - a_{i-1} \geq a_{n-i-1} - a_i,$$
$$\therefore s_{i-1}(a_{n-i} - a_{i-1}) \geq s_i(a_{n-i-1} - a_i),$$
$$\therefore S_{i-1} \geq S_i.$$

In summary, if given s_i ($i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1$) such that $\mathbb{S}_i \in [0,1]$ or $\mathbb{S}_i \in [-1,0]$ (depending on whether $s_i \geq s_{i-1}$ or not), $i = \{1, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$, the scale transformation guarantees to generate an NVF fuzzy set.

Conversely, if two valid sets $A = (a_0, \ldots, a_{n-1})$ and $A' = (a'_0, \ldots, a'_{n-1})$ are given, which have the same RV, the scale rate of the bottom support, s_0, and the scale ratio of the ith support, $\mathbb{S}_i$ ($\mathbb{S}_i$, $i = \{1, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$) can be calculated by:

$$s_0 = \frac{a'_{n-1} - a'_0}{a_{n-1} - a_0} \tag{5.21}$$

$$\mathbb{S}_i = \begin{cases} \dfrac{\frac{a'_{n-i-1} - a'_i}{a'_{n-i} - a'_{i-1}} - \frac{a_{n-i-1} - a_i}{a_{n-i} - a_{i-1}}}{1 - \frac{a_{n-i-1} - a_i}{a_{n-i} - a_{i-1}}} \in [0,1] \\ (if\ \frac{a'_{n-i-1} - a'_i}{a_{n-i-1} - a_i} \geq \frac{a'_{n-i} - a'_{i-1}}{a_{n-i} - a_{i-1}} \geq 0) \\ \dfrac{\frac{a'_{n-i-1} - a'_i}{a'_{n-i} - a'_{i-1}} - \frac{a_{n-i-1} - a_i}{a_{n-i} - a_{i-1}}}{\frac{a_{n-i-1} - a_i}{a_{n-i} - a_{i-1}}} \in [-1,0] \\ (if\ \frac{a'_{n-i} - a'_{i-1}}{a_{n-i} - a_{i-1}} \geq \frac{a'_{n-i-1} - a'_i}{a_{n-i-1} - a_i} \geq 0) \end{cases} \tag{5.22}$$

Given that A and A' are both valid, the ranges of $\mathbb{S}_i$ as indicated above can be proved as follows.

proof 7 *When* $\frac{a'_{n-i-1} - a'_i}{a_{n-i-1} - a_i} \geq \frac{a'_{n-i} - a'_{i-1}}{a_{n-i} - a_{i-1}} \geq 0$*,*

$$\because 1 \geq \frac{a'_{n-i-1} - a'_i}{a'_{n-i} - a'_{i-1}} \geq \frac{a_{n-i-1} - a_i}{a_{n-i} - a_{i-1}} \geq 0,$$
$$\therefore 1 \geq \mathbb{S}_i \geq 0.$$

When $\frac{a'_{n-i}-a'_{i-1}}{a_{n-i}-a_{i-1}} \geq \frac{a'_{n-i-1}-a'_{i}}{a_{n-i-1}-a_{i}} \geq 0$,

$$\because 1 \geq \frac{a_{n-i-1}-a_i}{a_{n-i}-a_{i-1}} \geq \frac{a'_{n-i-1}-a'_i}{a'_{n-i}-a'_{i-1}} \geq 0,$$
$$\therefore 0 \geq \mathbb{S}_i \geq -1.$$

5.2.3 Move Transformation for General RV Definition

Now, consider the move transformation (as shown in Fig. 5.7) applied to an arbitrary polygonal fuzzy membership function $A = (a_0, \ldots, a_{n-1})$ to generate $A' = (a'_0, \ldots, a'_{n-1})$ such that they have the same representative value and the same lengths of supports, and $a'_i = a_i + l_i$, $i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}$. In order to achieve this, the move transformation is

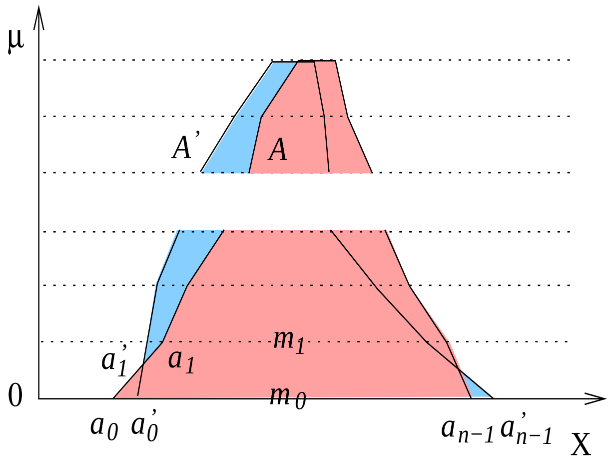

Figure 5.7: Move transformation

decomposed into $(\lceil \frac{n}{2} \rceil - 1)$ sub-moves. The ith sub-move ($i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}$) moves the ith support (from the bottom to the top beginning with 0) to the desired place. It moves all the characteristic points on and above the ith support, whilst keeping unaltered for those points under this support. To measure the degree of the ith sub-move, the first maximal possible move distance (in the sense that the sub-move doesn't lead to the above invalidity) should be computed first. To simplify the description of the sub-move procedure, only the right direction move (from a_i's point of view) is considered in the discussion hereafter. The left direction simply mirrors this operation.

If the ith point is supposed to move to the right direction, the maximal position $a_i^{(i)*}$ can be calculated as follows when $\sum_{j=i}^{\lceil \frac{n}{2} \rceil -1}(w_j + w_{n-1-j}) > 0$:

$$a_i^{(i)*} = \frac{\sum_{j=i}^{\lceil \frac{n}{2} \rceil -1} a_j(w_j + w_{n-1-j}) - A}{\sum_{j=i}^{\lceil \frac{n}{2} \rceil -1}(w_j + w_{n-1-j})} \tag{5.23}$$

where $A = \sum_{\substack{w_k<0 \\ i<k<\lceil \frac{n}{2} \rceil}} [(S_{k-1} - S_k) \sum_{m=k}^{\lceil \frac{n}{2} \rceil -1}(w_m + w_{n-1-m})]$ and S_k is the length of the kth support (either before or after move transformation as they are the same). If however $\sum_{j=i}^{\lceil \frac{n}{2} \rceil -1}(w_j + w_{n-1-j}) < 0$, the maximal position $a_i^{(i)*}$ is calculated similarly to (5.23) except that the condition $w_k < 0$ in term A is changed to $w_k > 0$. The calculation of (5.23) can be shown as follows.

proof 8 *As the sub-move doesn't change the RV and supports' lengths, according to (5.9), assume that*

$$\sum_{i=0}^{\lceil \frac{n}{2} \rceil -1} a_i'(w_i + w_{n-1-i}) = \sum_{i=0}^{\lceil \frac{n}{2} \rceil -1} a_i(w_i + w_{n-1-i}) = D$$

In addition, as the ith sub-move doesn't move the points under the ith support, it can therefore be assumed that

$$\sum_{j=i}^{\lceil \frac{n}{2} \rceil -1} a_j'(w_j + w_{n-1-j}) = \sum_{j=i}^{\lceil \frac{n}{2} \rceil -1} a_j(w_j + w_{n-1-j}) = E$$

Considering move point $a_i^{(i-1)}$ (a_i's new position after the $(i-1)$th sub-move) to the *right direction and* $\sum_{j=i}^{\lceil\frac{n}{2}\rceil-1}(w_j+w_{n-1-j})>0$,

$$
\begin{aligned}
a'_i(w_i+w_{n-1-i}) &= E-\sum_{j=i+1}^{\lceil\frac{n}{2}\rceil-1}a'_j(w_j+w_{n-1-j})\\
&\leq \begin{cases}
E-\sum_{j=i+1}^{\lceil\frac{n}{2}\rceil-2}a'_j(w_j+w_{n-1-j})\\
-a'_{\lceil\frac{n}{2}\rceil-2}(w_{\lceil\frac{n}{2}\rceil-1}+w_{n-\lceil\frac{n}{2}\rceil})\\
(if\ w_{\lceil\frac{n}{2}\rceil-1}+w_{n-\lceil\frac{n}{2}\rceil}>0)\\
E-\sum_{j=i+1}^{\lceil\frac{n}{2}\rceil-2}a'_i(w_i+w_{n-1-i})\\
-a'_{\lceil\frac{n}{2}\rceil-2}(w_{\lceil\frac{n}{2}\rceil-1}+w_{n-\lceil\frac{n}{2}\rceil})\\
-(S_{\lceil\frac{n}{2}\rceil-2}-S_{\lceil\frac{n}{2}\rceil-1})(w_{\lceil\frac{n}{2}\rceil-1}+w_{n-\lceil\frac{n}{2}\rceil})\\
(if\ w_{\lceil\frac{n}{2}\rceil-1}+w_{n-\lceil\frac{n}{2}\rceil}<0)
\end{cases}
\end{aligned}
$$

where $S_{\lceil\frac{n}{2}\rceil-2}$ *and* $S_{\lceil\frac{n}{2}\rceil-1}$ *are the lengths of the* $(\lceil\frac{n}{2}\rceil-2)$*th and* $(\lceil\frac{n}{2}\rceil-1)$*th supports, respectively. That is to say, if* $w_{\lceil\frac{n}{2}\rceil-1}+w_{n-\lceil\frac{n}{2}\rceil}>0$*, in order to get the maximal value of* $a'_i(w_i+w_{n-1-i})$*,* $a'_{\lceil\frac{n}{2}\rceil-1}$ *is assigned the same value as that of* $a'_{\lceil\frac{n}{2}\rceil-2}$*. This leads to the top left slope being vertical. Similarly, if however* $w_{\lceil\frac{n}{2}\rceil-1}+w_{n-\lceil\frac{n}{2}\rceil}<0$*,* $a'_{\lceil\frac{n}{2}\rceil-1}=a'_{\lceil\frac{n}{2}\rceil-2}+S_{\lceil\frac{n}{2}\rceil-2}-S_{\lceil\frac{n}{2}\rceil-1}$ *and it thus results in the top right slope being vertical. Repeating this procedure from the top down to the ith support leads that*

$$
\begin{aligned}
a'_i(w_i+w_{n-1-i}) &\leq E-a'_i\sum_{j=i+1}^{\lceil\frac{n}{2}\rceil-1}(w_j+w_{n-1-j})\\
&-\sum_{\substack{w_k<0\\ i<k<\lceil\frac{n}{2}\rceil}}[(S_{k-1}-S_k)\sum_{m=k}^{\lceil\frac{n}{2}\rceil-1}(w_m+w_{n-1-m})],
\end{aligned}
$$

which can therefore be rearranged to (5.23). The proof for the case with $\sum_{j=i}^{\lceil\frac{n}{2}\rceil-1}(w_j+w_{n-1-j})<0$ *is omitted as it simply follows. Note that it is meaningless for* $\sum_{j=i}^{\lceil\frac{n}{2}\rceil-1}(w_j+w_{n-1-j})=0$*. With such a weight vector, the RV cannot represent the overall location of a given fuzzy set. This is because the RV of a fuzzy set always keeps the same when the fuzzy set is merely moved without changing the geometrical shape.*

From the proof, the other extreme points $a_j^{(i)*}$ $(j=\{i+1,\ldots,\lceil\frac{n}{2}\rceil-1\})$ which are on the left side of the fuzzy set in the ith sub-move can be calculated by:

$$a_j^{(i)*} = \begin{cases} a_{j-1}^{(i)*} \;\; if \; w_j + w_{n-1-j} > 0 \\ a_{j-1}^{(i)*} + S_{j-1} - S_j \;\; if \; w_j + w_{n-1-j} < 0 \end{cases} \tag{5.24}$$

It can be proved that all the extreme points form an NVF fuzzy set $A^{(i)*}$ (as shown in Fig. 5.8) which must have at least a vertical slope between any two consecutive α-cuts above the ith support. This fuzzy set has the same RV as $A^{(i-1)}$. That is:

$$\sum_{j=0}^{\lceil \frac{n}{2} \rceil - 1} a_j^{(i)*}(w_j + w_{n-1-j}) = \sum_{j=0}^{\lceil \frac{n}{2} \rceil - 1} a_j^{(i-1)}(w_j + w_{n-1-j}) \tag{5.25}$$

The proof is ignored here as it is obvious from the calculation of the extreme point $a_i^{(i)*}$.

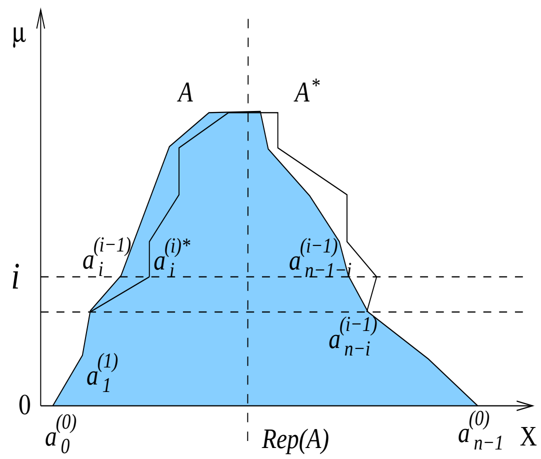

Figure 5.8: The extreme move positions in the ith sub-move

The move to the left direction from the viewpoint of a_i is omitted as it mirrors the right direction move.

From above, the first maximal move distance can be calculated. However, the ith sub-move not only needs to consider the possible above invalidity, but also needs to pay attention to the possible below invalidity. Otherwise it may still lead to invalidity as shown in Fig. 5.8. To avoid this, the second maximal move distance is calculated as $a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}$. It is intuitive to pick the minimal of these two maximal move distances as the maximal move distance which doesn't lead to either above or below invalidity. The move ratio $\mathbb{M}_i$, which is used to measure the degree of such a sub-move, is thus calculated by:

$$\mathbb{M}_i = \begin{cases} \frac{l_i-(a_i^{(i-1)}-a_i)}{min\{a_i^{(i)*}-a_i^{(i-1)},a_{n-i}^{(i-1)}-a_{n-1-i}^{(i-1)}\}} \\ (if\ l_i \geq (a_i^{(i-1)}-a_i)) \\ \\ \frac{l_i-(a_i^{(i-1)}-a_i)}{min\{a_i^{(i-1)}-a_i^{(i)*},a_i^{(i-1)}-a_{i-1}^{(i-1)}\}} \\ (if\ l_i \leq (a_i^{(i-1)}-a_i)) \end{cases} \tag{5.26}$$

where the notation $a_i^{(i-1)}$ represents a_i's new position after the $(i-1)$th sub-move. Initially, $a_i^{(-1)} = a_i$. If $\mathbb{M}_i \in [0,1]$ when $l_i \geq (a_i^{(i-1)} - a_i)$, or $\mathbb{M}_i \in [-1,0]$ when $l_i \leq (a_i^{(i-1)} - a_i)$, the sub-move is carried out as follows: the characteristic points under the ith support are not changed:

$$a_j^{(i)} = a_j^{(i-1)}, \quad j = \{0,\ldots,i-1,n-i,\ldots,n-1\}$$

while the other points $a_i, a_{i+1}, \ldots, a_{n-1-i}$ are being moved. Initially, when $i = 0$, all characteristic points are being moved of course. If moving to the right direction from the viewpoint of $a_i^{(i-1)}$, i.e., $\mathbb{M}_i \in [0,1]$, the moving distances of a_j ($j = \{i, i+1, \ldots, \lceil \frac{n}{2} \rceil - 1\}$) which are on the left side of fuzzy set are calculated by multiplying $\mathbb{M}'_i$ with the distances between the extreme positions $a_j^{(i)*}$ and themselves. In so doing, $a_j^{(i-1)}$ will move the same proportion of distances to their respective extreme positions. $a_j^{(i)}$ can thus be computed by:

$$a_j^{(i)} = a_j^{(i-1)} + \mathbb{M}'_i(a_j^{(i)*} - a_j^{(i-1)}), \tag{5.27}$$

where

$$\mathbb{M}'_i = \mathbb{M}_i \frac{min\{a_i^{(i)*} - a_i^{(i-1)}, a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}\}}{a_i^{(i)*} - a_i^{(i-1)}}. \tag{5.28}$$

$\mathbb{M}'_i$ is the *applied move ratio* for the ith sub-move. If $\mathbb{M}_i \in [0,1]$, $\mathbb{M}'_i \in [0, \mathbb{M}_i]$. The adoption of applied move ratio $\mathbb{M}'_i$ avoids the possible below invalidity. Such a move strategy leads to an NVF set $A^{(i)} = \{a_0^{(i)}, \ldots, a_{n-1}^{(i)}\}$ which has the same representative value as A and has the new point $a_i^{(i)}$ on the desired position, i.e., $Rep(A^{(i)}) = Rep(A)$ and $a_i^{(i)} = a_i + l_i$. All these properties can be proved as follows.

proof 9 *Considering the ith point during the ith sub-move ($i = \{0,\ldots,\lceil\frac{n}{2}\rceil-2\}$), substituting (5.26) and (5.28) to (5.27) leads to $a_i^{(i)} = a_i + l_i$, which is the desired position for a_i to be moved to. As the ith support length is fixed, a_{n-1-i} is also moved to the desired position via this sub-move. Initially, the 0th sub-move moves a_0 and a_{n-1} to the correct positions, and the first sub-move moves a_1 and a_{n-2} to the correct positions while keeping a_0 and a_{n-1} unchanged. Following this by induction, the ith sub-move moves $a_0,\ldots,a_i,a_{n-1-i},\ldots,a_{n-1}$ to the correct positions.*

The distances between $a_{j+1}^{(i)}$ and $a_j^{(i)}$ ($j = \{i,i+1,\ldots,\lceil\frac{n}{2}\rceil-2\}$) are calculated as follows according to (5.27):

$$a_{j+1}^{(i)} - a_j^{(i)} = (a_{j+1}^{(i-1)} - a_j^{(i-1)})(1-\mathbb{M}'_i) + \mathbb{M}'_i(a_{j+1}^{(i)*} - a_j^{(i)*}).$$

Initially, when $i = 0$, $a_{j+1}^{(i-1)} - a_j^{(i-1)} = a_{j+1}^{(-1)} - a_j^{(-1)} = a_{j+1} - a_j \geq 0$ and $a_{j+1}^{(i)*} - a_j^{(i)*} = a_{j+1}^{(0)*} - a_j^{(0)*} \geq 0$ ($j = \{0,1,\ldots,\lceil\frac{n}{2}\rceil-2\}$) as A and $A^{(0)*}$ are valid. This leads to $a_{j+1}^{(0)} - a_j^{(0)} \geq 0$, $j = \{0,1,\ldots,\lceil\frac{n}{2}\rceil-2\}$, which in turn leads to $a_{j+1}^{(1)} - a_j^{(1)} \geq 0$, $j = \{1,\ldots,\lceil\frac{n}{2}\rceil-2\}$. Also, as this sub-move causes moves to the right direction, $a_1^{(1)} \geq a_0^{(0)} = a_0^{(1)}$. So $a_{j+1}^{(1)} - a_j^{(1)} \geq 0$, $j = \{0,\ldots,\lceil\frac{n}{2}\rceil-2\}$. By induction, it follows that

$$a_{j+1}^{i} - a_j^{i} \geq 0, \quad j = \{0,\ldots,\lceil\frac{n}{2}\rceil-2\ \}.$$

The new positions of a_j ($j = \{n-\lceil\frac{n}{2}\rceil,\ldots,n-1-i\}$) which are on the right side of A can be calculated similarly:

$$a_j^{(i)} = a_j^{(i-1)} + \mathbb{M}'_i(a_{n-1-j}^{(i)*} - a_{n-1-j}^{(i-1)}). \tag{5.29}$$

Thus, the distances between $a_{j+1}^{(i)}$ and $a_j^{(i)}$ ($j = \{n-\lceil\frac{n}{2}\rceil,\ldots,n-2-i\}$) are calculated by:

$$\begin{aligned} &a_{j+1}^{(i)} - a_j^{(i)} = a_{j+1}^{(i-1)} - a_j^{(i-1)} \\ &+\mathbb{M}'_i(a_{n-2-j}^{(i)*} - a_{n-2-j}^{(i-1)} - a_{n-1-j}^{(i)*} + a_{n-1-j}^{(i-1)}). \end{aligned}$$

From (5.24),

$$a_{n-1-j}^{(i)*} = \begin{cases} a_{n-2-j}^{(i)*} \quad (if\ w_{n-1-j} + w_j > 0) \\ a_{n-2-j}^{(i)*} + S_{n-2-j} - S_{n-1-j} \\ (if\ w_{n-1-j} + w_j < 0) \end{cases}$$

$$\therefore a^{(i)*}_{n-2-j} - a^{(i)*}_{n-1-j} \geq S_{n-1-j} - S_{n-2-j}$$
$$\therefore a^{(i)}_{j+1} - a^{(i)}_{j} \geq a^{(i-1)}_{j+1} - a^{(i-1)}_{j} + \mathbb{M}'_i(a^{(i-1)}_{j} - a^{(i-1)}_{j+1})$$
$$= (a^{(i-1)}_{j+1} - a^{(i-1)}_{j})(1-\mathbb{M}'_i) \geq 0$$

Initially, $(a^{(i-1)}_{j+1} - a^{(i-1)}_{j})(1-\mathbb{M}'_i) = (a^{(-1)}_{j+1} - a^{(-1)}_{j})(1-\mathbb{M}'_i) = (a_{j+1}-a_j)(1-\mathbb{M}'_i) \geq 0$ $(j = \{n-\lceil \frac{n}{2} \rceil, \ldots, n-2\})$*. This leads to* $a^{(0)}_{j+1} - a^{(0)}_{j} \geq 0$ $(j = \{n-\lceil \frac{n}{2} \rceil, \ldots, n-2\})$*, which in turn leads to* $a^{(1)}_{j+1} - a^{(1)}_{j} \geq 0$ $(j = \{n-\lceil \frac{n}{2} \rceil, \ldots, n-3\})$*. Also, the adoption of applied move ratio ensures* $a^{(1)}_{n-1} = a^{(0)}_{n-1} \geq a^{(1)}_{n-2}$*, so* $a^{(1)}_{j+1} - a^{(1)}_{j} \geq 0$ $(j = \{n-\lceil \frac{n}{2} \rceil, \ldots, n-2\})$*. Again, by induction,*

$$a^{(i)}_{j+1} - a^{(i)}_{j} \geq 0 \quad j = \{n - \lceil \frac{n}{2} \rceil, \ldots, n-2\}.$$

Also, as $a^{(i)}_{n-\lceil \frac{n}{2} \rceil} - a^{(i)}_{\lceil \frac{n}{2} \rceil - 1} = S_{\lceil \frac{n}{2} \rceil - 1} \geq 0$*. Thus, it can be summarised that*

$$a^{(i)}_{j+1} - a^{(i)}_{j} \geq 0 \quad j = \{0, \ldots, n-2\},$$

i.e., $A^{(i)}$ *is an NVF set.*

The representative value of A after the ith sub-move, $Rep(A^{(i)})$*, is the same as its original* $Rep(A)$*. This is because the following holds according to (5.27), (5.29) and (5.25):*

$$\sum_{j=0}^{\lceil \frac{n}{2} \rceil - 1} a^{(i)}_{j}(w_j + w_{n-1-j})$$
$$= \sum_{j=0}^{\lceil \frac{n}{2} \rceil - 1} a^{(i-1)}_{j}(w_j + w_{n-1-j})$$
$$= \ldots$$
$$= \sum_{j=0}^{\lceil \frac{n}{2} \rceil - 1} a_{j}(w_j + w_{n-1-j})$$

The proofs of the properties including moving to the desired position, preservation of RV and validity for moving to the left direction (i.e., $\mathbb{M}_i \in [-1, 0]$*) are omitted as they mirror the derivations as given above.*

In summary, if given move ratios $\mathbb{M}_i \in [-1, 1]$, $(i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\})$, the $(\lceil \frac{n}{2} \rceil - 1)$ sub-moves transform the given NVF set $A = (a_0, \ldots, a_{n-1})$ to a new NVF set $A' = (a'_0, \ldots, a'_{n-1})$ with the same lengths of supports and the same RV.

In the converse case, where two valid fuzzy sets $A = (a_0, \ldots, a_{n-1})$ and $A' = (a'_0, \ldots, a'_{n-1})$ are given, which have the same representative value, the move ratios $\mathbb{M}_i$, $i = \{0, 1, \ldots, \lceil \frac{n}{2} \rceil - 2\}$, are computed by:

$$\mathbb{M}_i = \begin{cases} \frac{a'_i - a_i^{(i-1)}}{min\{a_i^{(i)*} - a_i^{(i-1)}, a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}\}} \\ (if\ a'_i \geq a_i^{(i-1)}) \\ \\ \frac{a'_i - a_i^{(i-1)}}{min\{a_i^{(i-1)} - a_i^{(i)*}, a_i^{(i-1)} - a_{i-1}^{(i-1)}\}} \\ (if\ a'_i \leq a_i^{(i-1)}) \end{cases} \tag{5.30}$$

where $a_i^{(i-1)}$ is the a_i's new position after the $(i-1)$th sub-move. Initially, when $i = 0$, $a_i^{(-1)} = a_i$. This sub-move (bottom sub-move) will not lead to below invalidity as there are no characteristic points underneath, whilst the other sub-moves need to consider situations where invalidity arises both above and underneath. Initially, when $i = 0$, $a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}$ and $a_i^{(i-1)} - a_{i-1}^{(i-1)}$ are not defined. In order to keep integrity of (5.30), both of them take on value 1 to present the bottom case.

Given that $A = (a_0, \ldots, a_{n-1})$ and $A' = (a'_0, \ldots, a'_{n-1})$ are both valid, the ranges of $\mathbb{M}_i$ (i.e., $\mathbb{M}_i \in [0, 1]$ when $a'_i \geq a_i^{(i-1)}$ or $\mathbb{M}_i \in [-1, 0]$ when $a'_i \leq a_i^{(i-1)}$) are obvious and hence no proof is needed.

Moreover, the present work is readily extendable to rules involving variables that are represented by Gaussian and other bell-shaped membership functions. For instance, consider the simplest case where two rules $A_1 \Rightarrow B_1$, $A_2 \Rightarrow B_2$ and the observation A^* all involve the use of Gaussian fuzzy sets of the form (Fig. 5.9):

$$p(x) = e^{\frac{-(x-c)^2}{2\sigma^2}}, \tag{5.31}$$

where c and σ are the mean and standard deviation respectively. The construction of the intermediate rule is slightly different from the polygonal fuzzy membership function cases in the sense that the standard deviations are used to interpolate. Since the Gaussian shape is symmetrical, c is chosen to be the representative value of such a fuzzy set. In so doing, the antecedent value A' of the intermediate rule has the same representative value as that of observation A^*. That means only scale transformation from A' to A^* as depicted in Fig. 5.9 is needed to carry out interpolation. Heuristics can be employed to represent the scale rate s in terms of the standard deviation σ. One of the simplest definitions is to calculate the ratio of two fuzzy sets' σ values when considering transformation from one to the other. The scale rate s can therefore be written as:

$$s = \frac{\sigma_{A^*}}{\sigma_{A'}}. \tag{5.32}$$

The transformations involving other bell-shaped membership functions follows this idea analogously.

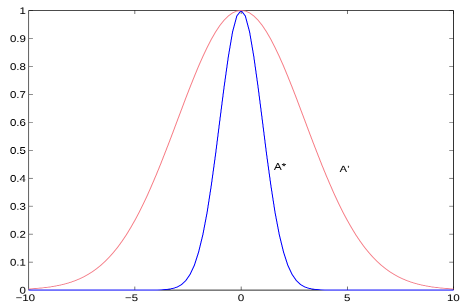

Figure 5.9: Gaussian scale transformation

5.2.4 Algorithm Outline

As indicated earlier, it is intuitive to maintain the similarity degree between the consequent parts $B' = (b'_0, \ldots, b'_{n-1})$ and $B^* = (b^*_0, \ldots, b^*_{n-1})$ to be the same as that between the antecedent parts $A' = (a'_0, \ldots, a'_{n-1})$ and $A^* = (a^*_0, \ldots, a^*_{n-1})$, in performing interpolative reasoning. The proposed scale and move transformations can be used to entail this by the following algorithm:

1. Calculate scale rates s_i ($i = \{0, 1 \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$) of the ith support from A' to A^* by

$$s_i = \frac{a^*_{n-1-i} - a^*_i}{a'_{n-1-i} - a'_i}. \tag{5.33}$$

2. Calculate scale rate s_0 of the bottom support (or just get from the first step) and scale ratios $\mathbb{S}_i$ ($i = \{1 \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$) of the ith support from A' to A^* by (5.21) and (5.22).

3. Apply scale transformation to A' with scale rates s_i calculated in the first step to obtain A''.

4. Assign scale rate s'_0 of the bottom support of B' to the value of s_0 (i.e., $s'_0 = s_0$), with the scale ratios $\mathbb{S}'_i$, ($i = \{1 \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$) of the ith support of B' calculated as per (5.22) under the condition that they are equal to $\mathbb{S}_i$ ($i = \{1 \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$) as calculated in step 2:

$$s'_i = \begin{cases} s_i & (if\ i = 0) \\ \dfrac{s'_{i-1}(s_i - s_{i-1})(\frac{b'_{n-i} - b'_{i-1}}{b'_{n-i-1} - b'_i} - 1)}{s_{i-1}(\frac{a'_{n-i} - a'_{i-1}}{a'_{n-i-1} - a'_i} - 1)} + s'_{i-1} & (if\ s_i \geq s_{i-1} \geq 0) \\ \dfrac{s'_{i-1} s_i}{s_{i-1}} & (if\ s_{i-1} \geq s_i \geq 0) \end{cases} \tag{5.34}$$

5. Apply scale transformation to B' using s'_i ($i = \{0, 1 \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$) as calculated in step 4 to obtain $B'' = (b''_0, \ldots, b''_{n-1})$.

6. Decompose the move transformation to $(\lceil \frac{n}{2} \rceil - 1)$ sub-moves. For $i = 0, 1, \ldots, \lceil \frac{n}{2} \rceil - 2$,

   (a) Calculate the ith sub-move ratio $\mathbb{M}_i$ from $A^{(i-1)}$ to A^* by (5.30), where $A^{(i-1)}$ is the fuzzy set obtained after the $(i-1)$th sub-move with initialization $A^{(-1)} = A''$.

   (b) Apply move transformation to $A^{(i-1)}$ using $\mathbb{M}_i$ to obtain $A^{(i)} = \{a_0^{(i)}, a_1^{(i)}, \ldots, a_n^{(i)}\}$.

   (c) Apply move transformation to $B^{(i-1)}$ using $\mathbb{M}_i$ to obtain $B^{(i)} = \{b_0^{(i)}, b_1^{(i)}, \ldots, b_n^{(i)}\}$.

7. Return $A^{(\lceil \frac{n}{2} \rceil - 2)} = A^*$ and $B^{(\lceil \frac{n}{2} \rceil - 2)}$, which is the required resultant fuzzy set B^*, once the *for* loop of step 6 terminates.

Clearly, B' and B^* will then retain the same similarity degree as that between the antecedent parts A' and A^*.

There are two specific cases worth noting when applying the scale transformation. The first is that if A^* is a singleton while A' is a regular normal and valid fuzzy set, the scale transformation from A' to A^* is 0. This case can be easily handled by setting the result B^* to a singleton whose value interpolates between $Rep(B_1)$ and $Rep(B_2)$ in the same way as A^* does between $Rep(A_1)$ and $Rep(A_2)$. The second case (which only exists if both antecedents A_1 and A_2 are singletons) is that if A^* is a regular normal and valid fuzzy set while A' is a singleton, the scale transformation from A' to A^* will be infinite. Since infinity cannot be used to generate the resulting fuzzy set, a modified strategy is created for this. The ratio between the individual support length of fuzzy set A^* and the distance of $Rep(A_1)$ and $Rep(A_2)$ is calculated in order to compute the corresponding support length of fuzzy set B^* by equalizing the corresponding ratio. Note that the fuzzy set obtained by the scale transformation from a singleton is an isosceles polygonal one.

5.3 Further Development of Transformation Based Interpolation

The proposed scale and move transformations help generate unique, valid and normal fuzzy results, making the interpolation inference possible for real life sparse rule bases. However, a disadvantage of the previously proposed method is that the computation complexity increases more quickly than the increasing of the point size (see chapter 6 for details). In addition, the *piecewise linearity* is preferred to generate piecewise linear results from the given piecewise linear rules and observations. Almost all existing interpolation methods do not preserve piecewise linearity in general cases. Only a few (including the proposed one) retain this property in triangular cases. In this section, a further development is made to the previously proposed scale and move transformations, not only to reduce the computation efforts but also to maintain piecewise linearity in arbitrary polygonal cases. Note that this development does not affect the definitions of RV and the construction of the intermediate rules. Attention is only drawn to the modification of scale and move transformations.

5.3.1 Enhanced Scale Transformations

This enhanced version of scale transformation has the same process as the one proposed in subsection 5.2.2. The only difference is the way of calculating scale rates. For completeness, the description is partially repeated.

Consider applying scale transformation to an arbitrary polygonal fuzzy membership function $A = (a_0, \ldots, a_{n-1})$ (as shown in Fig. 5.10) to generate $A' = (a'_0, \ldots, a'_{n-1})$ such that they have the same RV, and $a'_{n-1-i} - a'_i = s_i(a_{n-1-i} - a_i)$, where s_i are scale rates and $i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$. In order to achieve this, $\lfloor \frac{n}{2} \rfloor$ equations $a'_{n-1-i} - a'_i =$

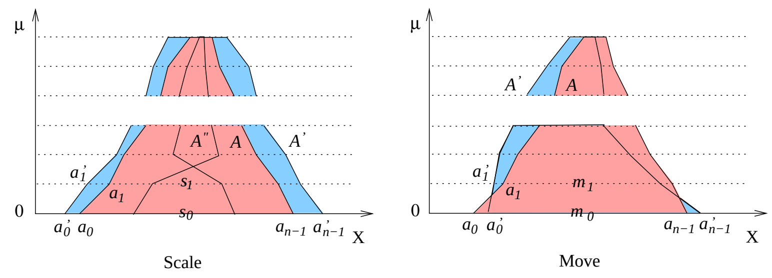

Figure 5.10: Enhanced scale and move transformations

$s_i(a_{n-1-i} - a_i)$, $i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$, are imposed to obtain the supports with desired lengths, and $(\lceil \frac{n}{2} \rceil - 1)$ equations $\frac{a'_{i+1} - a'_i}{a'_{n-1-i} - a'_{n-2-i}} = \frac{a_{i+1} - a_i}{a_{n-1-i} - a_{n-2-i}}$, $i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}$ are imposed to equalise the ratios between the left $(\lceil \frac{n}{2} \rceil - 1)$ slopes' lengths and the right $(\lceil \frac{n}{2} \rceil - 1)$ slopes' lengths of A' to those counterparts of the original fuzzy set A. The equation $\sum_{i=0}^{n-1} w_i a'_i = \sum_{i=0}^{n-1} w_i a_i$ which ensures the same representative values before and after the transformation is added to make up of $\lfloor \frac{n}{2} \rfloor + (\lceil \frac{n}{2} \rceil - 1) + 1 = n$ equations. All these n equations are collectively written as:

$$\begin{cases} a'_{n-1-i} - a'_i = s_i(a_{n-1-i} - a_i) = S_i \\ (i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}) \\ \frac{a'_{i+1} - a'_i}{a'_{n-1-i} - a'_{n-2-i}} = \frac{a_{i+1} - a_i}{a_{n-1-i} - a_{n-2-i}} = R_i \\ (i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}) \\ \sum_{i=0}^{n-1} w_i a'_i = \sum_{i=0}^{n-1} w_i a_i \end{cases} \tag{5.35}$$

where S_i is the ith support length of the resultant fuzzy set and R_i is the ratio between the left ith slope length and the right ith slope length. Solving these n equations simultaneously results in an unique and valid fuzzy set A' given that the resultant set has a descending order of the support lengths from the bottom to the top.

So far the enhanced scale transformation remains the same as the original one. The difference is in the way of calculating scale rates. Recall that the scale ratios

$\mathbb{S}$ are introduced in the original scale transformations, to ensure the support lengths decreased from the bottom support to the top support. Instead, *left scale criterion* $\mathbb{SL}_i$ and *right scale criterion* $\mathbb{SR}_i$ are introduced for the ith support, $i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}$.

$$\mathbb{SL}_i = \frac{a'_{i+1} - a'_i}{a_{i+1} - a_i}, \tag{5.36}$$

$$\mathbb{SR}_i = \frac{a'_{n-1-i} - a'_{n-2-i}}{a_{n-1-i} - a_{n-2-i}}. \tag{5.37}$$

Obviously, $\mathbb{SL}_i \geq 0$ and $\mathbb{SR}_i \geq 0$ if both A and A' are valid. Having introduced these, the scale rate of the ith support is computed:

$$\begin{aligned}
s_i = \frac{S'_i}{S_i} &= \frac{a'_{n-1-i} - a'_i}{a_{n-1-i} - a_i} \\
&= \frac{\mathbb{SL}_i(a_{i+1} - a_i) + a'_{n-2-i} - a'_{i+1} + \mathbb{SR}_i(a_{n-1-i} - a_{n-2-i})}{a_{n-1-i} - a_i} \\
&= \frac{\mathbb{SL}_i(a_{i+1} - a_i) + s_{i+1}(a_{n-2-i} - a_{i+1}) + \mathbb{SR}_i(a_{n-1-i} - a_{n-2-i})}{a_{n-1-i} - a_i},
\end{aligned} \tag{5.38}$$

where S'_i and S_i are the lengths of the ith support of A' and A respectively. As $S'_i = S'_{i+1} + \mathbb{SL}_i(a_{i+1} - a_i) + \mathbb{SR}_i(a_{n-1-i} - a_{n-2-i})$, if $\mathbb{SL}_i \geq 0$ and $\mathbb{SR}_i \geq 0$, then $\mathbb{SL}_i(a_{i+1} - a_i) \geq 0$ and $\mathbb{SR}_i(a_{n-1-i} - a_{n-2-i}) \geq 0$, hence $S'_i \geq S'_{i+1}$ must hold. So the scale transformation guarantees generation of an NVF fuzzy set.

Conversely, if two valid sets $A = (a_0, \ldots, a_{n-1})$ and $A' = (a'_0, \ldots, a'_{n-1})$ which have the same RV are given, the left and right scale criterion of the ith support, $\mathbb{SL}_i$, $\mathbb{SR}_i$ ($i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}$) can be calculated by (5.36) and (5.37) respectively. Given that A and A' are both valid, $\mathbb{SL}_i \geq 0$ and $\mathbb{SR}_i \geq 0$ must hold.

Special treatments are needed if: 1) A has a vertical left slope on the ith support level, the term of $(a_{i+1} - a_i)$ in (5.36) is replaced by the vertical distance of the ith and $(i+1)$th points to avoid division by zero; and 2) A has a vertical right slope on the ith support level, the term of $(a_{n-1-i} - a_{n-2-i})$ in (5.37) is replaced by the vertical distance of the ith and $(i+1)$th points.

The above scale criteria are calculated from top to bottom (so are the scale rates). If on the contrary, the calculation order is from bottom to top, then it would be possible that the scaled fuzzy set becomes invalid, as A'' illustrated in Fig. 5.10.

5.3.2 Enhanced Move Transformations

The enhanced move transformation is no longer like the original proposed one. Instead, it appears rather like the scale transformation, which is the reason that the computation complexity is significantly reduced from $O(n^2)$ to $O(n)$ (n is the size of characteristic points, see chapter 6 for details).

After performing the scale transformation, the lengths of supports of a fuzzy set become equal to those of the desired fuzzy set. Now the move transformation is used to move the supports to appropriate positions. Consider applying move transformation to an arbitrary polygonal fuzzy membership function $A = (a_0, \ldots, a_{n-1})$ (as shown in Fig. 5.10) to generate $A' = (a'_0, \ldots, a'_{n-1})$ such that they have the same RV and the same lengths of supports. In order to achieve this, $\lfloor \frac{n}{2} \rfloor$ equations $a'_{n-1-i} - a'_i = a_{n-1-i} - a_i$, $i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$, are imposed to ensure the same lengths of supports, and $(\lceil \frac{n}{2} \rceil - 1)$ equations $\frac{a'_{i+1} - a'_i}{a'_{n-1-i} - a'_{n-2-i}} / \frac{a_{i+1} - a_i}{a_{n-1-i} - a_{n-2-i}} = \mathbb{RC}_i$, where $i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}$ and $\mathbb{RC}_i$ are the *move criterion*, are imposed to set the ratios between the ith left slope length and the ith right slope length of A', to their counterparts of the original fuzzy set A. The equation $\sum_{i=0}^{n-1} w_i a'_i = \sum_{i=0}^{n-1} w_i a_i$ which ensures the same representative values before and after the transformation is added to make up of $\lfloor \frac{n}{2} \rfloor + (\lceil \frac{n}{2} \rceil - 1) + 1 = n$ equations. All these n equations are collectively written as:

$$\begin{cases} a'_{n-1-i} - a'_i = a_{n-1-i} - a_i = S_i \\ (i = \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}) \\ \frac{a'_{i+1} - a'_i}{a'_{n-1-i} - a'_{n-2-i}} / \frac{a_{i+1} - a_i}{a_{n-1-i} - a_{n-2-i}} = \mathbb{RC}_i \\ (i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}) \\ \sum_{i=0}^{n-1} w_i a'_i = \sum_{i=0}^{n-1} w_i a_i \end{cases} \tag{5.39}$$

where S_i is the ith support length of the fuzzy set (either before or after moving) and $\mathbb{RC}_i$ is the move criterion for ith support. If $\mathbb{RC}_i \geq 0$, solving these n equations simultaneously results in a unique and valid fuzzy set.

Conversely, if two valid sets $A = (a_0, \ldots, a_{n-1})$ and $A' = (a'_0, \ldots, a'_{n-1})$ are given, which have the same RV and the same lengths of supports, the move criterion of the ith support, $\mathbb{RC}_i$ ($i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}$) can be calculated by (5.39). Given that A and A' are both valid, $\mathbb{RC}_i \geq 0$ must hold.

Unlike the scale transformation, the move transformation does not have to follow a fixed order for calculation. In particular, the calculation for all α-cut levels is carried out simultaneously. However, there are special cases which need extra consideration in calculating the move criterion: 1) If A' has a vertical right slope on the ith support level, the move criterion is set to -1 in the implementation. When any fuzzy sets are moved using such a move criterion, they become fuzzy sets with vertical right slopes on the ith support level. 2) If the original fuzzy set A has a vertical left slope on the ith support level, the term $(a_{i+1} - a_i)$ will be replaced by the vertical distance between the ith and $(i+1)$th points. 3) If A has a vertical right slope on the ith support level, the term $(a_{n-i-1} - a_{n-i-2})$ will be replaced by the vertical distance between the ith and $(i+1)$th points. These are needed to avoid division by zero.

5.3.3 Algorithm Outline

Now the proposed scale and move transformations allow the similarity degree between two fuzzy sets to be measured by the *scale criterion* and *move criterion*, the desired conclusion B^* can be obtained as follows:

1. Calculate scale rates s_i ($i = \{0, 1 \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$) of the ith support from A' to A^* according to $s_i = \frac{a^*_{n-1-i} - a^*_i}{a'_{n-1-i} - a'_i}$.
2. Apply scale transformation to A' using scale rates s_i ($i = \{0, 1 \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$) computed above to obtain A'', by simultaneously solving n linear equations as shown in (5.35).
3. Calculate left and right scale criterion $\mathbb{SL}_i$, $\mathbb{SR}_i$, $i = \{0 \ldots, \lceil \frac{n}{2} \rceil - 2\}$), of the ith support from A' to A^* according to (5.36) and (5.37).
4. Calculate scale rates s'_i ($i = \{0, 1 \ldots, \lceil \frac{n}{2} \rceil - 2\}$) of the ith support from B' to B^* according to (5.38). Note that if B' has two points of membership value 1, $s'_{\lfloor \frac{n}{2} \rfloor - 1} = s_{\lfloor \frac{n}{2} \rfloor - 1}$.
5. Apply scale transformation to B' using s'_i ($i = \{0, 1 \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$) as calculated in step 4 to obtain $B'' = (b''_0, \ldots, b''_{n-1})$, by simultaneously solving the n linear equations as shown in (5.35).

6. Calculate move criterion $\mathbb{RC}_i$, $i = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\}$, on the ith support level from A'' to A^* according to (5.39).

7. Apply move transformation to B'' using the move criterion as calculated in step 6 to obtain B^*, by simultaneously solving the n linear equations as shown in (5.39).

Clearly, B' and B^* will retain the same similarity degree as that between the antecedent parts A' and A^*.

5.4 Multiple Antecedent Variables Interpolation

The one variable case described above concerns interpolation between two adjacent rules with each involving one antecedent variable. This is readily extendable to rules with multiple antecedent attributes. This section describes the multiple antecedent variables interpolation using the originally proposed scale and move transformations. The one using the enhanced transformations is ignored as it follows straightforwardly. Of course, the attributes appearing in both rules must be the same to make sense for interpolation.

Without losing generality, suppose that two adjacent rules R_i and R_j are represented by

$$if\ X_1\ is\ A_{1i}\ and \ldots and\ X_m\ is\ A_{mi}\ then\ Y\ is\ B_i,$$
$$if\ X_1\ is\ A_{1j}\ and \ldots and\ X_m\ is\ A_{mj}\ then\ Y\ is\ B_j.$$

Thus, when a vector of observations (A_1^*, ..., A_k^*, ..., A_m^*) is given, by direct analogy to one variable case, the values A_{ki} and A_{kj} of X_k, $k = 1, 2, \ldots, m$, are used to obtain a new NVF set A_k':

$$A_k' = (1 - \lambda_k)A_{ki} + \lambda_k A_{kj}, \tag{5.40}$$

where

$$\lambda_k = \frac{d(Rep(A_{ki}), Rep(A_k^*))}{d(Rep(A_{ki}), Rep(A_{kj}))}.$$

Clearly, the representative value of A_k' remains the same as that of the kth observation A_k^*.

The resulting A'_k and the given A^*_k are used to compute the integrated transformation

$$T(A'_k, A^*_k) = \{s_{k0}, \mathbb{S}_{k1}, \ldots, \mathbb{S}_{k(\lfloor \frac{n}{2} \rfloor - 1)}, \mathbb{M}_{k0}, \ldots, \mathbb{M}_{k(\lceil \frac{n}{2} \rceil - 2)}\}$$

just like the one variable case. From this, the combined scale rate s_c, scale ratios $\mathbb{S}_{ci}$, $(i = \{1, \ldots, \lfloor \frac{n}{2} \rfloor - 1\})$ and move ratios $\mathbb{M}_{cj}$ $(j = \{0, \ldots, \lceil \frac{n}{2} \rceil - 2\})$ over the m conditional attributes are respectively calculated as the arithmetic means of s_{k0}, $\mathbb{S}_{ki}$ and $\mathbb{M}_{kj}$, $k = 1, 2, \ldots, m$:

$$s_{c0} = \frac{1}{m} \sum_{k=1}^{m} s_{k0}, \tag{5.41}$$

$$\mathbb{S}_{ci} = \frac{1}{m} \sum_{k=1}^{m} \mathbb{S}_{ki}, \tag{5.42}$$

$$\mathbb{M}_{cj} = \frac{1}{m} \sum_{k=1}^{m} \mathbb{M}_{kj}. \tag{5.43}$$

Note that, other than using the arithmetic mean, different mechanisms such as the geometric mean may be employed for this purpose. These means help capture the intuition that when no particular information regarding which variable has a more dominating influence upon the conclusion, all the variables are treated equally. If such information is available, a weighted mean operator may be better to use.

Regarding the consequences, by analogy to expression (5.16), B' can be computed by

$$B' = (1 - \lambda_a) B_i + \lambda_a B_j. \tag{5.44}$$

Here, λ_a is deemed to be the average of λ_k, $k = 1, 2, \ldots, m$, to mirror the approach taken previously:

$$\lambda_a = \frac{1}{m} \sum_{k=1}^{m} \lambda_k. \tag{5.45}$$

As the integrated transformation

$$T = \{s_{c0}, \mathbb{S}_{c1}, \mathbb{S}_{c2}, \ldots, \mathbb{S}_{c(\lfloor \frac{n}{2} \rfloor - 1)}, \mathbb{M}_{c0}, \mathbb{M}_{c1}, \ldots, \mathbb{M}_{c(\lceil \frac{n}{2} \rceil - 2)}\}$$

reflects the similarity degree between the observation vector and the values of the given rules, the fuzzy set B^* of the conclusion can then be estimated by transforming B' via the application of the same T.

5.5 Case Studies

In this section, the example problems given in [HCL98, YMQ95] together with several new problem cases are used to illustrate the originally proposed and enhanced interpolation methods (denoted as OHS and EHS methods). The comparative studies to the work of [KH93a, KH93c] (denoted as KH, as stated before) and [HCL98] (denoted as HCL) are provided. All the results except example 7 discussed below concern the interpolation between two adjacent rules $A_1 \Rightarrow B_1$ and $A_2 \Rightarrow B_2$, while example 7 shows a case of interpolation between rules involving two antecedent variables.

Example 1. This example demonstrates the use of the proposed method involving only triangular fuzzy sets. The average RV is used in this example. All the conditions are shown in Table 5.1 and Fig. 5.11, which also include the results of interpolation. Suppose $A^* = (7,8,9)$. First, according to (5.14) and (5.16), $A'(5.30,8.85,9.85)$

Table 5.1: Results for example 1, with $A^* = (7,8,9)$

Attribute Values	Results	
$A_1 = (0,5,6)$	Method	B^*
$A_2 = (11,13,14)$	KH	(6.36, 5.38, 7.38)
$B_1 = (0,2,4)$	HCL	(6.36, 6.58, 7.38)
$B_2 = (10,11,13)$	OHS	(5.83, 6.26, 7.38)
	EHS	(5.54, 5.97, 7.97)

and $B'(4.81,6.33,8.33)$ are calculated by interpolation of A_1, A_2 and B_1, B_2, respectively, with $\lambda_{Rep} = 0.48$, which is calculated from (5.12). Then, the calculations are varied with respect to original and enhanced HS methods. For the former, the scale rate $s = 0.44$ and move rate $m = 0.36$ in the integrated transformation from A' and A^* are calculated with regard to (4.17) and (4.23). Finally, the s and m are used to transform B' according to (4.14)-(4.16) and (4.18)-(4.20), resulting in consequence $B^*(5.83,6.26,7.83)$. For the latter, the scale rate $s = 0.69$ and move criterion $\mathbb{RC} = 0.28$ are calculated from (5.38) and (5.35), which are used to scale and move B' to result in $B^*(5.54,5.97,7.97)$.

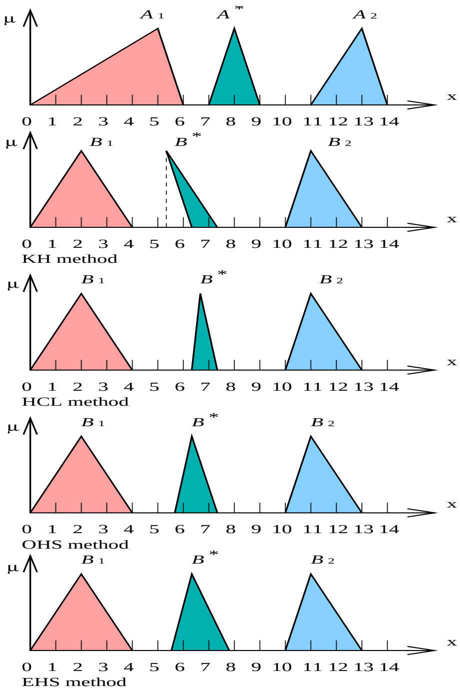

Figure 5.11: Example 1

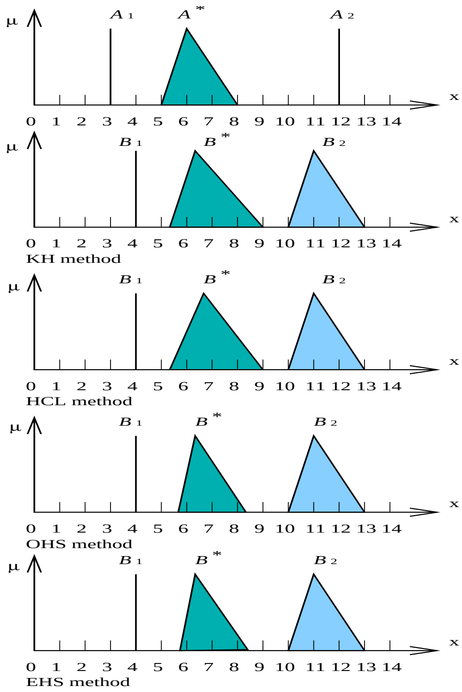

Figure 5.12: Example 2

For this case, the KH method resulted in an invalid conclusion (not even a membership function) while the other three concluded with normal and valid fuzzy sets.

Example 2. The second case considers the infinity of the scale rate. The given observation is a triangular fuzzy set $(5,6,8)$. Table 5.2 and Fig. 5.12 present the antecedents and interpolated fuzzy sets. The OHS interpolation $(5.71, 6.28, 8.16)$ is ob-

Table 5.2: Results for example 2, with $A^* = (5,6,8)$

Attribute Values	Results	
$A_1 = (3,3,3)$	Method	B^*
$A_2 = (12,12,12)$	KH	(5.33, 6.33, 9.00)
$B_1 = (4,4,4)$	HCL	(5.33, 6.55, 9.00)
$B_2 = (10,11,13)$	OHS	(5.71, 6.28, 8.16)
	EHS	(5.74, 6.23, 8.18)

tained as follows: First the ratio between the support of A^* and the distance of $Rep(A_1)$

and $Rep(A_2)$ is calculated. The support of B^* is then computed by retaining the same ratio but based on the distance of $Rep(B_1)$ and $Rep(B_2)$. Finally, the move transformation is applied as usual. With the same scale rate as in the EHS method, the enhanced scale transformation results in the same scaled fuzzy set. However, the enhanced move transformation leads to a different output of $(5.74, 6.23, 8.18)$. The comparative results show that the KH and HCL methods perform similarly (the supports of the resultant fuzzy sets are identical since they are computed in the same way) while the OHS and EHS methods also generate very reasonable outcomes.

Example 3. The third case considers a similar situation to example 1 but the observation is a singleton $A^* = (8,8,8)$. Table 5.3 and Fig. 5.13 present the results. In this

Table 5.3: Results for example 3, with $A^* = (8,8,8)$

Attribute Values	Results	
$A_1 = (0,5,6)$	Method	B^*
$A_2 = (11,13,14)$	KH	(7.27, 5.38, 6.25)
$B_1 = (0,2,4)$	HCL	[7.27,6.25]
$B_2 = (10,11,13)$	OHS	(6.49, 6.49, 6.49)
	EHS	(6.49, 6.49, 6.49)

case, the KH method once again generates an invalid fuzzy set and the HCL method even produces a non-triangular fuzzy set. However, the OHS and EHS result in the same singleton conclusions, which are rather intuitive given the singleton-valued condition.

Example 4. This example concerns a trapezoidal based fuzzy interpolation and the compatible RV definition is used here. As there is no obvious indication for HCL method to handle trapezoidal fuzzy sets, only KH method is used in comparison. All the attributes and results with observation $A^* = (6,6,9,10)$ are shown in Table 5.4 and Fig. 5.14. For the OHS method, $A' = (5.30, 7.85, 8.85.9.85)$ and $B' = (4.81, 6.33, 7.33, 8.33)$ are calculated by interpolation of A_1, A_2 and B_1, B_2, respectively, with $\lambda = 0.48$, which is calculated from (5.12). The interpolation via scale and move transformations is then carried out according to the steps listed in section

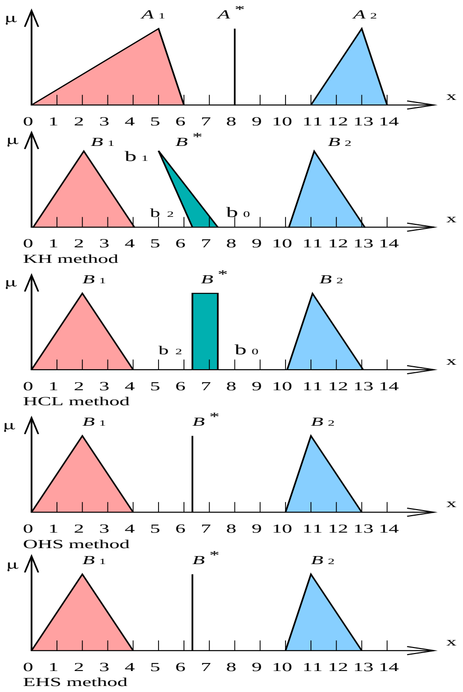

Figure 5.13: Example 3

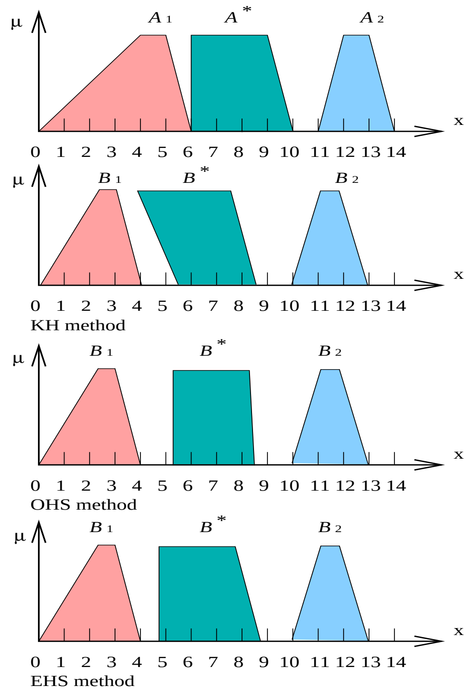

Figure 5.14: Example 4

Table 5.4: Results for example 4, with $A^* = (6,6,9,10)$

Attribute Values	Results	
$A_1 = (0,4,5,6)$	Method	B^*
$A_2 = (11,12,13,14)$	KH	(5.45, 4.25, 7.5, 8.5)
$B_1 = (0,2,3,4)$	HCL	-
$B_2 = (10,11,12,13)$	OHS	(5.23, 5.23 ,7.61, 8.32)
	EHS	(4.83, 4.83, 7.83, 8.83)

5.2.4: 1) The bottom support scale rate (0.88) and top support scale rate (3.0) from A' to A^* are calculated according to (5.33) respectively. 2) The top support scale ratio (0.68) from A' to A^* is calculated according to (5.22). 3) A' is scaled to generate $A'' = (5.76, 6.48, 9.48, 9.76)$ using the bottom and top scale rates calculated in step 1. Note that A'' is a valid fuzzy set which has the same representative value and has the same bottom and top support lengths as A^*. 4) According to (5.34), the bottom and top support scale rates (0.88 and 2.38) over B' are computed. 5) B' is scaled to generate $B'' = (5.09, 5.52, 7.90, 8.18)$ using the bottom and top scale rates calculated in step 4. 6) The move ratio is calculated from A'' to A^* according to (5.26). Its value is 1.0 as A^* has a vertical left slope. This move ratio is used to move B'' to obtain the resultant fuzzy set $B^* = (5.23, 5.23, 7.61, 8.32)$. Similarly the enhanced HS method results in $B^* = (4.83, 4.83, 7.83, 8.83)$. In this case, the KH method once again generates an invalid fuzzy set (which does not satisfy the definition of a membership function). However, both the OHS and EHS methods result in valid conclusions, which still maintain the property of the left vertical slopes.

Example 5. This example shows an interpolation of rules involving hexagonal fuzzy sets. It also demonstrates the interpolation involving different shapes of fuzzy sets. For simplicity, the average RV definition is adopted in this example. Again, since there is no obvious indication for the HCL method to be able to handle such fuzzy sets, only the KH method is used in comparison. All the attribute values and results with respect to the observation $A^* = (6, 6.5, 7, 9, 10, 10.5)$ are shown in Table 5.5 and Fig. 5.15. Note that in this example, the two intermediate points a_1 and a_4 of each fuzzy set involved have a membership value of 0.5.

Table 5.5: Results for example 5, with $A^* = (6, 6.5, 7, 9, 10, 10.5)$

Attribute Values	Results	
$A_1 = (0, 1, 3, 4, 5, 5.5)$	Method	B^*
$A_2 = (11, 11.5, 12, 13, 13.5, 14)$	KH	(5.73, 6.00,5.89,8.56, 9.59, 10.09)
$B_1 = (0, 0.5, 1, 3, 4, 4.5)$	HCL	-
$B_2 = (10.5, 11, 12, 13, 13.5, 14)$	OHS	(5.64, 5.98, 6.29, 8.63, 9.46, 9.93)
	EHS	(5.28, 5.62, 5.94, 8.86, 9.86, 10.36)

The original HS interpolation is chosen to illustrate the procedure of the calculation. $A' = (5.94, 6.67, 7.86, 8.86, 9.59, 10.09)$ and $B' = (5.67, 6.17, 6.94, 8.40, 9.13, 9.63)$ are calculated by interpolation of A_1, A_2 and B_1, B_2 (with $\lambda = 0.54$), respectively. Then, the interpolation via scale and move transformations is carried out according to the steps listed in section 5.2.4: 1) The bottom support scale rate (1.08), middle

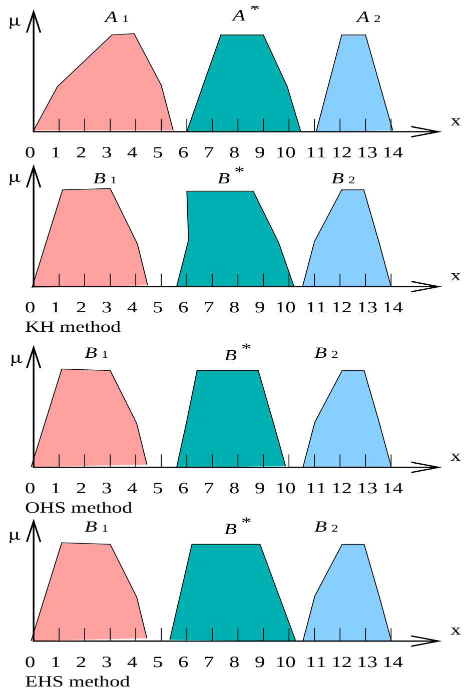

Figure 5.15: Example 5

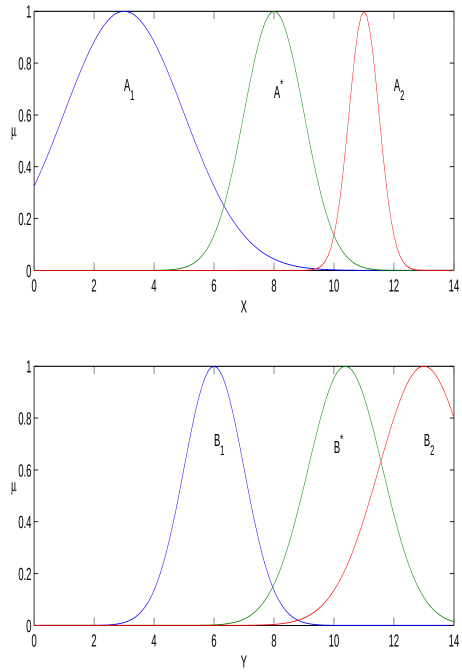

Figure 5.16: Example 6

support scale rate (1.20) and top support scale rate (2.0) from A' to A^* are calculated according to (5.33), respectively. 2) The middle and top support scale ratios (0.25 and 0.35) from A' to A^* are calculated according to (5.22). 3) A' is scaled to generate $A'' = (5.79, 6.39, 7.32, 9.32, 9.89, 10.29)$ using the bottom, middle and top scale rates calculated in step 1). Note that A'' is a valid fuzzy set which has the same representative value and the same three supports lengths as A^*. 4) According to (5.34), the bottom, middle and top support scale rates (1.08, 1.18 and 1.60) over B' are computed. 5) B' is scaled to generate $B'' = (5.50, 5.91, 6.50, 8.83, 9.39, 9.80)$ using the scale rates calculated in step 4). 6) Two sub-moves are required in performing the move transformation in this case: 6.1), The bottom sub-move ratio (0.29) is calculated from A'' to A^* according to (5.26). This sub-move ratio is used to move A'' to get $A^{(0)} = (6.00, 6.42, 7.08, 9.08, 9.92, 10.50)$, and to move B'' to obtain $B^{(0)} = (5.64, 5.93, 6.35, 8.68, 9.41, 9.93)$. Note that after this sub-move, A'' has the same bottom support as A^*. 6.2) The second sub-move moves the middle and top supports of $A^{(0)}$ to the desired places. In particular, the sub-move ratio (0.24) calculated from

(5.26) is used to move $B^{(0)}$ to the final result $B^* = (5.64, 5.98, 6.29, 8.63, 9.46, 9.93)$. As a verification, A^* is obtained by moving $A^{(0)}$ with the same sub-move ratio.

In this case, both the OHS and EHS methods still ensure unique, normal and valid fuzzy sets, compared to the invalid result generated via the KH method.

Example 6. This case considers an interpolation with Gaussian membership functions. As there are no explicit Gaussian based interpolation solutions for HCL and KH methods, only the results of OHS (or EHS, as they result in the same outputs) method together with the attribute values and observation $A^* = p(x) = e^{\frac{-(x-8)^2}{2*1^2}}$ are presented in Table 5.6 and Fig. 5.16. The OHS (or EHS) method results in a sensible Gaussian conclusion in this case.

Table 5.6: Results for example 6, with $A^* = e^{\frac{-(x-8)^2}{2*1^2}}$

Attribute Values	Results	
$A_1 = e^{\frac{-(x-3)^2}{2*2^2}}$	Method	B^*
$A_2 = e^{\frac{-(x-11)^2}{2*0.5^2}}$	KH	-
$B_1 = e^{\frac{-(x-6)^2}{2*1^2}}$	HCL	-
$B_2 = e^{\frac{-(x-13)^2}{2*1.5^2}}$	OHS (or EHS)	$e^{\frac{-(x-10.38)^2}{2*1.24^2}}$

Example 7. This example concerns an interpolation of multiple antecedent variables with trapezoidal membership functions. Specially, two rules $A_{11} \wedge A_{21} \Rightarrow B_1$, $A_{12} \wedge A_{22} \Rightarrow B_2$ and the observations $A_1^* = (6, 7, 9, 11)$, $A_2^* = (6, 8, 10, 12)$ are given to determine the result B^*. For demonstration purposes, only the original HS method and the compatible RV definition are employed in this example. Table 5.7 and Fig. 5.17 summarise the results. In this case, the parameters λ_1 for the first variable is 0.54 and λ_2 for the second is 0.44. The average 0.49 is used to calculate the intermediate rule result B'. The average of two bottom support scale rates (1.14 and 1.69) and the average of two top support ratios (0.22 and 0.07) are computed, equalling 1.41 and 0.15 respectively, and used as the combined bottom support scale rate and top support scale ratio. These together with the combined move rate, the average (0.35) of the two

Table 5.7: Results for example 7, with $A_1^* = (6, 7, 9, 11)$ and $A_2^* = (6, 8, 10, 12)$

Attribute Values	Results	
$A_{11} = (0, 4, 5, 6)$	Method	B^*
$A_{12} = (12, 14, 15, 16)$	KH	$(5.45, 5.94, 7.13, 8.31)$
$A_{21} = (11, 12, 13, 14)$	HCL	-
$A_{22} = (1, 2, 3, 4)$	OHS	$(4.37, 5.55, 7.48, 9.33)$
$B_1 = (0, 2, 3, 4)$		
$B_2 = (10, 11, 12, 13)$		

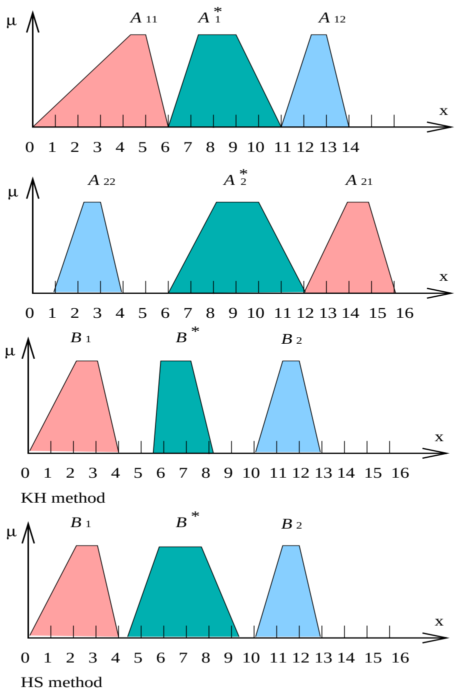

Figure 5.17: Example 7

move rates (0.53 and 0.18), are employed to transfer B' to achieve the final result B^*. Both the KH method and HS method resulted in a valid set in this example. Interestingly, the resultant fuzzy set of the OHS method reflects better shapes of the original observations than that obtained by the KH method.

5.6 Extensions

All fuzzy interpolation techniques in the literature assume that two closest adjacent rules to the observation are available. In addition, most interpolation methods presume that such rules must flank the observation for each attribute (but not necessarily in the same order). In practice, however, there may be a different number of the closest rules to a given observation, and the attribute values of these rules may lie on one side of the observation. The adoption of these two assumptions inevitably limits the potential applications of the existing work. In fact, this is the reason why the existing interpolation methods are limited to toy examples and have not yet been applied to real world prediction or classification problems. To resolve this problem, this section extends the HS methods to allow interpolations that involve multiple rules, without making the strong condition that antecedent attributes flank the observation. Furthermore, exploiting the generality of this newly developed method, extrapolation can be performed over multiple rules in a straightforward manner.

5.6.1 Interpolation with Multiple Rules

To allow fuzzy interpolation with more than two rules given a rule base, the first step is to choose n ($n \geq 2$) closest rules from the rule base. Then, selected rules are used to construct the intermediate fuzzy rule. Once the intermediate rule is worked out, the rest of the process remains the same as described in Sections 5.2 and 5.3. The following shows these two important steps:

5.6.1.1 Choose the Closest *n* Rules

Without losing generality, suppose that a rule R_i and an observation are represented by

$$Rule\ R_i : if\ X_1\ is\ A_{1i}\ and \ldots and\ X_m\ is\ A_{mi}\ then\ Y\ is\ B_i, \tag{5.46}$$

$$Observation : X_1\ is\ A_1^*\ and \ldots and\ X_m\ is\ A_m^*. \tag{5.47}$$

According to the distance definition (5.11) between two fuzzy terms, the distances d_k, $k = 1, \ldots, m$, between the pairs of A_{ki} and A_k^* can be calculated as:

$$d_k = d(A_{ki}, A_k^*) = d(Rep(A_{ki}), Rep(A_k^*)). \tag{5.48}$$

As attributes may have different domains, the absolute distances may not be compatible with each other. To make these comparable, each distance measure is normalised into the range of 0 to 1:

$$d'_k = \frac{d(A_{ki}, A^*_k)}{Max_k - Min_k} = \frac{d(Rep(A_{ki}), Rep(A^*_k))}{Max_k - Min_k}, \tag{5.49}$$

where Max_k and Min_k are the maximal and minimal values of attribute k given. The distance dis between a rule and an observation can be calculated as the average of all attributes' distances. A particular distance definition, which is to be used in the later implementation, can be written as follows:

$$dis = \sqrt{{d'_1}^2 + {d'_2}^2 + \ldots + {d'_m}^2}. \tag{5.50}$$

If, however, the importance of attributes are not equal, weights may be used. Note that if a conditional part of a rule is missing, the distance of this attribute is treated as 0 to reflect that any data value is very close to the *null* attribute value. This allows for measuring the distance between a given observation and rules which may not have fuzzy sets associated with certain attributes.

Once the distance definition of (5.50) is given, the distances between a given observation and all rules in the rule base can be calculated. The n rules which have minimal distances are chosen as the closest n rules from the observation. It is worth noting that the n closest rules do not necessarily flank the observation. In the extreme case, all the chosen rules may lie on one side, resulting in extrapolation rather than interpolation (see section 5.6.2).

5.6.1.2 Construct the Intermediate Rule

This section proposes how to construct the intermediate rule after n ($n \geq 2$) closest rules have been chosen. Let W_{ki}, $i = 1, \ldots, n$, $k = 1, \ldots, m$, denote the weight to which the kth term of the ith fuzzy rule contributes to constructing the kth intermediate fuzzy term A'_k. Obviously, the longer the distance from A_{ki} to A^*_k, the less value W_{ki} should take. In particular, the inversion of the distance is used:

$$W_{ki} = \frac{1}{d(A_{ki}, A^*_k)}, \tag{5.51}$$

where $d(A_{ki}, A_k^*)$ is defined in (5.48). Of course, alternative non-increasing functions such as $W_{ki} = exp^{-d(A_{ki}, A_k^*)}$ may be adopted to assign different weights.

For each attribute k, the weights W_{ki}, $i = 1, \ldots, n$ are used to compute the intermediate fuzzy term A_k^*. Prior to that, they are normalised as follows:

$$W'_{ki} = \frac{W_{ki}}{\sum_{t=1,\ldots,n} W_{kt}}, \tag{5.52}$$

so that their sum equals to 1. The intermediate fuzzy term A''_k, $k = 1, \ldots, m$, are computed as:

$$A''_k = \sum_{i=1,\ldots,n} W'_{ki} A_{ki}, \tag{5.53}$$

which is the same as (5.14) when only two rules ($n = 2$) are considered for interpolation. That is, the two-rule interpolation case is one special case of the generalised multi-rule interpolation.

In the two-rule interpolation case, the A''_k calculated via (5.53) has the same Rep as the input A_k^*. However, this is generally not true when more than two rules are involved (that is why symbol A''_k, rather than A'_k, is used here). Thus, it does not satisfy the requirement of having the same Rep value, as imposed by the scale and move transformations. In order to solve this problem, two possible ways, namely the *zoom* and *shift*, are suggested to modify A''_k so that it becomes a new fuzzy intermediate term A'_k which has the same Rep as A_k^*.

First, the *zoom* is suggested in which A''_k is zoomed by γ_k, $k = 1, \ldots, m$ as follows:

$$A'_k = \gamma_k A''_k, \tag{5.54}$$

where γ_k is a constant defined as

$$\gamma_k = \frac{Rep(A_k^*)}{Rep(A''_k)}. \tag{5.55}$$

In so doing, the following holds:

$$Rep(A'_k) = Rep(A_k^*). \tag{5.56}$$

Regarding the consequent, by analogy to (5.53), the intermediate fuzzy output B'' can be computed by

$$B'' = \sum_{i=1,\ldots,n} W'_{ai} B_i, \tag{5.57}$$

where W'_{ai} is the mean of W'_{ki}:

$$W'_{ai} = \frac{1}{m}\sum_{k=1}^{m} W'_{ki}. \tag{5.58}$$

B'' is then zoomed to B' as follows:

$$B' = B''\gamma_a, \tag{5.59}$$

where γ_a is the mean of the zoom parameters γ_k, $k = 1, \ldots, m$,

$$\gamma_a = \frac{1}{m}\sum_{k=1}^{m} \gamma_k. \tag{5.60}$$

Alternatively, *shift* may be applied to A''_k, $k = 1, \ldots, m$ as follows:

$$A'_k = A''_k + \delta_k(Max_k - Min_k), \tag{5.61}$$

where Max_k and Min_k are maximal and minimal values of attribute k and δ_k is defined as

$$\delta_k = \frac{Rep(A^*_k) - Rep(A''_k)}{Max_k - Min_k}. \tag{5.62}$$

In so doing, the following holds:

$$Rep(A'_k) = Rep(A^*_k). \tag{5.63}$$

Similarly, the intermediate fuzzy output B'' can be computed by

$$B'' = \sum_{i=1,\ldots,n} W'_{ai}B_i, \tag{5.64}$$

where W'_{ai} is the mean of W'_{ki}:

$$W'_{ai} = \frac{1}{m}\sum_{k=1}^{m} W'_{ki}. \tag{5.65}$$

B'' is then shifted to B' as follows:

$$B' = B'' + \delta_a(Max - Min), \tag{5.66}$$

where Max and Min are maximal and minimal values of output variable and δ_a is the mean of the shift parameters δ_k, $k = 1, \ldots, m$,

$$\delta_a = \frac{1}{m}\sum_{k=1}^{m} \delta_k. \tag{5.67}$$

Using either the zoom or shift method, the intermediate fuzzy rule

$$if\ X_1\ is\ A'_1\ and\ldots and\ X_m\ is\ A'_m\ then\ Y\ is\ B'_m$$

can be obtained from (5.54), (5.59) or (5.61), (5.66). The rest of the interpolation reasoning is hence applied to this intermediate rule and the observed fuzzy term vector, in the same way as presented in sections 5.2 or 5.3. An example follows to explain how this works.

Example 8. Three rules $A_i \wedge B_i \Rightarrow C_i$, $i = 1,2,3$ and the observations A^*, B^* are given in Table 5.8. For the first attribute A, the distances between A_i, $i = 1,2,3$ and the

Table 5.8: Example 8

Attribute Values	$A_1 = (0,1,3), B_1 = (1,2,3), C_1 = (0,2,3)$ $A_2 = (8,9,10), B_2 = (7,9,10), C_2 = (9,10,11)$ $A_3 = (11,13,14), B_3 = (11,12,13), C_3 = (12,13,14)$
Observation	$A^* = (3.5,5,7), B^* = (5,6,7)$

observation (A^*) are calculated as 4, 4, and 8 respectively (assuming the center of core Rep is adopted). According to (5.51), the weights are calculated as 0.25, 0.25, and 0.13 respectively. They are normalised using (5.52) with the new weights being 0.4, 0.4 and 0.2. According to (5.53), a fuzzy term $A'' = (5.4,6.6,8.0)$ is obtained using the normalised weights. As A'' does not have the same Rep as the input A^*, either zoom or shift method should be applied.

The zoom method is applied first. According to (5.55), $\gamma_A = 0.76$ is computed. The fuzzy term A'' is zoomed by γ_A to generate the required intermediate fuzzy set $A' = (4.09,5,6.06)$. Similarly, B_1, B_2 and B_3 have normalised weights 0.33, 0.44 and 0.22 in constructing the intermediate fuzzy set $B'' = (5.89,7.33,8.33)$. With $\gamma_B = 0.82$, it is zoomed to $B' = (4.82,6,6.82)$. The fuzzy set $C'' = (6.33,7.7,8.7)$ can be computed using the average weights of A and B for three rules (0.37, 0.42 and 0.21 respectively) according to (5.57). The intermediate output $C' = (4.99,6.07,6.86)$ can then be computed using the average of γ_A and γ_B, that is 0.79, with respect to (5.59). This is shown in Fig. 5.18.

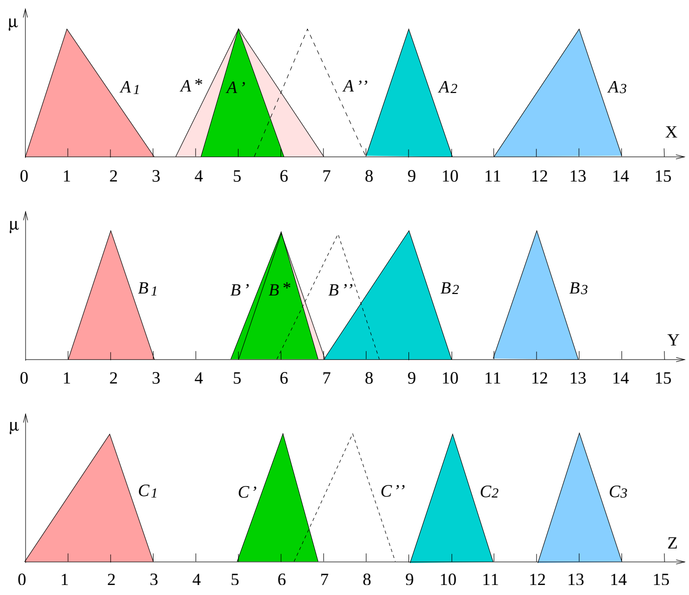

Figure 5.18: Example 8

Alternatively, the shift method can be applied. According to (5.62), $\delta_A = -0.11$ is computed. The fuzzy term A'' and δ_A are used to generate the required intermediate fuzzy set $A' = (3.8, 5, 6.4)$. Similarly, $B'' = (5.89, 7.33, 8.33)$ is constructed with the normalised weights of B_1, B_2 and B_3 (0.33, 0.44, and 0.22 respectively). $B' = (4.56, 6, 7)$ is then computed based on B'' and $\delta_B = -0.11$. For the consequent, fuzzy set $C'' = (6.33, 7.7, 8.7)$ can be computed using the average weights of attributes A and B for three rules (0.37, 0.42 and 0.21 respectively) according to (5.64). The intermediate output $C' = (4.76, 6.13, 7.13)$ can then be computed using the average of δ_A and δ_B, that is -0.11, with respect to (5.66). It is worth noting that zoom and shift methods produce the same intermediate fuzzy rule $A'' \wedge B'' \Rightarrow C''$, but not $A' \wedge B' \Rightarrow C'$.

5.6.2 Extrapolation

The extrapolation is readily extendable. It is a special case of interpolation with multiple rules as described Section 5.6.1. In particular, when all of the n closest rules chosen (see 5.6.1.1) lie on one side of the given observation, the interpolation problem becomes an extrapolation one. In fact, both choosing the closest rules and constructing the intermediate rule are carried out here in the same way as performed in Section 5.6.1.

An example follows to explain the computation. Suppose only the second and third rules in example 8 are considered, the interpolation becomes an extrapolation of two rules.

Example 9. Two rules $A_i \wedge B_i \Rightarrow C_i$, $i = 2,3$ and the observations A^*, B^* are given in Table 5.8 to carry out fuzzy extrapolation. Again, assume the center of core Rep is used. For the first attribute A, the normalised weights of A_i, $i = 2,3$ are computed to be 0.67 and 0.33. According to (5.53), a fuzzy term $A'' = (9, 10.33, 11.33)$ is obtained. As A'' does not have the same Rep as the input A^*, zoom or shift method has to be used.

Consider the use of zoom method first. According to (5.55), $\gamma_A = 0.48$ is computed. The fuzzy term A'' is zoomed by γ_A to $A' = (4.36, 5, 5.48)$. Similarly, B_2 and B_3 have normalised weights 0.67 and 0.33 in constructing $B'' = (8.33, 10, 11)$. With $\gamma_B = 0.6$, B'' is zoomed to $B' = (5, 6, 6.6)$. The fuzzy set $C'' = (10, 11, 12)$ can be computed using the average weights of A and B for two rules (0.67 and 0.33) according to (5.57). The intermediate output $C' = (5.42, 5.96, 6.50)$ can then be computed using the average of γ_A and γ_B, that is 0.54, with respect to (5.59). This is shown in Fig. 5.19.

Alternatively, the shift method can be used. According to (5.62), $\delta_A = -0.38$ is obtained. Fuzzy term A'' and δ_A are used to generate the required intermediate fuzzy set $A' = (3.67, 5, 6)$. Similarly, B_2 and B_3 have normalised weights 0.67 and 0.33 in constructing the intermediate fuzzy set $B'' = (8.33, 10, 11)$. With $\delta_B = -0.33$, B'' is shifted to $B' = (4.33, 6, 7)$. The fuzzy set $C'' = (10, 11, 12)$ can be computed using the average weights of A and B for two rules (0.67 and 0.33) according to (5.64). The intermediate output $C' = (5, 6, 7)$ can then be computed using the average of δ_A and δ_B, that is -0.36, with respect to (5.66).

The rules which are used for extrapolation may be twisted. That is, their associated fuzzy sets may not have the same order (as in Example 9) for each attribute. The following shows this case.

Example 10. Two rules $A_2 \wedge B_3 \Rightarrow C_2$ and $A_3 \wedge B_2 \Rightarrow C_3$, and the observations A^*, B^* are given in Table 5.8 for fuzzy extrapolation. For the first attribute A, $A'' =$

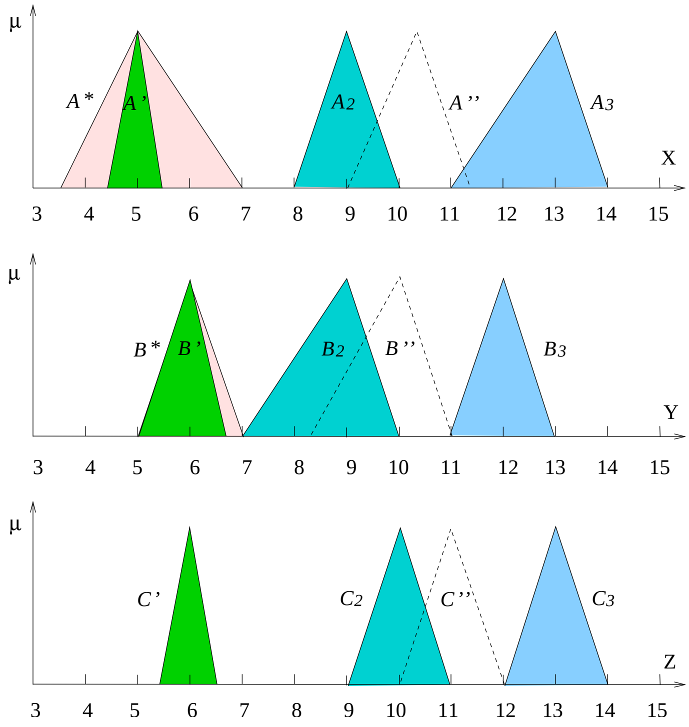

Figure 5.19: Example 9

$(9, 10.33, 11.33)$ is obtained with the normalised weights of A_i, $i = 2, 3$, being 0.67 and 0.33. Consider the zoom method is used. According to (5.55), $\gamma_A = 0.48$ is computed. The fuzzy term A'' is zoomed by γ_A to generate $A' = (4.36, 5, 5.48)$. Similarly, B_2 and B_3 have normalised weights 0.33 and 0.67 in constructing $B'' = (8.33, 10, 11)$. With $\gamma_B = 0.6$, B'' is zoomed to $B' = (5, 6, 6.6)$. The fuzzy set $C'' = (10.5, 11.5, 12.5)$ can be computed using the average weights of A and B for two rules (0.5 and 0.5) according to (5.57). The intermediate output $C' = (5.69, 6.23, 6.78)$ can then be computed using the average of γ_A and γ_B, that is 0.54, with respect to (5.59). This is shown in Fig. 5.20.

Alternatively, the shift method can be applied. Fuzzy term A'' is shifted (with $\delta_A = -0.38$) to $A' = (3.67, 5, 6)$. B_2 and B_3 have normalised weights 0.33 and 0.67 in constructing $B'' = (8.33, 10, 11)$. With $\delta_B = -0.33$, B'' is shifted to $B' = (4.33, 6, 7)$. Fuzzy set $C'' = (10.5, 11.5, 12.5)$ can be computed using the average weights of 0.5 and 0.5. The intermediate output $C' = (5.5, 6.5, 7.5)$ can then be computed using the

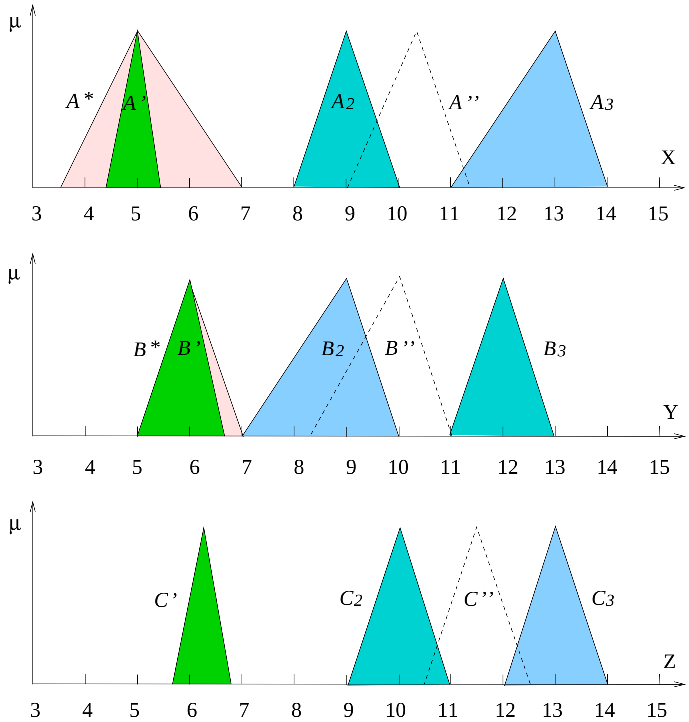

Figure 5.20: Example 10

average of δ_A and δ_B, that is -0.36, with respect to (5.66).

It is worth noting that the values of γ and δ should be close to 1 and 0, respectively, with respect to (5.55) and (5.62). If they are far away to those values, they may cause problems, which will be discussed in chapter 8.

5.7 Summary

This chapter has proposed a generalised, scale and move transformation-based, interpolative reasoning method (OHS method) which can handle interpolation of complex polygonal, Gaussian and other bell-shaped fuzzy membership functions. The enhanced HS method has also been proposed to preserve the piecewise linearity property in interpolating any polygonal fuzzy sets. The case studies have been given showing how the methods work in numerical examples. In addition, the extension to handle interpolation (and extrapolation) involving multiple variables and multiple rules has been addressed. This helps bridge the gap between theory and application as the existing fuzzy interpolations have not been applied to real world prediction or classification problems, which may often require reasoning with multiple rules and extrapolation.

The original HS method not only inherits the common advantages of fuzzy interpolative reasoning – allowing inferences to be performed with simple and sparse rule bases, but also has two other advantages. Firstly it provides a degree of freedom to choose various RV definitions for different application requirements. Secondly, it can handle the interpolation of multiple rules, with each rule having multiple antecedent variables associated with arbitrary polygonal fuzzy membership functions. In addition to the advantages the OHS having, the enhanced HS method has extra two. Firstly, it has less computation cost than OHS (see chapter 6). Secondly, it preserves the piecewise linearity property for any polygonal fuzzy functions (see chapter 6). It is worth stressing that the piecewise linearity property is essential for ignoring artificial characteristic points in performing fuzzy interpolations.

Chapter 6

Transformation Based Interpolation: Evaluations

This chapter evaluates the interpolative reasoning methods proposed in chapter 5 from different aspects including the dependency of the fuzziness of conclusion on observation, the preservation of the piecewise linearity and the computational complexity. The comparisons to other existing approaches such as KH and the general method are provided. The results show that the original HS method preserves the piecewise linearity in interpolations involving triangular fuzzy sets and has $O(n^2)$ computation complexity (n is the number of characteristic points for each fuzzy set). The results are more encouraging for the enhanced HS method, which not only preserves piecewise linearity for interpolations involving arbitrary polygonal fuzzy sets, but also requires only $O(n)$ computation time.

6.1 Evaluation Criteria

In order to compare different interpolative reasoning methods, the evaluation criteria in terms of the dependency of the fuzziness of conclusion on observation, the preservation of the piecewise linearity and the computational complexity have been used.

The dependency of the fuzziness of conclusion on observation shows the degree of uncertainty of the interpolative reasoning method. It is computed by the ratio of the fuzziness of the conclusion with respect to that of the observation. This evaluation brings different views of fuzziness derivation for fuzzy interpolation methods. Generally speaking, the fuzzier the observation is, the fuzzier the conclusion should be. Fuzzy dependency can not be simply used to justify some interpolation methods outperforming others. Instead, it is used as a guide to choose suitable fuzzy interpolative reasoning methods under certain circumstances.

Preservation of piecewise linearity is an essential property which reflects how good the interpolative reasoning method handles the points between two consecutive characteristic points. If the piecewise linearity is preserved, it is safe to merely consider the characteristic points rather than the infinite pairs of points (generated from an infinite number of α-cut levels). Therefore, it is worth investigating what happens to the non-characteristic points when interpolation is applied. Are they still on the line between two interpolated characteristic points? Or if not, what is the deviation?

Fuzzy interpolation techniques are desired to give prompt responses when they are implemented in time critical applications. Therefore, the complexity analysis [CLRS90] in terms of time and space is an important issue for the interpolation methods. However, more attention is drawn to time complexity rather than space complexity as the latter nearly vanishes when the technology for storage hardware has recently made significant progress. Although the current processors have been developed to a high comparative speed, they are still not able to handle NP complexity problems. In this chapter, the time complexity with respect to n (the number of characteristic points for the fuzzy sets involved) is investigated for three existing interpolative reasoning methods along with the proposed ones.

6.2 Evaluations

6.2.1 Dependency of the Fuzziness of Conclusion on Observation

The uncertainty of the interpolative reasoning method can be captured by the dependency of the fuzziness of conclusion on observation. Using this criterion, the work of [TB00, MBKK99] has compared the following methods: KH [KH93a, KH97], modified KH [TB00], Vass-Kalinóv-Kóczy (VKK) [VKK92], Kóczy-Hirota-Gedeou

(KHG) [KHG97] and the general fuzzy rule interpolation algorithm [BGK96, BGK95].

Due to the variety of the existing interpolative approaches, two simplest rules $A_1 \Rightarrow B_1$ and $A_2 \Rightarrow B_2$ and an observation A^* which involve only triangular fuzzy sets are used here to provide a uniform platform for comparisons. To be compatible to the comparison in [TB00, MBKK99], the three characteristic points are indexed with 1, 2 and 3. That is $A_i = (a_{i1}, a_{i2}, a_{i3})$, $B_i = (b_{i1}, b_{i2}, b_{i3})$, $i = 1, 2$, $A^* = (a_1^*, a_2^*, a_3^*)$, and $B^* = (b_1^*, b_2^*, b_3^*)$. Instead of comparing the fuzziness of the whole conclusion $(b_3^* - b_1^*)$ to observation $(a_3^* - a_1^*)$, the partial fuzziness of them are investigated. For this purpose, the central point (or reference point, defined in the work of [BGK96, BGK95]) of a fuzzy set A is adopted as follows.

$$cp(A) = \frac{inf\{supp(A_\alpha)\} + sup\{supp(A_\alpha)\}}{2}, \tag{6.1}$$

where $\alpha = height(A)$, i.e., the highest membership degree of a fuzzy set A. The fuzziness of the conclusion $b_3^{*c} = b_3^* - b_2^*$ is estimated with respect to the observation fuzziness $a_3^{*c} = a_3^* - a_2^*$. The following shows the dependency functions for the KH, modified KH, VKK, KHG and the general fuzzy interpolation methods. Interested readers may refer to the work of [TB00, MBKK99] for further relevant discussions.

From section 2.4.1, the dependency of the right point of the conclusion on that of the observation, f_f^{KH}, can be expressed as

$$b_3^{*c} = f_f^{KH}(a_3^{*c}) = \lambda^{KH} a_3^{*c} + \delta^{KH}, \tag{6.2}$$

where

$$\begin{aligned} \lambda^{KH} &= \frac{b_{23} - b_{13}}{a_{23} - a_{13}}, \\ \delta^{KH} &= \lambda^{KH}(-a_{13}^c) + b_{13}^c. \end{aligned}$$

For the modified KH method [TB00],

$$b_3^{*c} = f_f^{MKH}(a_3^{*c}) = \lambda^{MKH} a_3^{*c} + \delta^{MKH}, \tag{6.3}$$

where

$$\begin{aligned} \lambda^{MKH} &= \frac{(b_{23} - b_{13}) - (b_{22} - b_{12})}{a_{23} - a_{13}}, \\ \delta^{MKH} &= \lambda^{MKH}(-a_{13}^c) + b_{13}^c. \end{aligned}$$

The fuzziness dependency of the VKK interpolative reasoning method [MSK99] is computed by

$$\begin{aligned}
b_3^{*c} &= f_f^{VKK}(a_3^{*c}) \\
&= \frac{1}{2}[\lambda_1^{VKK}(a_3^{*c})^2 + (2a_2^*\lambda_1^{VKK} + \lambda_2^{VKK} + \lambda_3^{VKK})a_3^{*c} - \lambda_1^{VKK}(a_3^*)^2 \\
&+ (\lambda_3^{VKK} - \lambda_2^{VKK})a_3^* + \lambda_1^{VKK}(a_2^*)^2 + (\lambda_2^{VKK} + \lambda_3^{VKK})a_2^* + \lambda_4^{VKK}] - \delta^{VKK},
\end{aligned} \tag{6.4}$$

where

$$\begin{aligned}
\lambda_1^{VKK} &= \frac{\frac{b_{23}-b_{21}}{a_{23}-a_{21}} - \frac{b_{13}-b_{11}}{a_{13}-a_{11}}}{(a_{23}+a_{21}) - (a_{13}+a_{11})} \\
\lambda_2^{VKK} &= \frac{\frac{b_{13}-b_{11}}{a_{13}-a_{11}}(a_{23}+a_{21}) - \frac{b_{23}-b_{21}}{a_{23}-a_{21}}(a_{13}+a_{11})}{(a_{23}+a_{21}) - (a_{13}+a_{11})} \\
\lambda_3^{VKK} &= \frac{(b_{21}+b_{23}) - (b_{11}+b_{13})}{(a_{21}+a_{23}) - (a_{11}+a_{13})} \\
\lambda_4^{VKK} &= \frac{(a_{21}+a_{23})(b_{11}+b_{13}) - (b_{21}+b_{23})(a_{11}+a_{13})}{(a_{21}+a_{23}) - (a_{11}+a_{13})} \\
\delta^{VKK} &= \frac{b_{22}-a_{22}}{a_{22}-a_{12}}a_2^* + \frac{a_{22}b_{12}-a_{12}b_{22}}{a_{22}-a_{12}}
\end{aligned}$$

The KHG fuzzy rule interpolation method [KHG97] has

$$b_3^{*c} = f_f^{KHG}(a_3^{*c}) = (a_3^{*c})\frac{b_{22}-b_{21}}{a_{22}-a_{21}}. \tag{6.5}$$

The fuzziness of the conclusion for the general fuzzy rule interpolation method [BGK95] is computed as

$$b_3^{*c} = f_f^{ge}(a_3^{*c}) = f^+(a_3^*, a_3^i, b_3^i, 0, x_M, 0, y_M), \tag{6.6}$$

where function f^+ is the revision function defined in [KHG97] (as shown in Fig. 2.5). a_3^i and b_3^i are the right points of the intermediate interpolated fuzzy set (see [BGK95]).

$$a_3^i = (1-\lambda_2)a_{13} + \lambda_2 a_{23}, \tag{6.7}$$

$$b_3^i = (1-\lambda_2)b_{13} + \lambda_2 b_{23}, \tag{6.8}$$

where $\lambda_2 = \frac{a_2^*-a_{12}}{a_{22}-a_{12}}$.

To facilitate the comparison to the proposed original HS method, the scale and move transformations are applied to the antecedent and consequent of the intermediate rule. The results for moving to the left direction (from a_1's point of view) can be computed as

$$b_1^* = \frac{b_1^i + b_2^i + b_3^i}{3} - \frac{2(a_3^* - a_1^*)(b_3^i - b_1^i)}{3(a_3^i - a_1^i)} + \frac{(a_3^* - a_2^*)(b_3^i - b_2^i)}{3(a_3^i - a_2^i)}, \tag{6.9}$$

$$b_2^* = \frac{b_1^i + b_2^i + b_3^i}{3} + \frac{(a_3^* - a_1^*)(b_3^i - b_1^i)}{3(a_3^i - a_1^i)} - \frac{2(a_3^* - a_2^*)(b_3^i - b_2^i)}{3(a_3^i - a_2^i)}, \tag{6.10}$$

$$b_3^* = \frac{b_1^i + b_2^i + b_3^i}{3} + \frac{(a_3^* - a_1^*)(b_3^i - b_1^i)}{3(a_3^i - a_1^i)} + \frac{(a_3^* - a_2^*)(b_3^i - b_2^i)}{3(a_3^i - a_2^i)}. \tag{6.11}$$

Thus,

$$b_3^{*c} = b_3^* - b_2^* = \frac{b_3^i - b_2^i}{a_3^i - a_2^i}(a_3^* - a_2^*) = \lambda^{OHS}(a_3^{*c}) + \delta^{OHS}, \tag{6.12}$$

where $\lambda^{OHS} = \frac{b_3^i - b_2^i}{a_3^i - a_2^i}$ and $\delta^{OHS} = 0$. However, for moving to the right direction, the results can be written as

$$b_1^* = \frac{b_1^i + b_2^i + b_3^i}{3} - \frac{(a_3^* - a_1^*)(b_3^i - b_1^i)}{3(a_3^i - a_1^i)} - \frac{(a_2^* - a_1^*)(b_2^i - b_1^i)}{3(a_2^i - a_1^i)}, \tag{6.13}$$

$$b_2^* = \frac{b_1^i + b_2^i + b_3^i}{3} - \frac{(a_3^* - a_1^*)(b_3^i - b_1^i)}{3(a_3^i - a_1^i)} + \frac{2(a_2^* - a_1^*)(b_2^i - b_1^i)}{3(a_2^i - a_1^i),} \tag{6.14}$$

$$b_3^* = \frac{b_1^i + b_2^i + b_3^i}{3} + \frac{2(a_3^* - a_1^*)(b_3^i - b_1^i)}{3(a_3^i - a_1^i)} - \frac{(a_2^* - a_1^*)(b_2^i - b_1^i)}{3(a_2^i - a_1^i)}. \tag{6.15}$$

Similarly,

$$b_3^{*c} = b_3^* - b_2^* = \frac{(a_3^* - a_1^*)(b_3^i - b_1^i)}{a_3^i - a_1^i} - \frac{(a_2^* - a_1^*)(b_2^i - b_1^i)}{a_2^i - a_1^i} = \lambda^{OHS}(a_3^{*c}) + \delta^{OHS}, \tag{6.16}$$

where $\lambda^{OHS} = \frac{b_3^i - b_1^i}{a_3^i - a_1^i}$ and $\delta^{OHS} = \lambda^{OHS}(-a_{13}^c) + b_{13}^c$.

If no move transformation is required, the results can be merely generated according to (4.14) - (4.16) as follows.

$$b_1^* = \frac{b_1^i(1+2s) + b_2^i(1-s) + b_3^i(1-s)}{3}, \tag{6.17}$$

$$b_2^* = \frac{b_1^i(1-s) + b_2^i(1+2s) + b_3^i(1-s)}{3}, \tag{6.18}$$

$$b_3^* = \frac{b_1^i(1-s) + b_2^i(1-s) + b_3^i(1+2s)}{3}. \tag{6.19}$$

Therefore,

$$b_3^{*c} = b_3^* - b_2^* = s(b_3^i - b_2^i) = \frac{(a_3^* - a_1^*)(b_3^i - b_2^i)}{a_3^i - a_1^i} = \lambda^{OHS}(a_3^{*c}) + \delta^{OHS}, \tag{6.20}$$

where $\lambda^{OHS} = \frac{b_3^i - b_2^i}{a_3^i - a_1^i}$ and $\delta^{OHS} = \lambda^{OHS}(-a_{13}^c) + b_{13}^c$.

For the enhanced HS method, the scale criteria $\mathbb{SL}$ and $\mathbb{SR}$ are calculated as

$$\mathbb{SL} = \frac{a_2^* - a_1^*}{a_2^i - a_1^i}, \tag{6.21}$$

$$\mathbb{SR} = \frac{a_3^* - a_2^*}{a_3^i - a_2^i}. \tag{6.22}$$

Thus, the scale rate of B^i is

$$s_b = \frac{\mathbb{SL}(b_2^i - b_1^i) + \mathbb{SR}(b_3^i - b_2^i)}{b_3^i - b_1^i}. \tag{6.23}$$

The enhanced scale and move transformations impose the following constraints:

$$\begin{cases} \frac{a_2^* - a_1^*}{a_3^* - a_2^*} \frac{a_3^i - a_2^i}{a_2^i - a_1^i} = \frac{b_2^* - b_1^*}{b_3^* - b_2^*} \frac{b_3^i - b_2^i}{b_2^i - b_1^i} \\ b_3^* - b_1^* = s_b(b_3^i - b_1^i) \end{cases}$$

The above equations can be reformed to

$$b_3^* - b_2^* = \frac{D}{E}(a_3^* - a_2^*)^2 + (D + \frac{C}{E})(a_3^* - a_2^*) + C, \tag{6.24}$$

where $C = \frac{a_2^* - a_1^*}{a_2^i - a_1^i}(b_2^i - b_1^i)$, $D = \frac{b_3^i - b_2^i}{a_3^i - a_2^i}$ and $E = \frac{(a_2^* - a_1^*)(a_3^i - a_2^i)(b_2^i - b_1^i)}{(a_2^i - a_1^i)(b_3^i - b_2^i)}$.

The results are shown in Fig. 6.1 (see [TB00]). In this figure, two coordinate systems are simultaneously used to demonstrate the dependency of fuzziness of the conclusion on the observation. One is $X \times Y$ that is the Cartesian product space of fuzzy sets and the other is positioned at $O(cp(A^*), cp(B^*))$, or $O(a_2^*, b_2^*)$. the solid lines show the fuzziness of conclusion with respect to the fuzziness of observation. In fact, the calculation of $cp(B^*)$ is independent of all the fuzzy interpolation methods concerned except for the proposed OHS (original HS) and EHS (enhanced HS) methods. The origin O of the inner coordinate system moves on the straight line $\overline{PR}$ from $P(a_{12}, b_{12})$ to $R(a_{22}, b_{22})$ due to the fact that the observation A^* lies between A_1 and A_2.

The straight line $\overline{KH}$ ($K(a_{13}, b_{13})$ and $H(a_{23}, b_{23})$) represents the function of (6.2) for KH method. It indicates that if the fuzziness of observation is less than a threshold

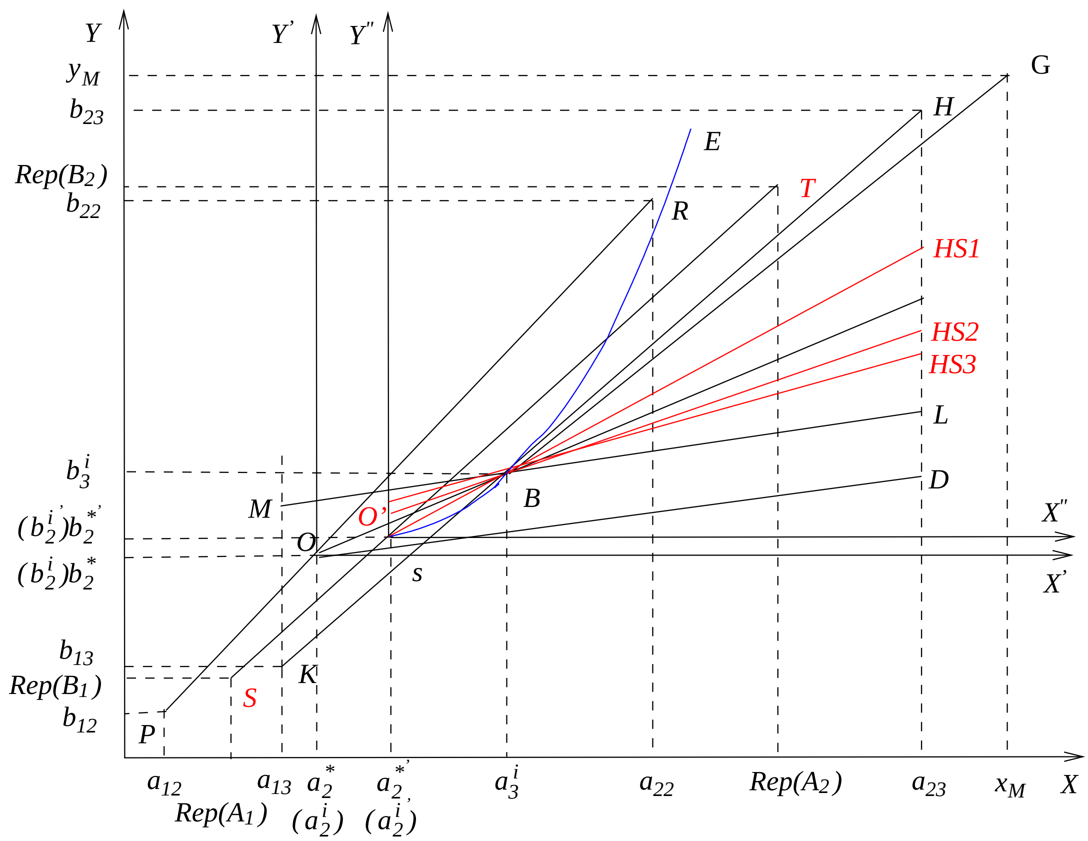

Figure 6.1: The dependency of the fuzziness of conclusion on that of observation

$s > 0$, which can be determined by the work of [KHM00], then subnormal conclusion is obtained using the KH method.

For the modified KH method, the function f_f^{MKH} can be determined by point $M(a_{13}, cp(B^*) + (b_{13} - b_{12}))$ and $L(a_{23}, cp(B^*) + (b_{23} - b_{22}))$ as these two points must be passed. $\overline{ML}$ has a slope of $\frac{(b_{23}-b_{13})-(b_{22}-b_{12})}{a_{23}-a_{13}}$ and involves fixed distances between M, L and axis X'.

The KHG method (function (6.5)) is represented by a straight line $\overline{OD}$ with slope $(b_{22} - b_{21})/(a_{22} - a_{21})$. Note that it cannot be interpreted when $a_{22} - a_{21} = 0$ and $b_{22} - b_{21} > 0$, but otherwise the conclusion is always a normal and valid fuzzy set.

The general fuzzy interpolation method (function (6.6)) yields two straight lines $\overline{OB}$ and $\overline{BG}$, where $B(a_3^i, b_3^i)$ is determined by $\frac{\overline{PO}}{\overline{PR}} = \frac{\overline{KB}}{\overline{KH}}$, and $G(x_M, y_M)$ involves the maximal values of the domains of X and Y. The general interpolation always obtains

NVF fuzzy sets as $\overline{OBG}$ never crosses the axis X'.

Now consider the original HS method. Line $\overline{ST}$ connects the points $S(Rep(A_1), Rep(B_1))$ and $T(Rep(A_2), Rep(B_2))$. Thus, point $O'(Rep(A^*), Rep(B^*))$ on this line has property $\frac{Rep(A^*)-Rep(A_1)}{Rep(A_2)-Rep(A_1)} = \frac{Rep(B^*)-Rep(B_1)}{Rep(B_2)-Rep(B_1)} = \lambda$, where $\lambda = \frac{\overline{PO}}{\overline{PR}}$. Initially, suppose $A^{*'} = (a_1^{*'}, a_2^{*'}, a_3^{*'})$ (used to replace the observation for the HS methods to distinguish it from the previously used observation A^*) is a singleton input, $a_2^{*'} = a_2^{i'} = a_3^{*'}, b_2^{*'} = b_2^{i'} = b_3^{*'}$. In the case when moving to the left (from $a_2^{*'}$'s point of view), the function of b_3^{*c} with respect to a_3^{*c} can be represented by line $\overline{HS1}$ which passes O' and B, with slope $\theta_1 = \frac{b_3^i - b_2^i}{a_3^i - a_2^i}$. When moving to the right, the fuzziness dependency can be represented by line $\overline{HS2}$ which passes B with slope $\theta_2 = \frac{b_3^i - b_1^i}{a_3^i - a_1^i}$. Similarly when no move is required, it can be represented by line $\overline{HS3}$ which passes B with slope $\theta_3 = \frac{b_3^i - b_2^i}{a_3^i - a_1^i}$.

Three points are worth mentioning for Figure 6.1.

1. It is different from the previously shown approaches that the point O' is actually not fixed. In particular, when $a_2^{*'} = a_2^{i'} = a_3^{*'}$, $b_2^{*'} = b_2^{i'} = b_3^{*'}$ holds. If $a_3^{*'}$ increases, $b_2^{*'}$ will change as well. So the Y'' axis indicates only the difference of $b_3^* - b_2^*$ (rather than the real values of b_2^* and b_3^*).

2. If the centre point value is chosen to be defined as RV,O' becomes the same as O, thus line $\overline{HS1}$ partially coincides with $\overline{OBG}$. In other words, when moving to the left, the a_3^* (less than a_3^i) generated from $\overline{HS1}$ is the same as that from the general interpolative reasoning method.

3. $\theta_3 \leq \theta_2$ and $\theta_3 \leq \theta_1$ always hold.

Finally, the enhanced HS interpolation method is shown by line $\overline{O'BE}$ according to function (6.24). It is a second degree polynomial function which passes points O' and B. It may be either valid or concave.

6.2.2 Preservation of Piecewise Linearity

Preservation of piecewise linearity is an essential property which reflects how good an interpolative reasoning method handles the points between two consecutive characteristic points. If the piecewise linearity is preserved, it is safe to merely consider the characteristic points rather than the infinite pairs of points (generated from an infinite number of α-cut levels). The preservation of piecewise linearity has been investigated in the work of [KHM00, TB00]. In both cases, they slightly deviate from the calculated linear fuzzy rule interpolations with some error bounds provided. This subsection first shows that the original HS method preserves the piecewise linearity only in interpolations involving triangular fuzzy sets, and then proves that the enhanced HS method preserves this property in interpolations involving arbitrary polygonal fuzzy sets.

6.2.2.1 Original HS Method

Consider a triangular-based fuzzy interpolation as shown in Fig. 6.2. Given rules $A_1 \Rightarrow B_1$, $A_2 \Rightarrow B_2$ and an observation A^*, the task is to determine B^*. The difference here is that all fuzzy sets A_1, A_2, A^*, B_1 and B_2 have five characteristic points rather than three. That is, each fuzzy set has two additional artificial characteristic points.

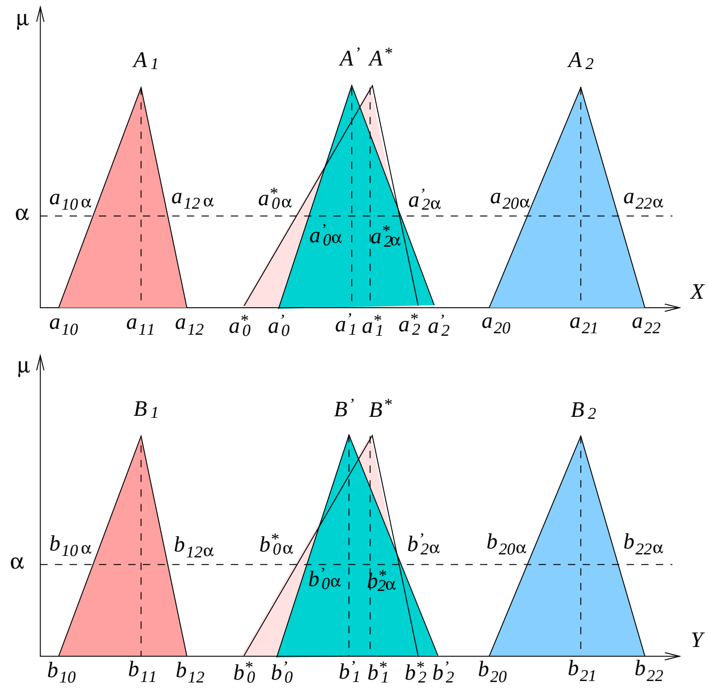

Figure 6.2: Interpolation involving triangular sets but with 5 characteristic points

The first step for OHS interpolation is to construct the intermediate rule $A' \Rightarrow B'$.

It can be shown in the following that both A' and B' are two triangular fuzzy sets, with each having 5 characteristic points.

proof 10 *As*

$$\begin{cases} a_{10\alpha} = (1-\alpha)a_{10} + \alpha a_{11} \\ a_{20\alpha} = (1-\alpha)a_{20} + \alpha a_{21} \\ a'_{0\alpha} = (1-\alpha)a'_0 + \alpha a'_1 \end{cases} \tag{6.25}$$

and

$$\begin{cases} a'_0 = (1-\lambda)a_{10} + \lambda a_{20} \\ a'_1 = (1-\lambda)a_{11} + \lambda a_{21} \end{cases} \tag{6.26}$$

Therefore

$$a'_{0\alpha} = (1-\alpha)[(1-\lambda)a_{10} + \lambda a_{20}] + \alpha[(1-\lambda)a_{11} + \lambda a_{21}] \tag{6.27}$$

$$= (1-\lambda)[(1-\alpha)a_{10} + \alpha a_{11}] + \lambda[(1-\alpha)a_{20} + \alpha a_{21}] \tag{6.28}$$

$$= (1-\lambda)a_{10\alpha} + \lambda a_{20\alpha}. \tag{6.29}$$

The point $a'_{0\alpha}$ which is the interacted point of line $\overline{a'_0 a'_1}$ and the α-cut level is also the interpolated point between $a_{10\alpha}$ and $a_{20\alpha}$, it is thereby an artificial characteristic point in A'. Similarly, $a'_{2\alpha}$ is an artificial characteristic point. So A' is a triangular set with 5 characteristic points. This proof also applies to B'.

Now the scale transformation will scale A' to A^s (as shown in Fig. 6.3) which has the same support lengths as those of A^*. As A' and A^* are both triangular sets, the scale rates for the supports of the bottom and the α-cut level remain the same, say, s. A^s can be computed from A' using the five equations imposed to the scale transformation:

$$\begin{cases} Rep(A^s) = Rep(A') \\ \frac{a^s_2 - a^s_0}{a'_2 - a'_0} = s \\ \frac{a^s_{2\alpha} - a^s_{0\alpha}}{a'_{2\alpha} - a'_{0\alpha}} = s \\ \frac{a^s_1 - a^s_{0\alpha}}{a^s_{2\alpha} - a^s_1} = \frac{a'_1 - a'_{0\alpha}}{a'_{2\alpha} - a'_1} \\ \frac{a^s_{0\alpha} - a^s_0}{a^s_2 - a^s_{2\alpha}} = \frac{a'_{0\alpha} - a'_0}{a'_2 - a'_{2\alpha}} \end{cases} \tag{6.30}$$

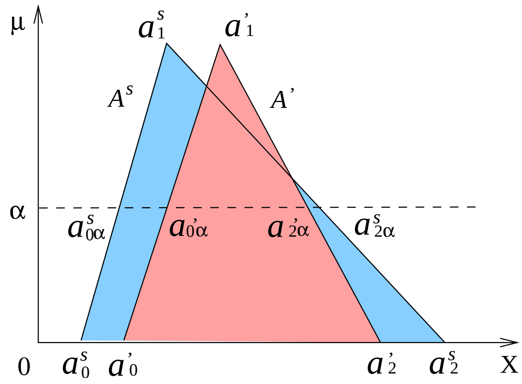

Figure 6.3: Preservation of piecewise linearity in triangular cases for original scale transformation

The scaled fuzzy set is therefore the same as that generated using only three characteristic points. That is, the scaled fuzzy set is a triangular fuzzy set but with 5 characteristic points. Similarly, a triangular fuzzy set B^s with 5 characteristic points can be obtained.

The preservation of piecewise linearity in move transformation is shown in Fig. 6.4. It might appear that in this case two sub-moves should be applied – the first moves a_0^s to a_0^*, and the second moves $a_{0\alpha}^{s'}$ to $a_{0\alpha}^*$, where $a_{0\alpha}^{s'}$ is the position for $a_{0\alpha}^s$ after the first move. However, that is not the case. In fact, only one move is required to transform A^s (with 5 characteristic points) to A^* (also with 5 characteristic points). This move simultaneously moves a_0^s, $a_{0\alpha}^s$ and a_1^s to the desired positions (a_0^*, $a_{0\alpha}^*$ and a_1^*) respectively, according to the move transformation.

In summary, the original HS preserves piecewise linearity in performing scale and move transformations, resulting in the preservation of that property in fuzzy interpolation. Unfortunately, this property cannot be preserved when the original HS method is applied to fuzzy interpolations involving fuzzy membership functions other than triangular sets. This is due to the way of calculating scale rates in scale transformation. For example, consider the scale transformation of trapezoidal fuzzy membership functions with each having two artificial characteristic points (see Fig. 6.5). The task is to calculate the scale rates for B' based on the transformation from A' to A^s. Let $a'_{0\alpha}$, $a'_{3\alpha}$, $a_{0\alpha}^s$

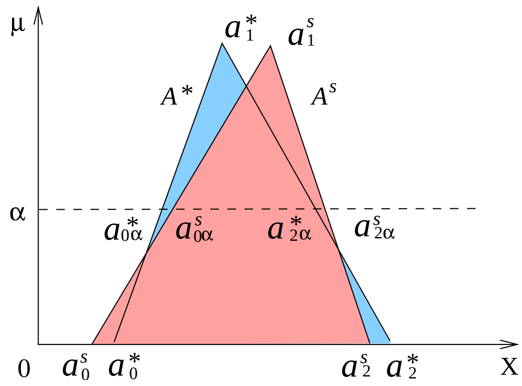

Figure 6.4: Preservation of piecewise linearity in triangular cases for original move transformation

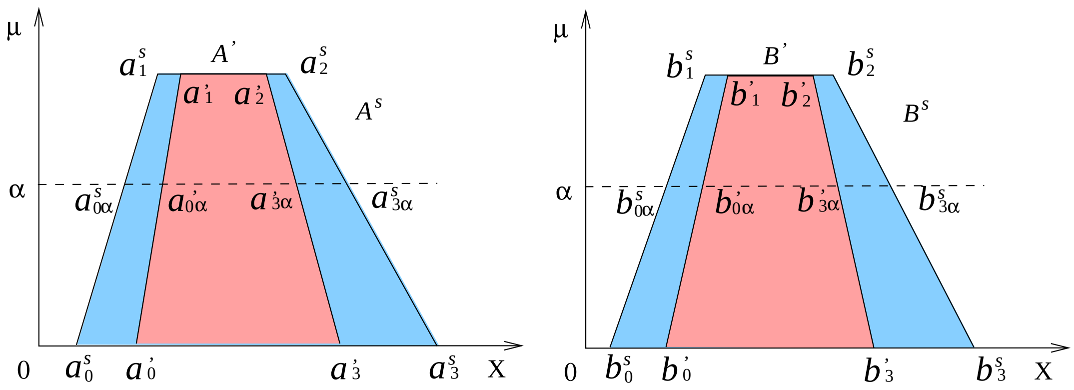

Figure 6.5: No preservation of piecewise linearity in trapezoidal cases for original scale transformation

and $a^s_{3\alpha}$ be artificial characteristic points, then

$$a'_{3\alpha} - a'_{0\alpha} = \alpha(a'_2 - a'_1) + (1-\alpha)(a'_3 - a'_2), \tag{6.31}$$

$$a^s_{3\alpha} - a^s_{0\alpha} = \alpha(a^s_2 - a^s_1) + (1-\alpha)(a^s_3 - a^s_2). \tag{6.32}$$

Let the scale rates for the bottom, middle and top supports of A' be denoted as s_0, s_α

and s_1 respectively. Then,

$$s_\alpha = \frac{a^s_{3\alpha} - a^s_{0\alpha}}{a'_{3\alpha} - a'_{0\alpha}} \tag{6.33}$$

$$= \frac{\alpha(a^s_2 - a^s_1) + (1-\alpha)(a^s_3 - a^s_2)}{\alpha(a'_2 - a'_1) + (1-\alpha)(a'_3 - a'_2)} \tag{6.34}$$

$$= \frac{\alpha s_1(a'_2 - a'_1) + (1-\alpha)s_0(a'_3 - a'_0)}{\alpha(a'_2 - a'_1) + (1-\alpha)(a'_3 - a'_0)}. \tag{6.35}$$

It can be reformed as

$$\frac{s_1 - s_\alpha}{s_\alpha - s_0} = \frac{(1-\alpha)(a'_3 - a'_0)}{\alpha(a'_2 - a'_1)}. \tag{6.36}$$

Consider the simple case of calculating scale rates, i.e., when $s_0 > s_\alpha$, then $s_0 > s_\alpha > s_1$ holds according to Fig. 6.5. Let the scale rates for the bottom, middle and top supports of B' be denoted as s'_0, s'_α and s'_1 respectively, according to the way of calculating the scale rates (see (5.34)), $s'_0 = s_0 > s'_\alpha = s_\alpha > s'_1 = s_1$. Given that $b'_{0\alpha}$ and $b'_{3\alpha}$ are artificial characteristic points, according to (6.36), the following must hold so that $b^s_{0\alpha}$ and $b^s_{3\alpha}$ will be artificial:

$$\frac{s'_1 - s'_\alpha}{s'_\alpha - s'_0} = \frac{(1-\alpha)(b'_3 - b'_0)}{\alpha(b'_2 - b'_1)}. \tag{6.37}$$

However, this is not true unless in the special situation where $\frac{a'_3 - a'_0}{a'_2 - a'_1} = \frac{b'_3 - b'_0}{b'_2 - b'_1}$. Thus the piecewise linearity cannot be always preserved in the trapezoidal cases.

6.2.2.2 Enhanced HS Method

The enhanced HS method preserves the piecewise linearity in interpolations involving arbitrary polygonal fuzzy membership functions. This subsection proves this in both scale and move transformations.

Fig. 6.6 illustrates the scale transformation in a trapezoidal case with six characteristic points for each fuzzy set. Suppose that $a^s_{0\alpha}$, $a^s_{3\alpha}$, $a'_{0\alpha}$, $a'_{3\alpha}$, $b'_{0\alpha}$ and $b'_{3\alpha}$ are artificial characteristic points. If B^s is transformed from B' using the same similarity calculated from A' to A^s, the question is whether $b^s_{0\alpha}$ and $b^s_{3\alpha}$ remain artificial. According to the enhanced scale method,

$$\frac{a^s_1 - a^s_{0\alpha}}{a'_1 - a'_{0\alpha}} = \frac{b^s_1 - b^s_{0\alpha}}{b'_1 - b'_{0\alpha}}, \tag{6.38}$$

$$\frac{a^s_{0\alpha} - a^s_0}{a'_{0\alpha} - a'_0} = \frac{b^s_{0\alpha} - b^s_0}{b'_{0\alpha} - b'_0}. \tag{6.39}$$

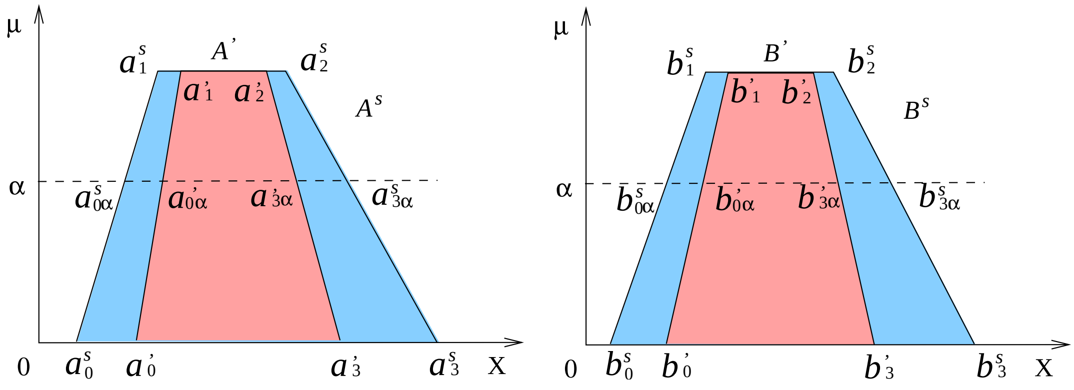

Figure 6.6: Preservation of piecewise linearity in enhanced scale transformation

Also, as $a'_{0\alpha}$ and $a^s_{0\alpha}$ are two artificial characteristic points, then

$$\frac{a^s_1 - a^s_{0\alpha}}{a'_1 - a'_{0\alpha}} = \frac{a^s_{0\alpha} - a^s_0}{a'_{0\alpha} - a'_0}. \tag{6.40}$$

From (6.38), (6.39) and (6.40),

$$\frac{b^s_1 - b^s_{0\alpha}}{b'_1 - b'_{0\alpha}} = \frac{b^s_{0\alpha} - b^s_0}{b'_{0\alpha} - b'_0}. \tag{6.41}$$

From (6.41) and the fact that $b'_{0\alpha}$ is an artificial characteristic point, it can be concluded that $b^s_{0\alpha}$ must be artificial. That is, B^s is piecewise linear in the left slope. Similarly, B^s is piecewise linear in the right slope. Thus the proposed method preserves the piecewise linearity in the scale transformations.

The proof is based on the trapezoidal cases and it in fact shows that the piecewise linearity is retained between two α-cut levels. For the scale transformation case involving arbitrary polygonal fuzzy membership functions, the proof applies between any two consecutive α-cut levels, resulting in the preservation of piecewise linearity in this case.

Now consider the move transformation which is shown in Fig. 6.7. Given A^s and A^* which have the same RV and the same lengthes of top, middle and bottom supports respectively, the task is to move B^s to obtain B^* using the same similarity between A^s and A^*. According to the enhanced move transformation,

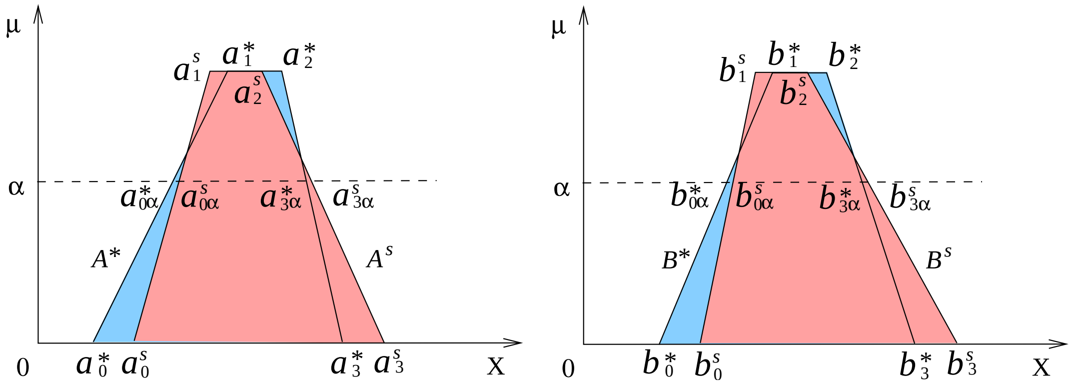

Figure 6.7: Preservation of piecewise linearity in enhanced move transformation

$$\frac{a_1^* - a_{0\alpha}^*}{a_{3\alpha}^* - a_2^*}\frac{a_{3\alpha}^s - a_2^s}{a_1^s - a_{0\alpha}^s} = \frac{b_1^* - b_{0\alpha}^*}{b_{3\alpha}^* - b_2^*}\frac{b_{3\alpha}^s - b_2^s}{b_1^s - b_{0\alpha}^s}, \tag{6.42}$$

$$\frac{a_{0\alpha}^* - a_0^*}{a_3^* - a_{3\alpha}^*}\frac{a_3^s - a_{3\alpha}^s}{a_{0\alpha}^s - a_0^s} = \frac{b_{0\alpha}^* - b_0^*}{b_3^* - b_{3\alpha}^*}\frac{b_3^s - b_{3\alpha}^s}{b_{0\alpha}^s - b_0^s}. \tag{6.43}$$

Assume that $a_{0\alpha}^s$, $a_{0\alpha}^*$, $a_{3\alpha}^s$ and $a_{3\alpha}^*$ are arbitrary characteristic points, then

$$\frac{a_1^* - a_{0\alpha}^*}{a_{3\alpha}^* - a_2^*}\frac{a_{3\alpha}^s - a_2^s}{a_1^s - a_{0\alpha}^s} = \frac{a_{0\alpha}^* - a_0^*}{a_3^* - a_{3\alpha}^*}\frac{a_3^s - a_{3\alpha}^s}{a_{0\alpha}^s - a_0^s}. \tag{6.44}$$

From (6.42), (6.43) and (6.44),

$$\frac{b_1^* - b_{0\alpha}^*}{b_{3\alpha}^* - b_2^*}\frac{b_{3\alpha}^s - b_2^s}{b_1^s - b_{0\alpha}^s} = \frac{b_{0\alpha}^* - b_0^*}{b_3^* - b_{3\alpha}^*}\frac{b_3^s - b_{3\alpha}^s}{b_{0\alpha}^s - b_0^s}. \tag{6.45}$$

Given that $b_{0\alpha}^s$ and $b_{3\alpha}^s$ are artificial characteristic points, it follows that

$$\frac{b_{3\alpha}^s - b_2^s}{b_1^s - b_{0\alpha}^s} = \frac{b_3^s - b_{3\alpha}^s}{b_{0\alpha}^s - b_0^s}. \tag{6.46}$$

From (6.45) and (6.46),

$$\frac{b_1^* - b_{0\alpha}^*}{b_{3\alpha}^* - b_2^*} = \frac{b_{0\alpha}^* - b_0^*}{b_3^* - b_{3\alpha}^*}. \tag{6.47}$$

As $b_2^s - b_1^s = b_2^* - b_1^*$, $b_{3\alpha}^s - b_{0\alpha}^s = b_{3\alpha}^* - b_{0\alpha}^*$ and $b_3^s - b_0^s = b_3^* - b_0^*$, it can be concluded that $b_{0\alpha}^*$ and $b_{3\alpha}^*$ are artificial.

Again, although the proof is based on the trapezoidal cases, it applies between any two consecutive α-cut levels in arbitrary polygonal fuzzy memberships, resulting in the preservation of piecewise linearity in the move transformation involving those fuzzy membership functions.

It can be proven that the construction of the intermediate fuzzy rule preserves piecewise linearity. Now the scale and move transformations has proven to preserve piecewise linearity, property 1 as shown below can be achieved by the enhanced HS method:

Property 1 *The interpolation of non-characteristic points which lie between two characteristic points generates non-characteristic points which still lie between the two interpolated characteristic points.*

Property 1 points out that only characteristic points affect the interpolated results using the EHS method. Non-characteristic points can be safely ignored as they are still non-characteristic in the reasoning results.

If the representative value of a fuzzy set keeps the same when more artificial characteristic points are considered in the EHS interpolation, then the following property holds:

Property 2 *The interpolation of the same fuzzy sets but with additional artificial characteristic points leads to the same result if the representative values of these fuzzy sets (with or without additional artificial characteristic points) are the same.*

The work of [YK98, YK00] represents each fuzzy set with n characteristic points as a point in an n-dimensional Cartesian space, thus a fuzzy interpolation problem becomes a high dimensional interpolation problem. Since the EHS interpolation method is capable of handling fuzzy interpolation involving infinite points (finite characteristic points plus infinite non-characteristic points), it may provide a solution to the interpolation problem within a very high dimensional Cartesian space.

6.2.2.3 Illustrative examples for the maintenance of piecewise linearity

In this section, the use of the average RV, compatible RV, weighted average RV and centre-of-core RV to conduct fuzzy interpolations is demonstrated and the results between the original HS and enhanced HS methods are compared. For simplicity, both examples discussed below concern the interpolation between two adjacent rules $A_1 \Rightarrow$

B_1 and $A_2 \Rightarrow B_2$. In order to verify the piecewise linearity property, additional "characteristic" points are added in the examples.

Table 6.1 shows values of the rule attributes and observations. Table 6.2 and Table 6.3 show the interpolated results using different RV definitions for the OHS method and EHS respectively. These results are also illustrated in Fig. 6.8 and Fig. 6.9. As can be seen, the original HS method satisfies property 1 only in triangular cases while the enhanced HS satisfies that in all cases. In particular, the latter further holds property 2 when the compatible and centre core representative values are used. As a comparison, the results of the KH method is also given in Fig. 6.8. It satisfies neither property 1 nor property 2.

Table 6.1: Attribute and observation values

	Triangular	Triangular (5 points)	Hexagonal	Hexagonal (8 points)
A_1	$(0,5,6)$	$(0,2.5,5,5.5,6)$	$(0,1,3,$ $4,5,5.5)$	$(0,0.5,1,2,3,$ $4,4.5,5,5.25,5.5)$
A_2	$(11,13,14)$	$(11,12,13,13.5,14)$	$(11,11.5,12,$ $13,13.5,14)$	$(11,11.25,11.5,11.75,12,$ $13,13.25,13.5,13.75,14)$
A^*	$(7,8,9)$	$(7,7.5,8,8.5,9)$	$(6,6.5,7,$ $9,10,10.5)$	$(6,6.25,6.5,6.75,7,$ $9,9.5,10,10.25,10.5)$
B_1	$(0,2,4)$	$(0,1,2,3,4)$	$(0,0.5,1,$ $3,4,4.5)$	$(0,0.25,0.5,0.75,1,$ $3,3.5,4,4.25,4.5)$
B_2	$(10,11,13)$	$(10,10.5,11,12,13)$	$(10.5,11,12,$ $13,13.5,14)$	$(10.5,10.75,11,11.5,12,$ $13,13.25,13.5,13.75,14)$

6.2.3 Computational Complexity

In this section, the time complexity with respect to n (the number of characteristic points for fuzzy sets involved) is estimated for interpolative reasoning methods, including the KH, the general interpolation, the modified KH, the original HS and the enhanced HS. To have a uniform platform for comparison, two simplest rules $A_1 \Rightarrow B_1$ and $A_2 \Rightarrow B_2$ and an observation A^* are used here.

The KH interpolative reasoning method can be written in pseudo code as shown

Table 6.2: Results of original HS method by using different RVs

	Triangular	Triangular (5 points)	Hexagonal	Hexagonal (8 points)
Average	(5.84, 6.26, 7.38)	(5.76, 5.97, 6.18, 6.74, 7.3)	(5.64, 5.98, 6.29, 8.63, 9.46, 9.93)	(5.64, 5.82, 5.98, 6.14, 6.28, 8.64, 9.05, 9.47, 9.71, 9.94)
Compatible	(5.84, 6.26, 7.38)	(5.84, 6.05, 6.26, 6.82, 7.38)	(5.67, 6.01, 6.33, 8.66, 9.50, 9.97)	(5.66, 5.84, 6.01, 6.16, 6.31 8.67, 9.08, 9.50, 9.74, 9.97)
Weighted Average	(5.63, 6.06, 7.16)	(5.63, 5.85, 6.06, 6.61, 7.16)	(5.61, 5.95, 6.26, 8.59, 9.42, 9.89)	(5.62, 5.80, 5.96, 6.11, 6.26, 8.62, 9.02, 9.44, 9.68, 9.91)
Centre of Core	(4.96, 5.38, 6.44)	(4.96, 5.17, 5.38, 5.91, 6.44)	(5.47, 5.79, 6.08 8.42, 9.23, 9.70)	(5.46, 5.64, 5.81, 5.95, 6.07, 8.43, 8.83, 9.25, 9.47, 9.70)

Table 6.3: Results of enhanced HS method by using different RVs

	Triangular	Triangular (5 points)	Hexagonal	Hexagonal (8 points)
Average	(5.54, 5.97, 7.97)	(5.49, 5.70, 5.92, 6.92, 7.92)	(5.28, 5.62, 5.94, 8.86, 9.86, 10.36)	(5.28, 5.45, 5.62, 5.79, 5.95, 8.87, 9.37, 9.87, 10.12, 10.37)
Compatible	(5.54, 5.97, 7.97)	(5.54, 5.76, 5.97, 6.97, 7.97)	(5.30, 5.65, 5.97, 8.88, 9.88, 10.38)	(5.30, 5.47, 5.65, 5.81, 5.97 8.88, 9.38, 9.88, 10.13, 10.38)
Weighted Average	(5.41, 5.83, 7.83)	(5.41, 5.62, 5.83, 6.83, 7.83)	(5.25, 5.59, 5.91, 8.85, 9.85, 10.35)	(5.26, 5.43, 5.60, 5.72, 5.92, 8.85, , 9.35, 9.85, 10.10, 10.35)
Centre of Core	(4.96, 5.38, 7.38)	(4.96, 5.17, 5.38, 6.38, 7.38)	(5.12, 5.45, 5.75 8.75, 9.75, 10.25)	(5.12, 5.28, 5.45, 5.60, 5.75, 8.75, 9.25, 9.75, 10.00, 10.25)

in Algorithm 1, where $\lambda[i]$ (see [KH93a, KH93c]) measures the important impact of

Algorithm 1 KH interpolation

Input: Polygonal fuzzy set A^* with n characteristic points

Output: Polygonal fuzzy set B^* with n characteristic points

```
for i = 0 to n − 1 do
    calculate λ[i] from the ith points of A_1, A_2 and A*
    calculate the ith point of B*
end for
```

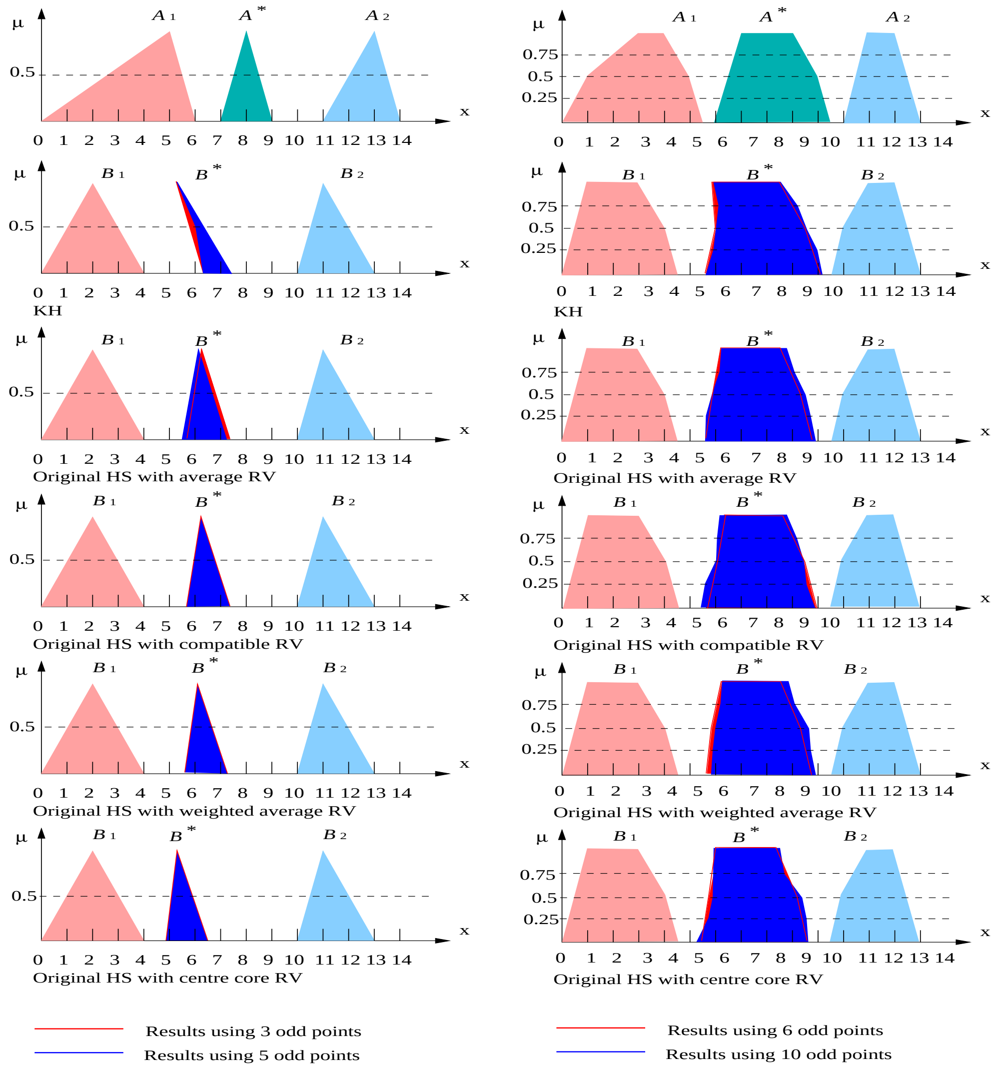

Figure 6.8: Examples of piecewise linearity for KH and original HS method

the ith characteristic point of A_2 (versus that of A_1) upon the ith point of A^*. The computation time in line 2 has a unit time of $O(1)$ as it simply consists of several basic calculations (no loop is involved). Similarly, line 3 costs another unit time of $O(1)$. The total computation time for line 2 and line 3 thus is $O(1)$. Since line 2 and line 3 are executed once for every loop from 0 to $(n-1)$, the total computation time for this algorithm is $O(n)$.

The general interpolation [BGK96, BGK95] method is described in Algorithm 2.

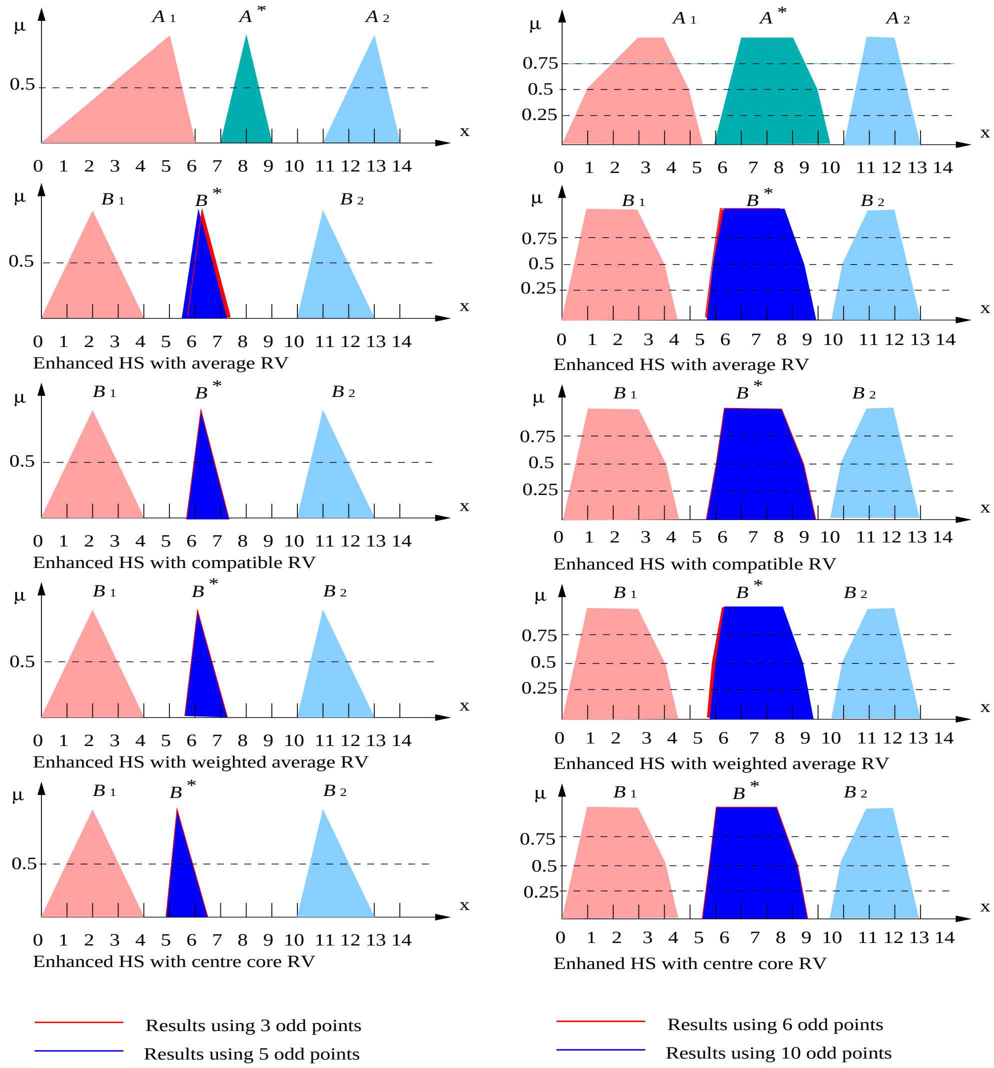

Figure 6.9: Examples of piecewise linearity for enhanced HS method

Line 1 costs $O(n)$ computation time as it has a loop running through all the characteristic points. As line 3 costs $O(1)$, the loop of line 2 costs $O(n)$ time. In total, $O(n)+O(n)=O(n)$ is the time complexity for this method.

Now consider the modified KH method [TB00] as shown in Algorithm 3. Line 1 costs $O(n)$ computation time as it actually has a loop running through all the characteristic points. Line 3 costs the other $O(n)$ as it simply reverses the operation in line 1. According to the KH interpolative reasoning method, line 2 takes $O(n)$ time. So the

Algorithm 2 General interpolation

Input: Polygonal fuzzy set A^* with n characteristic points

Output: Polygonal fuzzy set B^* with n characteristic points

1: Compute intermediate fuzzy sets A' and B'
2: **for** $i = 0$ to $n-1$ **do**
3: calculate the ith point of B^* according to those of A^*, A' and B'
4: **end for**

Algorithm 3 Modified KH interpolation

Input: Polygonal fuzzy set A^* with n characteristic points

Output: Polygonal fuzzy set B^* with n characteristic points

1: Convert the coordinate system to a different one
2: KH interpolation
3: Convert the coordinate system back to the original one

total time complexity is $O(n)+O(n)+O(n)=O(n)$.

The proposed original HS interpolative reasoning method is shown in Algorithm 4, where *calM* and *move* are two procedures which are called by the OHS interpolation

Algorithm 4 Original HS interpolation

Input: Polygonal fuzzy set A^* with n characteristic points

Output: Polygonal fuzzy set B^* with n characteristic points

1: Compute intermediate fuzzy sets A' and B'
2: Compute scale rates from A' to A^*
3: Scale A' with scale rates calculated by step 2 to generate set A^s
4: Compute scale rates applied to B'
5: Scale B' with scale rates calculated by step 4 to generate set B^s
6: $A^m = A^s$; $B^m = B^s$;
7: **for** $i = 0$ to $\lceil \frac{n}{2} \rceil - 2$ **do**
8: $MoveRatio[i] = calM(A^m, A^*, i)$
9: $A^m = move(A^m, i, moveRatio[i])$
10: $B^m = move(B^m, i, moveRatio[i])$
11: **end for**

algorithm. They are written in Algorithm 5 and Algorithm 6 respectively, where $a_j^{(i)}$

Algorithm 5 $CalM(A,B,i)$: calculate move ratio of the ith support level from A to B

Input: Two polygonal fuzzy sets A and B (with each having n characteristic points) and support level i

Output: Move ratio of support level i of A

$sum1 = 0$, $sum2 = 0$
for $j = i$ to $\lceil \frac{n}{2} \rceil - 1$ **do**
    $sum1 = sum1 + a_j^{(i-1)}(w_j + w_{n-1-j})$
    $sum2 = sum2 + (w_j + w_{n-1-j})$
end for
$extX = min\{\frac{sum1}{sum2} - a_i^{(i-1)}, a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}\}$
Compute move ratio according to $extX$

Algorithm 6 $Move(A,i,MoveRatio)$: move the ith support of A with the specified $MoveRatio$

Input: Polygonal fuzzy set A with n characteristic points, support level i and the specified move ratio $MoveRatio$

Output: Moved fuzzy set A^m

$sum1 = 0$, $sum2 = 0$
for $j = i$ to $\lceil \frac{n}{2} \rceil - 1$ **do**
    $sum1 = sum1 + a_j^{(i-1)}(w_j + w_{n-1-j})$
    $sum2 = sum2 + (w_j + w_{n-1-j})$
end for
$extX = min\{\frac{sum1}{sum2} - a_i^{(i-1)}, a_{n-i}^{(i-1)} - a_{n-1-i}^{(i-1)}\}$
for $j = i$ to $\lceil \frac{n}{2} \rceil - 1$ **do**
    Calculate $extXs[j]$ according to (5.24)
end for
for $j = i$ to $\lceil \frac{n}{2} \rceil - 1$ **do**
    $newXs[j] = curr[j] + (extXs[j] - curr[j])MoveRatio$
end for

is the jth point of A after moving the ith support level, w_i is the weight for point a_i, and $extX$, $extXs[]$, $newXs[]$ and $curr[]$ are the extreme moving point, extreme moving points, new points and current points respectively (see chapter 5).

In *CalM* algorithm, the **for** loop in line 2 – 5 costs $O(n)$ computation time and lines 1, 6 and 7 each take $O(1)$. Therefore it costs $O(n)$ in terms of time complexity in total. However, this calculation is based on the assumption that all the weights are non-negative. If, however, that is not the case, the calculation of $extX$ should be modified according to (5.23), resulting in a higher time complexity. Nevertheless, negative weights do not make sense in real world applications.

In the *Move* algorithm, lines 2 – 5 lead to a **for** loop which costs $O(n)$ computation time, and similarly, the part of lines 7 – 9, and that of lines 10 – 12 each take $O(n)$ time whilst line 1 and 6 each cost a unit time of $O(1)$. The whole algorithm thus needs $3O(n)+2O(1) = O(n)$ time complexity.

Now consider the time complexity of the main algorithm – the *Original HS interpolation* which invokes the *CalM* and *Move* algorithms. Lines 1 to 5 each take $O(n)$ computation time as each of them needs linear time with respect to the characteristic point number (n). Line 6 only requires a unit time of $O(1)$. However, lines 7 to 11 form a **for** loop with each step in the **for** loop (line 8, 9 or 10) taking time complexity $O(n)$, thus the whole loop costs $O(n) * O(n) = O(n^2)$ computation time. Based on this estimation, the original HS interpolation method needs more computation time than the KH, modified-KH or general interpolation methods. However, $O(n^2)$ is still acceptable given that n is not significantly large in most cases.

One of the most widely used representative value definitions – the centre of core is implicitly used in the KH, modified-KH and general interpolation methods, although the concept of representative value is not defined explicitly in those methods. In implementing the centre of core RV definition, lines 1 – 5 of the algorithm *CalM* are omitted as the extreme position is fixed to $a_{\lceil \frac{n}{2} \rceil -1}$, which is the top left point's position. The *CalM* algorithm therefore needs $O(1)$ computation time. Due to the same reason lines 1 – 9 of algorithm *Move* only take $O(1)$ time. However, lines 10 – 12 still cost $O(n)$ computation time, resulting in the eventual $O(n^2)$ time complexity for the *Original HS interpolation* algorithm. Nevertheless, the interpolation with the adoption of the core of centre RV definition significantly reduces the computation load.

The enhanced HS interpolative reasoning method is shown in Algorithm 7. Lines 1

Algorithm 7 Enhanced HS interpolation

Input: Polygonal fuzzy set A^* with n characteristic points

Output: Polygonal fuzzy set B^* with n characteristic points

1: Compute intermediate fuzzy sets A' and B'
2: Compute scale rates from A' to A^*
3: Scale A' with scale rates calculated by step 2 to generate set A^s
4: Compute scale rates applied to B'
5: Scale B' with scale rates calculated by step 4 to generate set B^s
6: $A^m = A^s$, $B^m = B^s$
7: Compute move criteria applied to B^m
8: Move B^m with move criteria as calculated in step 7 to generate B^*

to 5, 7 and 8 each take $O(n)$ computation time, while line 6 only costs $O(1)$ time. The whole method thus costs $O(n)$ computation time, which is less than the $O(n^2)$ time required by the original HS method.

The above estimations show that all the interpolation methods except the original HS have the time complexity of $O(n)$, while the original HS requires $O(n^2)$. However, the latter is acceptable given that the number of characteristic points of involved fuzzy sets is normally not significantly large.

6.3 Summary

This chapter has evaluated the original and enhanced HS interpolative reasoning methods as proposed in chapter 5. The comparisons to other existing approaches such as the KH and the general method have been provided with respect to the dependency of the fuzziness of conclusion on observation, the preservation of the piecewise linearity and the computational complexity.

Section 6.2.1 has shown the fuzziness derivation of different interpolative reasoning methods. This evaluation cannot be simply used to justify the performance of an interpolative reasoning method, but it can be used as a guide to choose suitable fuzzy interpolation methods for particular applications. Section 6.2.2 has shown that the original HS method preserves the piecewise linearity property in interpolations involving triangular fuzzy sets, whilst the enhanced HS method preserves this property in interpolations involving arbitrary polygonal fuzzy sets. It is worth noting that the EHS method is the first proposed method having this property. Section 6.2.3 has shown that all the interpolation methods except the original interpolation have computation complexity of $O(n)$, whilst the OHS has $O(n^2)$. However, this is not a problem as the current processors have been fast enough to handle such complexity.

Chapter 7

Transformation Based Interpolation: Realistic Applications

Fuzzy interpolation methods not only help reduce rule bases via removing fuzzy rules which can be approximated by their neighboring rules, but also support reasoning in sparse fuzzy rule bases. This chapter focuses on the original HS fuzzy interpolation method and demonstrates its usages over realistic applications (the usages of the enhanced HS method is omitted as it follows straightforwardly). It first introduces the fuzzy interpolation based inference and then illustrates two realistic applications. In particular, the truck backer-upper problem shows how the proposed OHS interpolation method helps reduce the redundant fuzzy rules, and the computer hardware problem shows how it serves as a fuzzy inference for sparse rule bases. The comparison to the most popularly used inference, Mamdani inference, is presented over these applications.

7.1 Interpolation Based Fuzzy Inferences

Fuzzy inference is used to predict or classify an observation based on a given fuzzy rule base. Traditional fuzzy inferences such as Mamdani [MA75], TSK [TS85, SK88] are designed for reasoning on dense rule bases. That is, at least one fuzzy rule can be chosen to fire for any given observation. If however, this is not the case, traditional fuzzy inferences cannot generate sensible results. In order to resolve this problem, interpolation based fuzzy inferences have been adopted.

Fuzzy interpolation inferences work by using the fuzzy interpolation methods such as the ones proposed in chapter 5. In addition to the capability of handling non-dense (sparse) rule bases, they have a flexibility on choosing different number of fuzzy rules to apply fuzzy interpolations. Fuzzy interpolation based inferences can be used together with traditional fuzzy inferences by setting a *firing threshold*. This threshold decides on which inference scheme will be chosen to use. For instance, a possible implementation as shown in Fig. 7.1 may be: the inference is carried out by the Mamdani method if the maximal firing strength of an observation is greater than the predetermined firing threshold, otherwise, the decision is handed over to a fuzzy interpolation based inference. This is quite flexible as the proportion of unknown data fired by the

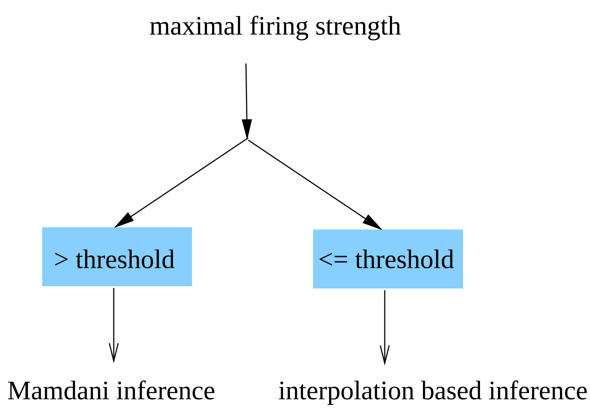

Figure 7.1: An implementation of fuzzy interpolation based inference

interpolation can be decided by the threshold. In the extreme cases, if the firing threshold is set to 0, no firing is made via fuzzy interpolations (if the rule base is dense). On the contrary, if the threshold is set to 1, all data will be fired via fuzzy interpolations.

7.2 Truck Backer-Upper Problem

To demonstrate the usage of interpolation methods, the truck backer-upper problem has been considered in this section. Truck backer-upper problem [NW90, KK92, WM92b, RR01] is considered a well-known benchmark in nonlinear control system and thus raises interest for many researchers. The first attempt is made by using neural network approaches [NW90]. The shortcoming is that neural network needs too much computation load. Then the fuzzy controller has been formulated with the basis of expert knowledge or identified from control data. Although the computation effort is significantly saved, the controller design risks by the curse of dimensionality and suffers the loss of comprehensibility from over-sized rule bases.

The truck backer-upper problem is illustrated in Fig. 7.2. The small cab is the truck which can be determined by three state variables $x \in [0,100]$, $y \in [0,200]$ and $\phi \in [-90,270]$. x and y are the coordinate values for horizontal and vertical axes

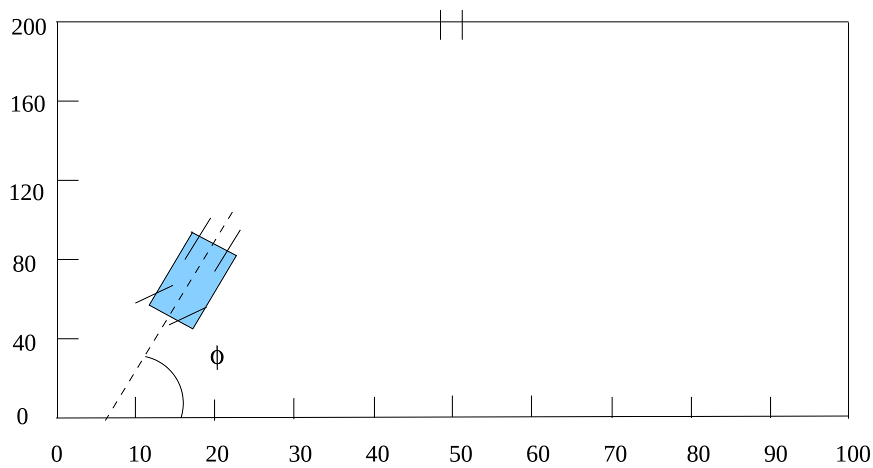

Figure 7.2: Truck backer-upper system

respectively, and ϕ is the azimuth angle between the horizontal axis and the truck's onward direction. The truck begins from certain initial position (x_0, y_0, ϕ_0) and should reverse to the desired end point $(50,200)$ with desired azimuth angle 90. To control the truck, the steering angle $\theta \in [-30,30]$ should be provided after every small move made by the truck. The control problem can thus be formulated as $\theta = f(x,y,\phi)$. Typically, it is assumed that enough clearance between the truck and the loading dock exist so that the truck y-position coordinate y can be ignored, simplifying the controller function to $\theta = f(x,\phi)$.

The demonstration of the interpolation is based on the FISMAT [Lot00] which originally has nine fuzzy rules as shown in Fig. 7.3. Each of the row is interpreted as

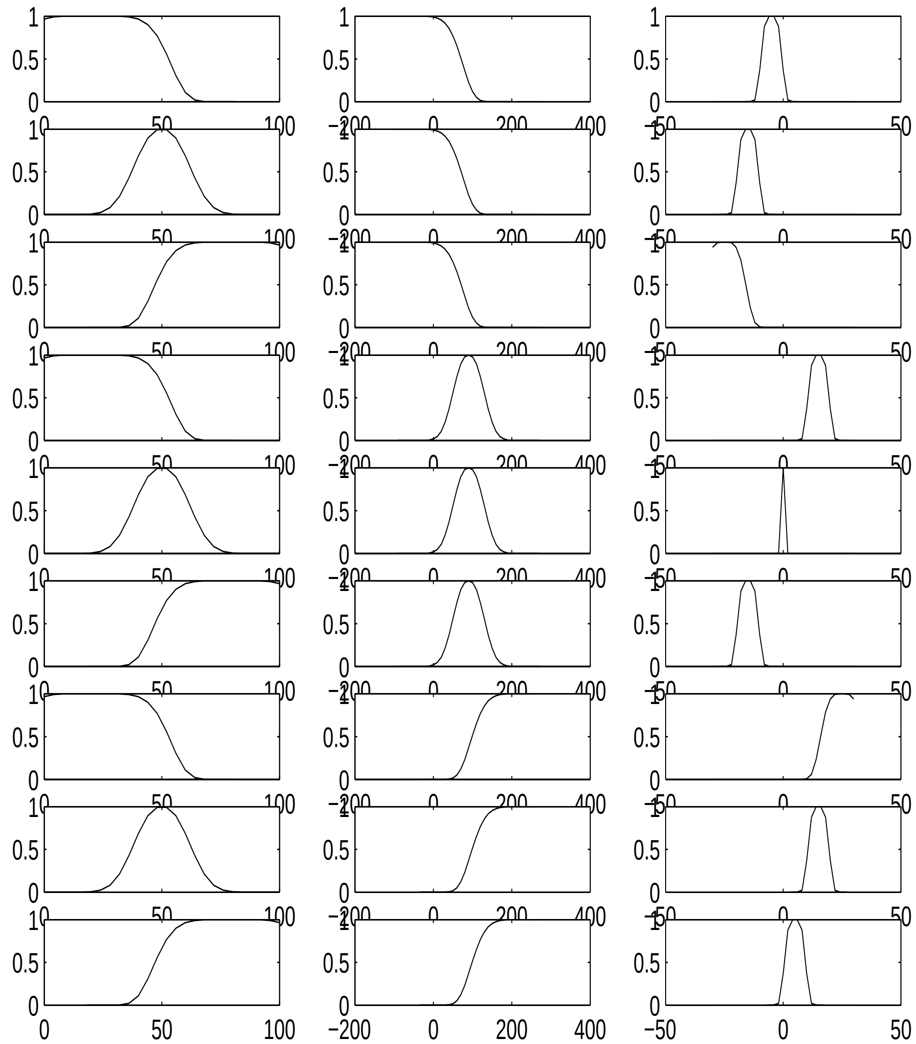

Figure 7.3: Membership functions for 9 rules

a fuzzy rule:

$$\text{IF } x \text{ is } A \text{ AND } \phi \text{ is } B \text{ THEN } \theta \text{ is } C,$$

where A, B and C are the linguistic labels of the system variables. As three linguistic labels are assigned for x and ϕ respectively, it leads to $3 \times 3 = 9$ fuzzy rules in total for this controller. Controlled by these nine fuzzy rules, the truck backing trajectories for

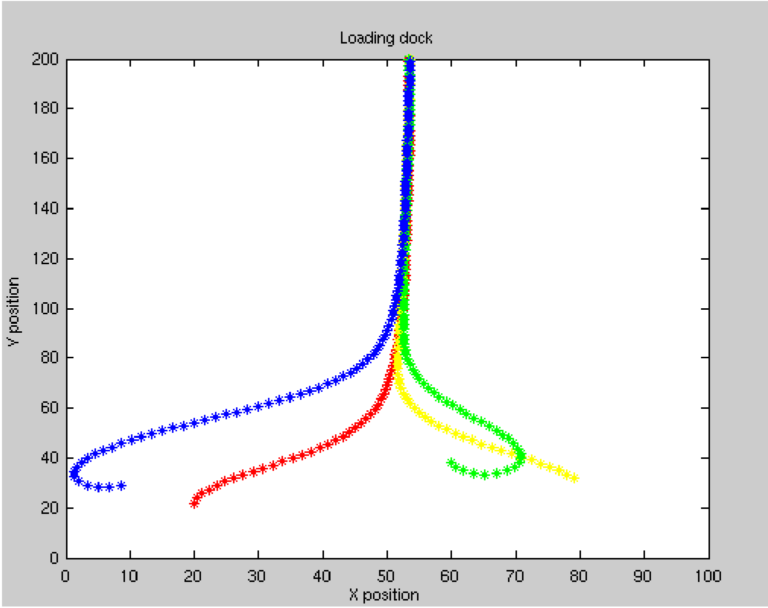

Figure 7.4: Trajectories for 9 fuzzy rules

four initial points are shown in Fig. 7.4. All these four trajectories roughly converge to destination point $(50, 200)$. The reaching position states including x and ϕ for four trajectories are shown in the second row of Table 7.1.

Table 7.1: Reaching positions states

Initial states	(20,20,90)	(80,30,120)	(60,40,-90)	(10,30,220)
9 rules without interpolation	(53.35, 89.69)	(53.45, 90.52)	(53.37, 90.35)	(53.37, 90.58)
6 rules without interpolation	(53.44, 89.51)	(53.40, 90.45)	(53.43, 90.84)	(53.48, 90.84)
6 rules with interpolation	(49.68, 84.65)	(49.49, 84.83)	(49.84, 97.97)	(49.71, 97.98)

Such an expert fuzzy controller may potentially suffers from the curse of dimensionality. That is, as the input variables and the fuzzy linguistic labels associated with each variable increase, the number of rules increases exponentially. This is because the domain partition which is associated with every variable's particular label has to be covered by at least one fuzzy rule, resulting in nine rules in this case. Based on the given nine fuzzy rules (Fig. 7.3), it is intuitive to find that they are symmetrical in some sense. For example, rule 4 and rule 6 are symmetrical if they are mirrored by rule 5: both rules 4 and 6 have the same ϕ, and they are symmetrical for attribute x and θ from rule 5's point of view. This indicates that rule 5 can be interpolated by rule 4 and 6. Thus it may be removed from this fuzzy controller. Similarly, rules 2 and 8 may be removed as they can be interpolated by rules 1 and 3, and rules 7 and 9 respectively. In so doing, a much more compact fuzzy controller which only consists of 6 fuzzy rules is obtained. The trajectories and reaching positions of the truck controlled by the 6 fuzzy rules are shown in Fig. 7.5 and the third row of Table 7.1, which still roughly converge to the destination point.

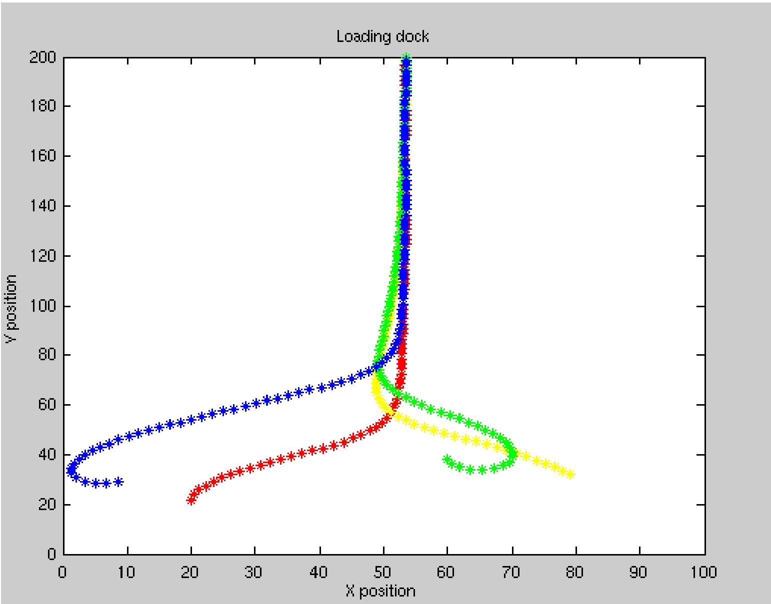

Figure 7.5: Trajectories for 6 fuzzy rules

This simplification potentially brings rule firing problem. As the rule base becomes more and more sparse (due to the removal of rules 2, 5 and 8), it is possible that no fuzzy rules fire for a given observation (truck state here), although this doesn't happen in this experiment. If, however, the firing strength threshold is set to be 0.7 (that is, any rule fires if only the firing strength is greater than 0.7), then no rule fires the observation with x being around 50 and ϕ being around 90. This leads to the sudden breaks of the trajectories as shown in Fig. 7.6.

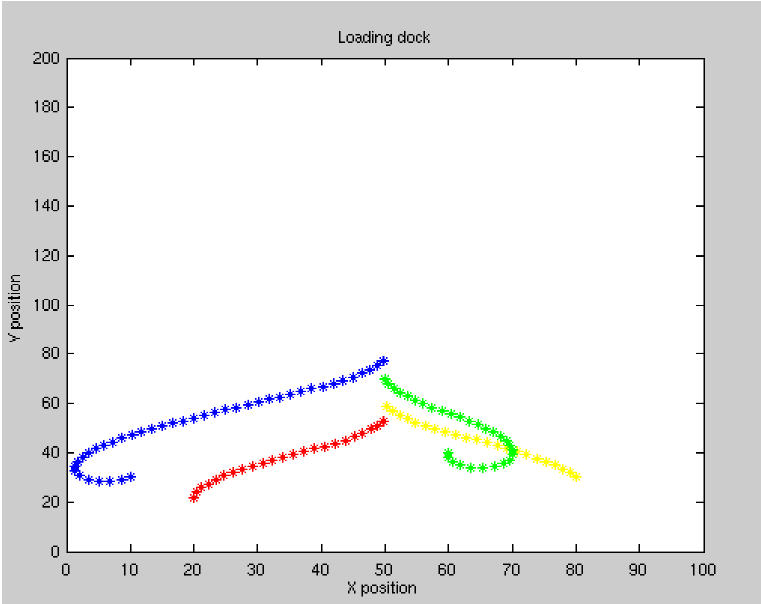

Figure 7.6: Sudden breaks of trajectories for 6 fuzzy rules with firing threshold 0.7

Fuzzy interpolation technique can be deployed to resolve this problem. A possible solution is to pre-determine a threshold to decide which inference (Mamdani or fuzzy interpolation based) should be applied. It indicates that, for a given observation under certain firing strength, the rule base should be treated as sparse. Therefore, the interpolation based inference becomes a natural choice. In this experiment, the threshold is set to be 0.72 after several trials. With the implementation of interpolation using two closest rules, Fig. 7.7 and fourth row of Table 7.1 show that four trajectories better converge to the destination, although with slightly more azimuth error.

This experiment shows that the interpolation method can help simplify a given rule base and support the inferences in a sparse rule base. First, it removes the fuzzy rules which can be approximated (interpolated) by their neighboring rules, resulting in a more compact rule model. This alleviates the curse of dimensionality by keeping important rules only, rather than using all possible rules. Of course, how to decide important fuzzy rules is still an open question, since in scaled-up applications it is not as easy as the selection of key rules in this small application; Second, as an alternative for traditional fuzzy inferences (such as Mamdani and Sugeno), it helps generate the

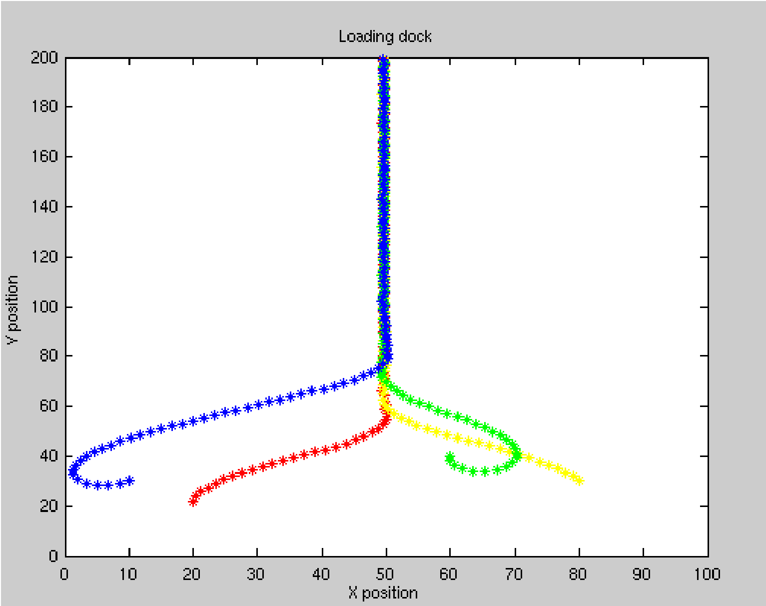

Figure 7.7: Trajectories for 6 fuzzy rules with interpolation

results even no fuzzy rules fire with certain firing strength.

7.3 Computer-Hardware

This section applies the RDFR simplification method (as proposed in chapter 3) and the original HS interpolation method to the computer hardware dataset [HBM98]. This experiment shows that RDFR can result in more compact rule bases and the OHS-based fuzzy inference can outperform Mamdani inference.

Computer hardware dataset concerns with the relative performance of computer processing power on the basis of a number of relevant attributes. This dataset has 209 data, each of which has 7 numerical attributes (including one numerical class). In this experiment, the dataset is divided into a training set and a test set in the following manner. For each data instance, assign a random value $r \in [0.1]$ to it. If $r < 0.5$, then such data instance is put into the training set; otherwise, to the test set. In so doing, 96 data are chosen for training and 113 for test.

For computational simplicity, trapezoidal fuzzy sets are adopted here. An optional factor in this experiment is the way of determining the fuzzy partitions for each numeric attribute. In order to provide an identical platform to compare the performances of using rule bases before and after applying the RDFR-based rule reduction, two methods of fuzzy partitions are used. The first evenly divides the universe of each attribute into partitions with a predetermined number, resulting in a fixed number of evenly distributed fuzzy partitions. The second uses the expectation maximization (EM) algorithm [WF99] to determine the number of clusters for each numeric attribute, and then uses the cluster information to determine the positions of the fuzzy partitions.

7.3.1 Experiment based on evenly divided fuzzy partitions

Due to its popularity in machine learning literature, fuzzy ID3 [Jan98] is applied to the training data of computer hardware dataset to obtain the original fuzzy rule set. As the dataset's output class is numeric, the *relative squared error* is used to evaluate the success of numeric prediction.

Definition 14 *Let the predicted values on the test data be $p_1, p_2, \ldots, p_n$ and the actual values be $a_1, a_2, \ldots, a_n$, the relative squared error is defined as*

$$RSE = \frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{(\overline{a} - a_1)^2 + \ldots + (\overline{a} - a_n)^2}, \tag{7.1}$$

where $\overline{a} = \frac{1}{n}\sum_i a_i$. In fact, relative squared error is made relative to what it would have been if a simple predictor had been used. And the simple predictor in question is just the average of the actual values from the training data. Thus relative squared error takes the total squared error and normalises it by dividing by the total squared error of the average predictor.

Fig. 7.8 shows the relative squared error of the test data with respect to the size of the evenly distributed fuzzy partitions and the number of leaf nodes (used as a criterion to terminate fuzzy ID3 training). As can be seen, the good performance is obtained when the number of fuzzy partitions are in the range of $[4, 11]$.

For comparison between the use of rule bases obtained before and after the RDFR method, a local optimal case (in terms of the relative squared error) with the number of fuzzy partitions being 7 and the number of leaf nodes being 2 is chosen as the base

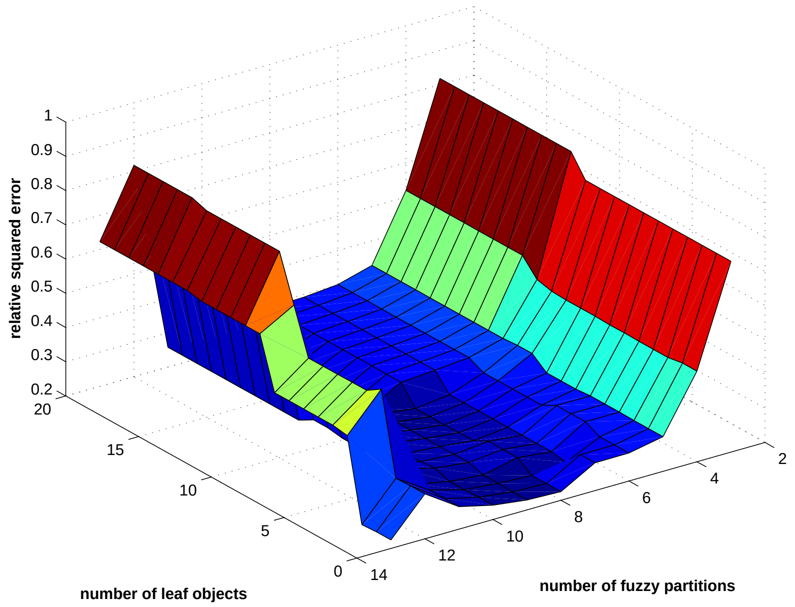

Figure 7.8: Relative squared error of fuzzy ID3 training based on evenly divided fuzzy partitions

comparison point. In this case, 25 fuzzy rules are obtained. Unfortunately, among the whole 113 test data there are 6 data which cannot be fired by any of these 25 rules, resulting in no outputs in these cases. In order to measure the error of the unfired data, each of them is assigned to the average actual output of the whole data (which is 105.62). In so doing, the relative squared error is calculated as 23.53% for this case.

7.3.1.1 OHS interpolation based inference vs. Mamdani

The previous error estimation is based on the use of the Mamdani fuzzy inference, which is not capable of handling the data falling in the gap of the fuzzy rule base (this is why 6 data were not fired). Now the proposed OHS fuzzy interpolation method is applied to the same rule base and test data. Note that during the interpolation, the zoom method is used to construct intermediate rules throughout all the experiments undertaken in this chapter. In contrast to the Mamdani inference, every data is being fired at this time. Fig. 7.9 and Table 7.2 show the relative squared error with respect to the number of interpolated rules and the firing threshold (see section 7.1). The number in brackets is the amount of data which are fired by the interpolation inference (rather than Mamdani inference).

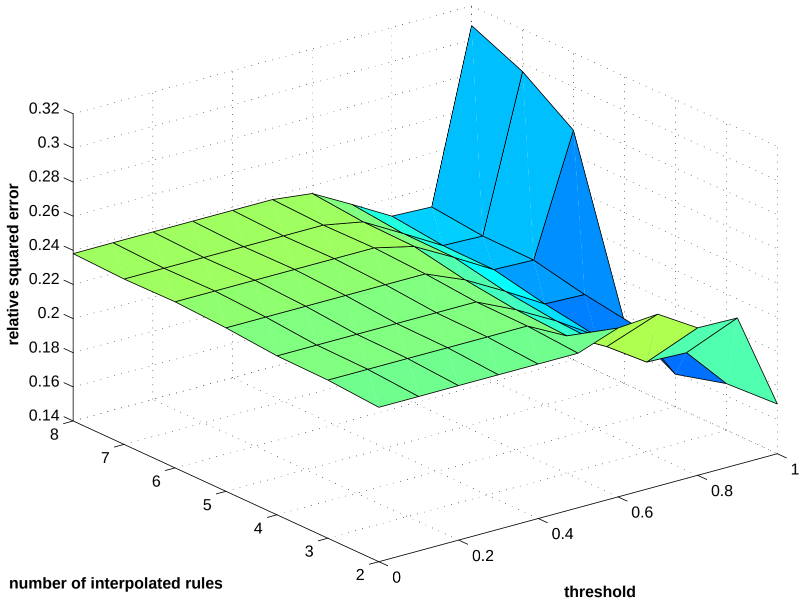

Figure 7.9: Relative squared error of the OHS interpolation inference based on evenly divided fuzzy partitions

Table 7.2: Relative squared error (%) of the OHS interpolation based inference

Threshold	0(6)	0.1(6)	0.2(6)	0.3(6)	0.4(6)	0.5(6)	0.6(14)	0.7(17)	0.8(51)	0.9(53)	1.0(113)
2	23.05	23.05	23.05	23.05	23.05	23.05	23.89	24.08	22.64	22.59	16.91
3	23.27	23.27	23.27	23.27	23.27	23.27	22.06	20.80	19.26	19.18	16.73
4	23.31	23.31	23.31	23.31	23.31	23.31	22.22	20.94	19.45	19.38	15.91
5	23.58	23.58	23.58	23.58	23.58	23.58	22.60	21.30	19.92	19.84	17.60
6	23.73	23.73	23.73	23.73	23.73	23.73	23.18	21.91	20.56	20.48	27.46
7	23.69	23.69	23.69	23.69	23.69	23.69	23.25	21.94	20.60	20.52	29.52
8	23.81	23.81	23.81	23.81	23.81	23.81	23.53	22.26	20.94	20.85	30.85

Generally speaking, the OHS fuzzy interpolation inference produces significantly less relative squared error than Mamdani inference. In particular, if the threshold is 1.0 (that means all 113 test data are fired through the OHS interpolation inference) and the number of participated rules in performing interpolation is in $\{2,3,4\}$ (the normally used cases), the average error is 16.52%, much less than 23.53%. This significant change in performance is due to the fact that the OHS interpolation inference can produce sensible firing results even when the data fall in the gaps which are not covered by the original rule set.

This experiment is based on the assumption that before the test data are fed into the inference mechanisms, they are fuzzified to trapezoidal fuzzy sets by assigning the top support length to be 20 and the bottom support length to be 40 (of course, the centres of the trapezoidals are the same as the original crisp values). The reason for fuzzification is that the test data may not be precise in practice – there are subjective factors such as measurements, readings during the data collections. Fuzzification of the test data may better represent the collected data. Fig. 7.10 shows that different fuzzifications of the test data cause different relative squared errors when the OHS interpolation inference (firing threshold set to 1) is applied. As can be seen, the difference is not significant when the normal size ($[2,5]$) of interpolated rules are used. For simplicity, the following experiments will be undertaken based on the fuzzification of $[20,40]$. Note that fuzzification $[0,0]$ is a specific case in which no fuzzification is required for the input test data. The performance is worse in that case.

7.3.1.2 RDFR rule base simplification

To further reduce the fuzzy rule bases, the RDFR simplification as proposed in chapter 3 is applied in the experiments. As exhaustive RDFR produces too many data (7^6 to be retrieved), random RDFR is used to generate less data, say 200, in this experiment. The PART algorithm [FW98, WF99] is applied to the retrieved data, resulting in 13 ordered rules (i.e. a decision list of 13 fuzzy rules). The performance is estimated through the use of three different fuzzy inference methods, namely, the ordered firing, Mamdani, and OHS interpolation based inference. As the newly generated rule base has a default rule which only consists of a class value and is used to fire a certain test datum if no other rules can fire, its existence may not be suitable for Mamdani and OHS interpolation based inference. To tackle this problem, the default fuzzy rule is simply removed due to 1) the default rule is not as important as other rules in the sense that the default rule usually covers less data than other rules do, and 2) the removal of

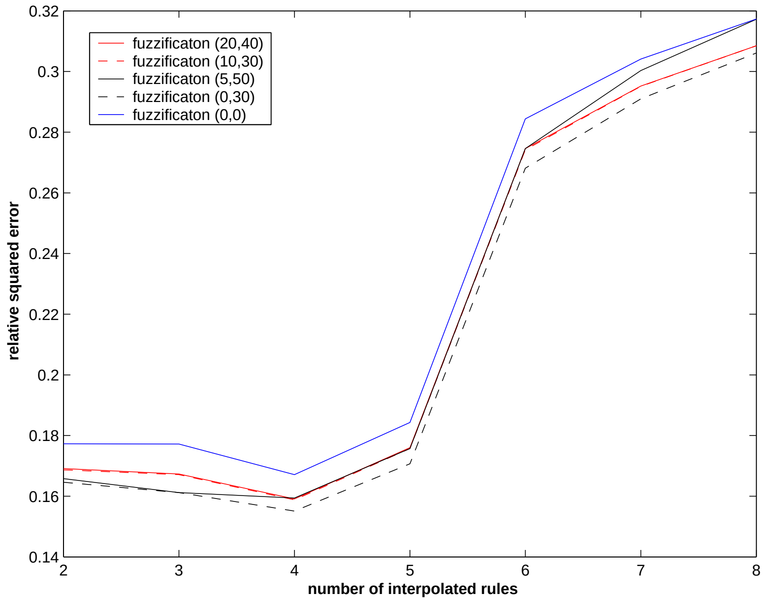

Figure 7.10: Relative squared error of different fuzzifications

this rule will not cause the loss of class entries as the class domain is in fact numerical.

Ordered firing The ordered firing works with a predetermined threshold. In particular, each rule (in the ordered list) attempts to fire the given observation data in turn, it stops when the firing strength of itself is greater than the threshold. The inference result is thus fully decided by this fuzzy rule. Table 7.3 and Fig. 7.11) show that the average of the errors following this approach is 24.63% and the best performance is 23.50% (when threshold = 0.7), which are quite good results in terms of the gain over rule base simplification (from 25 rules down to 13).

Table 7.3: Relative squared error of the ordered firing

Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Error (%)	24.23	24.23	24.23	24.48	24.48	24.48	24.48	23.50	25.93	26.28

Mamdani inference The test of using the 12 rules (after the removal of the default

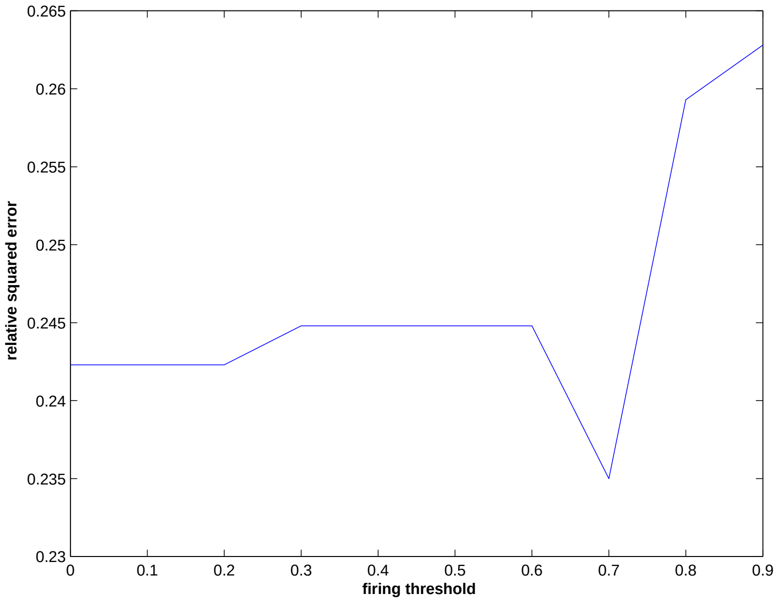

Figure 7.11: Relative squared error of the ordered firing

rule) leads to a relative squared error of 30.32%. Among the whole 113 test data, 6 are not fired. Obviously, this result is not good.

OHS interpolation based inference The errors of the OHS interpolation based inference with respect to the threshold and number of interpolated rules are given in Table 7.4 and Fig. 7.12. The threshold decides what portion of the test data are fired by the OHS interpolation based inference (rather than by Mamdani). The values of 0.7 and 1.0 are tested in this experience, resulting in 8 and 113 data fired by OHS, respectively. Consider the normally used cases (participated interpolated rules in $[3,5]$) and all test data fired via OHS interpolation based inference (threshold = 1.0), the errors are similar to that produced by Mamdani over the original rule base. The best performance here is only 11.84% (with threshold being 1.0 and the number of participated interpolated rules being 10).

Alternatively, the JRip algorithm [WF99] is applied to the retrieved data, resulting in 9 ordered rules. The performance is again examined through the use of three different inference methods.

Table 7.4: Relative squared error (%) of the OHS interpolation based inference

Rule No	2	3	4	5	6	7	8	9	10	11	12
0.7	25.61	25.46	27.31	26.76	26.12	25.86	25.19	25.33	25.51	25.72	25.80
1.0	41.41	23.28	24.65	21.65	16.28	16.75	14.37	11.89	11.84	12.07	26.05

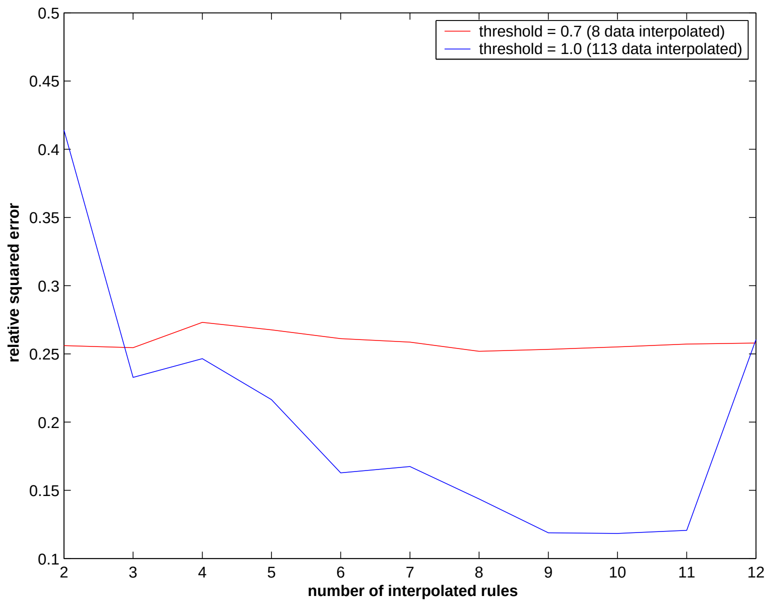

Figure 7.12: Relative squared error of the OHS interpolation based inference

Ordered firing Table 7.5 and Fig. 7.13 show that the ordered firing obtains an average error of 21.51% and the best performance of 20.11% (when threshold = 0.7), which is a much better result compared to the original one (25 rules with error 23.53%).

Table 7.5: Relative squared error of the ordered firing

Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Error(%)	21.57	21.57	21.57	21.57	21.57	21.57	21.57	20.11	21.99	21.99

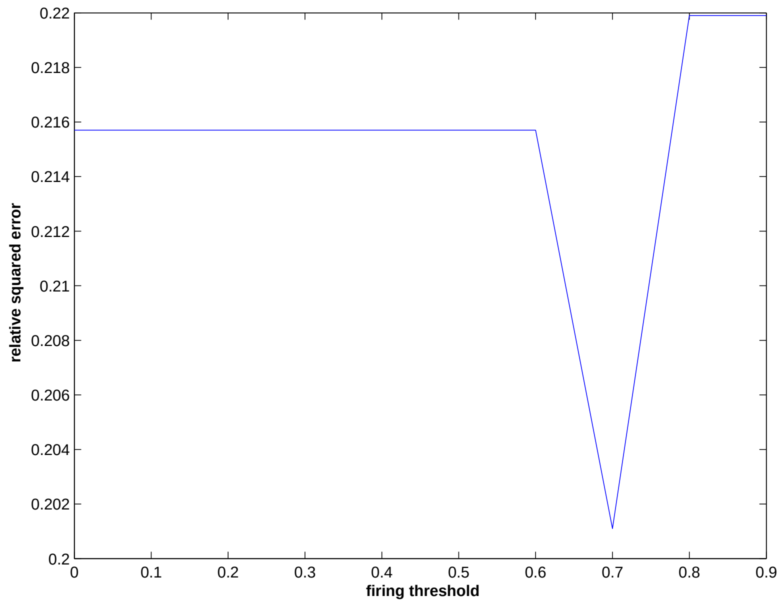

Figure 7.13: Relative squared error of the ordered firing

Mamdani inference The test of the 8 rules (after removal of the default fuzzy rule) leads to a relative squared error of 25.07%. The performance is quite good but it is strange that among the whole 113 test data, 95 are not fired. This is likely due to the reason that most of the unfired data are close to the average result (105.62). They do not contribute much error to the relative squared error of the whole test data.

OHS interpolation based inference The errors of the OHS interpolation based inference with respect to the threshold and number of interpolated rules are given in Table 7.6 and Fig. 7.14. The thresholds of 0.7 and 1.0 are tested in this experience, resulting in 98 and 113 data fired by OHS interpolation respectively. Consider the normally used cases (interpolated rules = [2,5]) and all test data fired via interpolation (threshold = 1.0), the average error is 23.84%, which is not bad compared to the original rule base. The best performance here is 12.89% (threshold = 0.7 and with 6 interpolated rules).

Table 7.6: Relative squared error (%) of the OHS interpolation based inference

Rule No	2	3	4	5	6	7	8
0.7	15.16	14.71	14.35	13.24	12.89	13.07	12.97
1.0	24.79	24.12	23.17	23.26	22.41	21.11	21.12

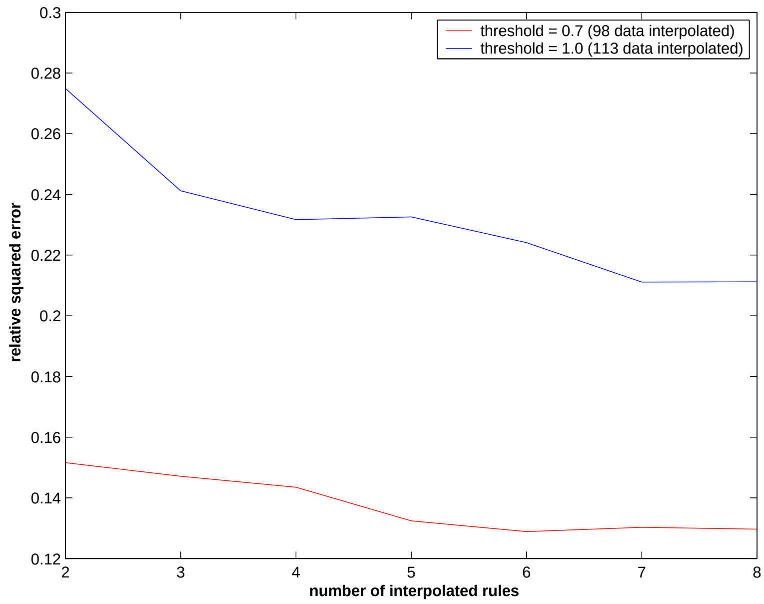

Figure 7.14: Relative squared error of the OHS interpolation based inference

7.3.2 Experiment based on assigned fuzzy partitions

This experiment uses the assigned fuzzy partitions rather than the evenly divided fuzzy partitions. In particular, the Expectation Maximization (EM) algorithm [WF99] is used to determine the number and locations of clusters for each attribute. Using such information, the seven attributes are assigned the fuzzy partitions of 3, 5, 3, 4, 2, 3 and 9 respectively. Note that the partitions of each attribute do not have the same shape, although all of them are trapezoidal fuzzy sets. Again, the fuzzy ID3 algorithm is applied to the training data. The results with respect to the number of leaf nodes are shown in Fig. 7.15. As can be seen, the best performance is obtained when the number of leaf nodes is 15, which is chosen as the base comparison point for future

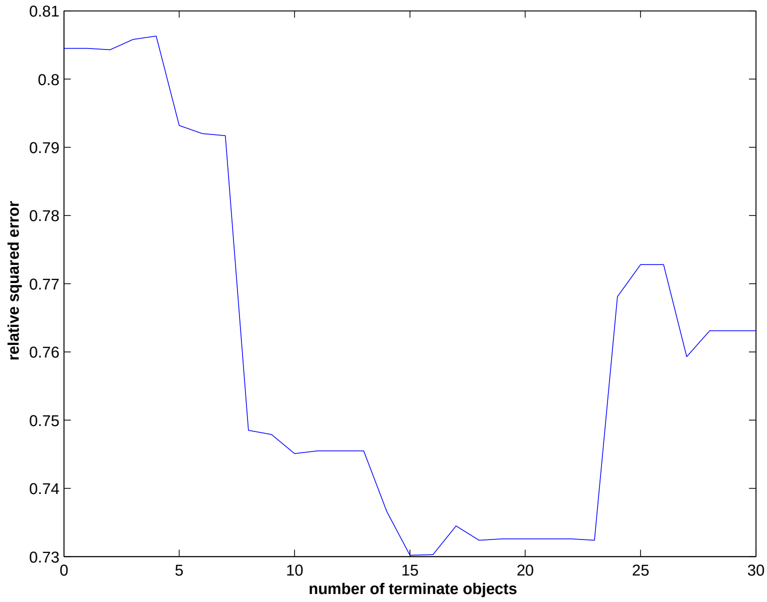

Figure 7.15: Relative squared error of Fuzzy ID3 training based on assigned fuzzy partitions

rule base simplifications. In this case, 33 fuzzy rules with a relative squared error of 73.02% are obtained. Unfortunately, 9 data are not covered by any rules. That is, it is a sparse rule base. This result is much worse than 23.53% which is produced by the experiment based on evenly divided fuzzy partitions. The main reason for this is that the fuzzy partitions for the conditional variables are reduced from 7 to an average of 3.33, which may not be sufficient enough to model the underlying structure. The following experiments show how the combination of RDFR based simplification and the OHS interpolation based inference help produce better results.

7.3.2.1 OHS interpolation based inference vs. Mamdani

With the same rule base and test data, the OHS fuzzy interpolation method outperforms Mamdani inference. Fig. 7.16 and Table 7.7 show the relative squared error of the OHS interpolation based inference with respect to the number of interpolated rules and the firing threshold. Note that the number in the brackets is the amount of data which are fired by the OHS interpolation inference. All the listed relative squared errors are less

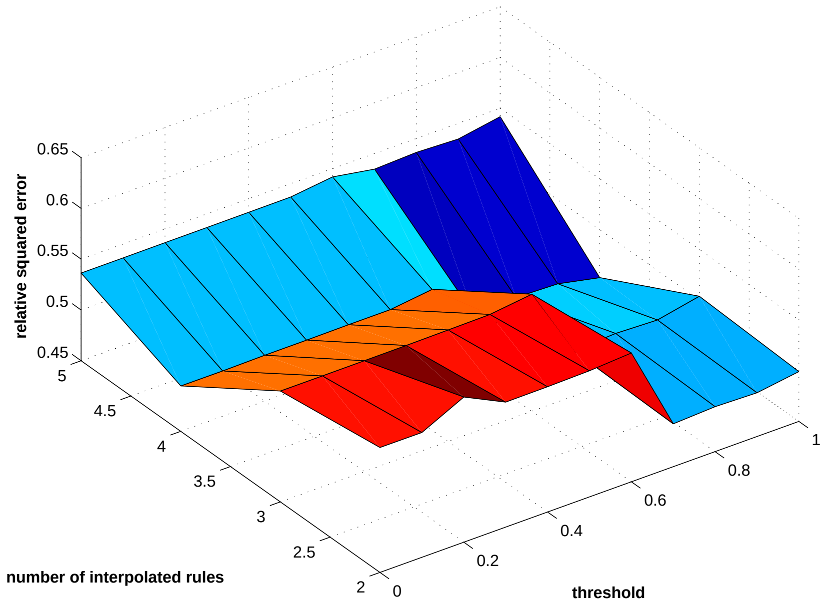

Figure 7.16: Relative squared error of the OHS interpolation inference based on assigned fuzzy partitions

Table 7.7: Relative squared error of the interpolation inference

Threshold	0(6)	0.1(6)	0.2(6)	0.3(6)	0.4(6)	0.5(6)	0.6(14)	0.7(17)	0.8(51)	0.9(53)	1.0(113)
2	57.29	57.29	59.29	57.30	57.34	57.42	57.69	49.22	49.42	49.31	49.90
3	55.91	55.91	55.92	55.94	55.98	56.02	56.53	49.48	49.69	49.53	50.38
4	49.46	49.46	49.51	49.52	49.54	49.55	50.05	46.12	46.29	46.15	45.27
5	53.64	53.64	53.65	53.66	53.67	53.69	54.18	53.46	53.62	53.45	54.17

than the one (73.02%) based on Mamdani inference. In particular, if the threshold = 1.0 (that means all the 113 test data are fired via the OHS interpolation inference) and the number of interpolated rules is in $\{2,3,4\}$ (the normally used cases), the average error is 48.52%, which is much less than 73.02%.

7.3.2.2 RDFR rule base simplification

Similar to the experiment based on the evenly divided fuzzy partitions, random RDFR is used to retrieve 200 data from the original 33 rules. The PART algorithm is applied to the retrieved data, resulting in 19 ordered rules (a decision list of 19 fuzzy rules). The performance is examined as follows.

Ordered firing Table 7.8 and Fig. 7.17 show that the errors of ordered firing are less than the original 73.02% if the firing threshold falls within $[0, 0.5]$. These results are acceptable given that the rule number has been significantly reduced from 33 to 19.

Table 7.8: Relative squared error of the ordered firing

Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Error (%)	72.21	69.97	70.88	70.18	70.53	70.94	74.64	77.80	81.94	93.89

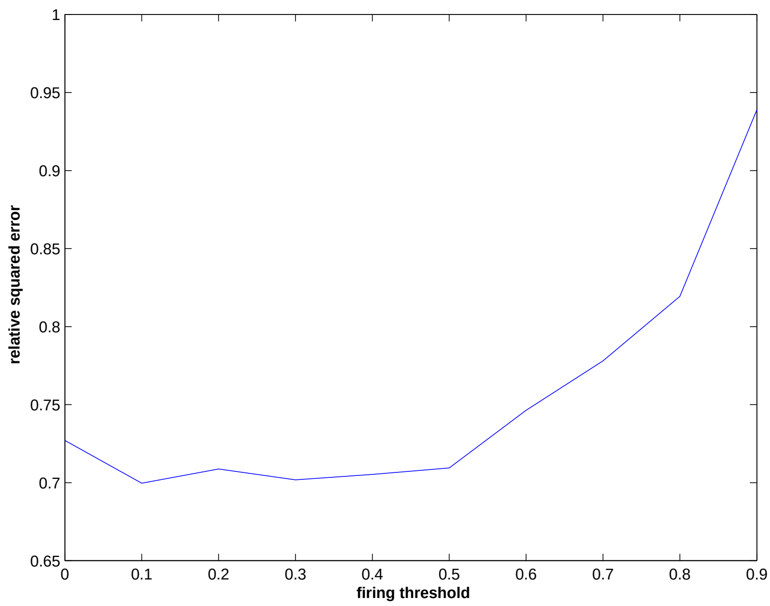

Figure 7.17: Relative squared error of the ordered firing

Mamdani inference The test of the 18 (after the removal of the default one) leads to a relative squared error of 75.32%, leaving 1 datum unfired.

OHS interpolation based inference The errors of the OHS interpolation based inference with respect to the threshold and number of participated rules in interpolation are given in Table 7.9 and Fig. 7.18. Again, the thresholds of 0.7 and 1.0 are tested here, resulting in 8 and 113 data fired by the OHS interpolation based inference respectively. Consider the normally used cases (the number of interpolated rules is in [2,5]) and all test data fired via interpolation (threshold = 1.0), an average error of 48.95% is obtained, which is much better than the error rate produced by the original rule base. The best performance here is 45.64% (when threshold = 1.0 and the number of interpolated rules is 4). Admittedly, such an error rate is itself quite high, but this does not affect the present comparative study as only the relative results are of actual interest.

Table 7.9: Relative squared error (%) of the OHS interpolation based inference

Rule No	2	3	4	5	6	7	8	9	10	11	12
0.7	50.96	48.46	45.50	46.38	54.52	61.98	63.07	61.24	55.15	54.41	57.02
1.0	54.41	49.18	45.64	46.56	58.12	66.04	65.01	61.99	52.26	57.65	56.76

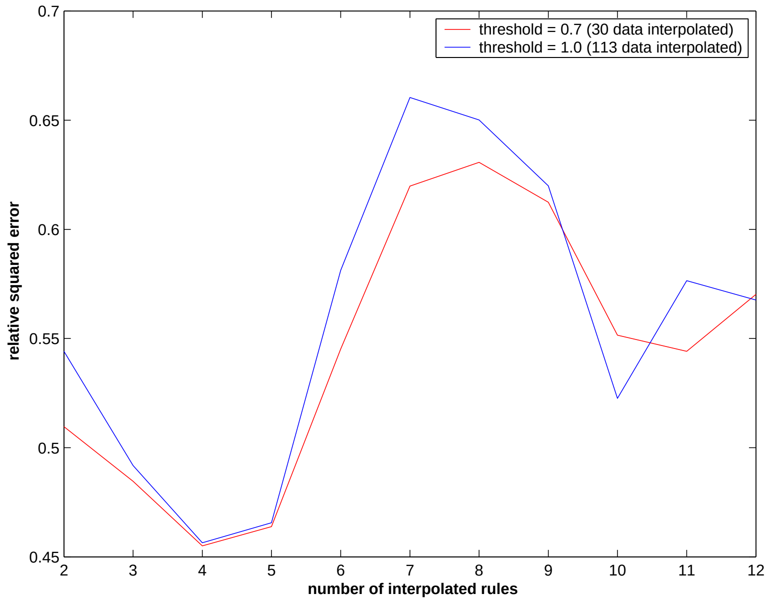

Figure 7.18: Relative squared error of the OHS interpolation based inference

Alternatively, if the JRip algorithm is applied to the retrieved data, 11 ordered rules are obtained. The performance estimations are shown as follows.

Ordered firing The ordered firing results (see Table 7.10 and Fig. 7.19) show that the best performance is 78.06% (when threshold = 0.5), which is worse than the original error rate of 73.02%. This is not considered as a successful simplification.

Table 7.10: Relative squared error of the ordered firing

Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Error(%)	78.48	79.25	79.24	79.10	78.09	78.06	79.95	98.57	98.58	98.56

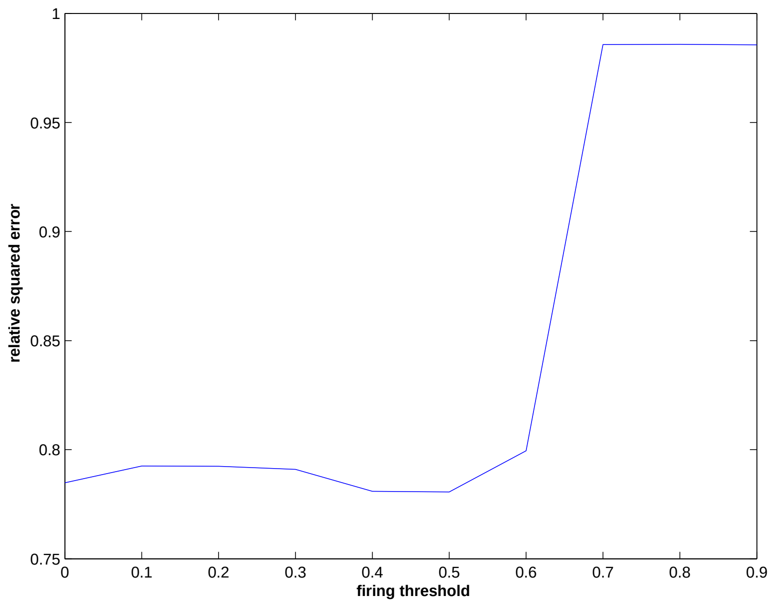

Figure 7.19: Relative squared error of the ordered firing

Mamdani inference The test of the 10 rules (after the removal of the default one) leads to a relative squared error of 74.59%, with 25 data unfired by any rules. It is a good simplification from 33 rules to 10 without significant loss of performance. However, the problem is that some data cannot be handled.

OHS Interpolation based inference The errors of the OHS interpolation based inference with respect to the threshold and number of interpolated rules are given in Table 7.11 and Fig. 7.20. The thresholds of 0.7 and 1.0 are once again tested, resulting in 86 and 113 data fired by the OHS interpolation based inference respectively. Consider the normally used cases (the number of interpolated rules is in $[2,5]$) and all test data fired via interpolation (threshold = 1.0), an average error of 39.16% is obtained, which is much better compared to the error rate produced by original rule base. The best performance here is 36.74% (when threshold = 1.0 and the number of interpolated rules is 9).

Table 7.11: Relative squared error (%) of the OHS interpolation based inference

Rule No	2	3	4	5	6	7	8	9	10
0.7	41.50	39.97	37.31	41.11	44.74	46.51	46.67	52.04	53.55
1.0	42.01	37.71	36.79	40.11	43.25	45.38	45.60	36.74	38.15

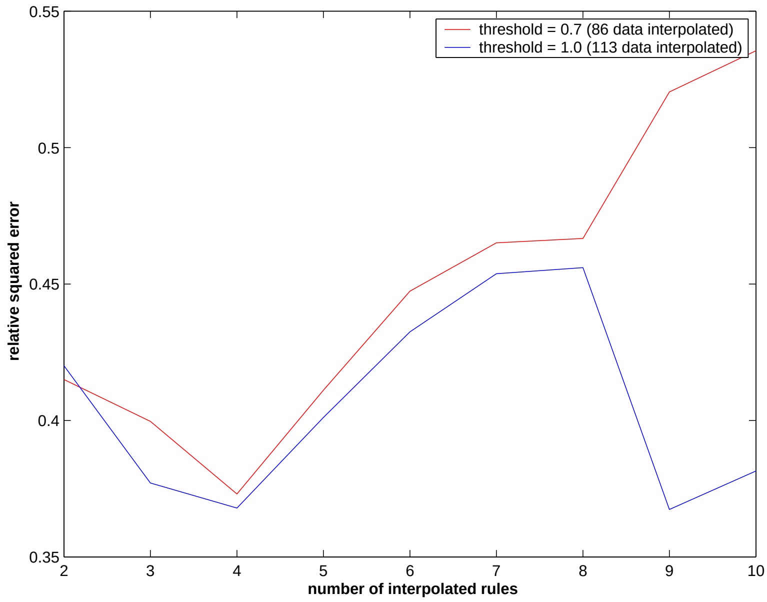

Figure 7.20: Relative squared error of the OHS interpolation based inference

7.4 Summary

This chapter demonstrates the effectiveness of the OHS interpolative reasoning method in two realistic applications. The truck backer-upper problem has shown how the OHS interpolation method help reduce the fuzzy rule bases, and the computer hardware problem has shown how the OHS method serves as an effective fuzzy inference to deal with sparse rule bases. The results in comparison with Mamdani inference have been provided, which highlight the outstanding merit of the present work. In addition, this chapter demonstrates the effectiveness of the RDFR-based rule base simplifications, which produce very good reductions with the use of the OHS method.

Chapter 8

Scaled-up Applications

Unlike some computation based models such as neural networks, rule base models provide a comprehensive and transparent way for system modelling. However, there are no principal routine methods to obtain the optimum fuzzy rule base which is not only compact but also retains high prediction performance. To this end, two major issues need to be addressed. First, the curse of dimensionality [Gui01, KJS02] deteriorates the model if only the structure-complete rules are adopted. Although some research efforts [RZK91, Wan98, ZK04, GP01] have been attempted in designing fuzzy systems with special structures so that the number of rules or parameters employed grows slower than exponentially as the dimension increases, unfortunately, these methods cannot reduce but transfer the complexity. In addition, the relationship between fuzzy rules and the linguistic knowledge in the special structured fuzzy system may no longer be preserved. The scale-up applications described in this chapter avoid this problem by using non structure-complete rules. Attribute selection techniques have also been integrated to simplify the fuzzy rule bases.

The second issue appears following the usage of non structure-complete rules – sparse rule bases (rather than dense ones) may be encountered. The traditional fuzzy inferences such as Mamdani [MA75] cannot handle such sparse rule bases. Interpolation methods have to be used under this circumstance. The comparison between the proposed OHS, EHS interpolation methods and other existing interpolation methods are investigated.

8.1 Task Domain

The computer activity database [RNe96] is a collection of measures over a computer system's activity. The data were collected from a Sun Sparc station 20/712 with 128 Mbytes of memory running in a multi-user university department. Users would typically be doing a large variety of tasks ranging from accessing the internet, editing files or running cpu-bound programs. The data were collected continuously on two separate occasions. On both occasions, system activity was gathered once every 5 seconds. The final dataset is taken from both occasions with equal numbers of observations coming from each collection epoch in random order. This dataset includes 8192 cases, with each involving 22 continuous attributes as shown below. The task is to predict *usr*, portion of time that cpus run in user mode from all attributes 1-21.

1. lread - Reads (transfers per second) between system memory and user memory
2. lwrite - Writes (transfers per second) between system memory and user memory
3. scall - Number of system calls of all types per second
4. sread - Number of system read calls per second
5. swrite - Number of system write calls per second
6. fork - Number of system fork calls per second
7. exec - Number of system exec calls per second
8. rchar - Number of characters transferred per second by system read calls
9. wchar - Number of characters transferred per second by system write calls
10. pgout - Number of page out requests per second
11. ppgout - Number of pages, paged out per second
12. pgfree - Number of pages per second placed on the free list
13. pgscan - Number of pages checked if they can be freed per second

14. atch - Number of page attaches (satisfying a page fault by reclaiming a page in memory) per second

15. pgin - Number of page-in requests per second

16. ppgin - Number of pages paged in per second

17. pflt - Number of page faults caused by protection errors (copy-on-writes)

18. vflt - Number of page faults caused by address translation

19. runqsz - Process run queue size

20. freemem - Number of memory pages available to user processes

21. freeswap - Number of disk blocks available for page swapping

22. usr - Portion of time (%) that cpus run in user mode

8.2 Experimental Results

The data are divided into two folds so that the training data have approximately $2/3$ of the whole data (5462) and test data take the rest (2730). Consider there may exist redundant or less relevant information in the original 22 attributes, a process of attribute selection is carried out to choose the most informative ones. For simplicity, the correlation-based feature subset selection [Hal99, WF99] is used for this, resulting in 11 (read, small, sread, swrite, exec, rchar, pflt, vflt, runqsz, freeswap, and usr) selected attributes.

8.2.1 Initial Fuzzy Rule Base

The well-known fuzzy ID3 training scheme [Jan98] is adopted here to create fuzzy rules. For simplicity, triangular fuzzy sets are used and they are assumed to be evenly distributed over each attribute domain. Fuzzy ID3 with different configurations (in terms of the number of fuzzy sets and the minimal leaf nodes) are carried out and the relative squared errors (relative to the simple average predictor) are shown in Fig. 8.1.

This reveals a trend in the given dataset that the more fuzzy sets used in the training,

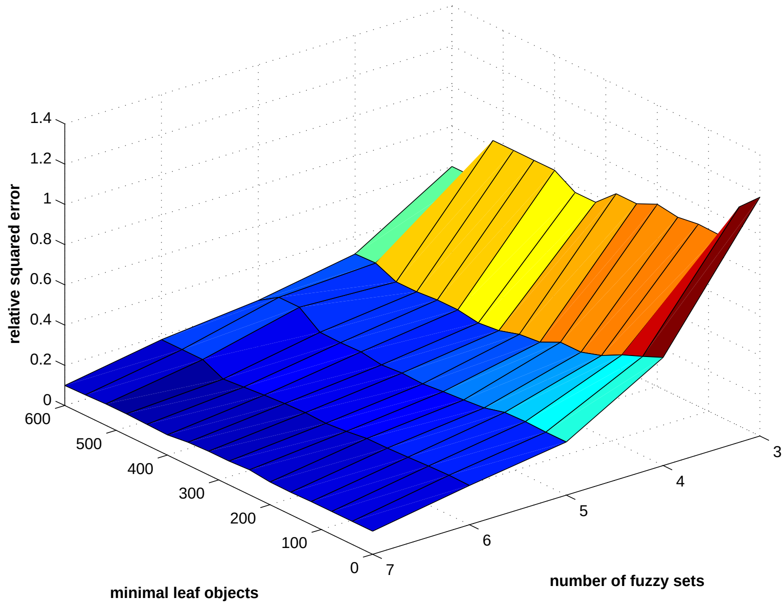

Figure 8.1: Relative squared error

the better performance the resulting rules have. However, the number of rules may become very large at the same time. For instance, with the number of the leaf nodes (as a criterion to terminate the fuzzy ID3 training) being 0, the resulting rule base size increases from 55 to 477 if the number of fuzzy sets increases from 3 to 7. In order to provide a fair platform to compare the interpolation based inference with the well known Mamdani inference [MA75], both the rule base size and the prediction error have to be considered. For this, an optimal resultant rule base, which has 47 rules and an error rate of 13.29%, is chosen (where the number of fuzzy sets is 6 and the number of minimal leaf nodes is 480). Note that in this rule base, 4 among the 2730 test data are not fired by any of the 47 rules using Mamdani inference. That is, the obtained rule base is in fact a sparse rule base.

8.2.2 Interpolation Based Fuzzy Inferences vs. Mamdani

The previous performance evaluation is based on the Mamdani inference. Now the interpolation based inference is tested over this rule base and test data. To provide a fair platform for comparison, all methods used here (the general [BGK96], QMY [QMY96], the proposed OHS and EHS methods) are intermediate rule based fuzzy interpolations (see section 2.4). That is, all of them make use of the intermediate rule in performing interpolations. Other methods such as KH [KH93a, KH93c], modified KH [TB00] are not considered as there is no indication for them to implement the n ($n > 1$) nearest rules interpolations.

It is possible that some attribute values of the intermediate rule exceed the limit of the domain space of that attribute. This is because during the construction of the intermediate rule, extrapolation may be involved and it may lead to the intermediate fuzzy terms becoming out of ranges. It is also possible that the fuzzified data objects exceed the domain space. Therefore, special treatments are desirable for interpolations: For general interpolation, if either the fuzzified data object or the fuzzy term of the intermediate rule exceeds the input space on a particular attribute, such an attribute is ignored in performing the interpolation as this method cannot handle it. Similarly, for QMY and EHS methods, if the intermediate rule has a vertical slope (on either side) for a certain attribute, such an attribute is ignored as these two methods cannot handle this case. However, there is no constraint over the proposed OHS method, thus no attributes would be dropped in performing interpolation with OHS.

The interpolations are based on the assumption that the test data are fuzzified to isosceles triangular fuzzy sets by assigning support lengths with proper portions of the support lengths of the fuzzy terms used in constructing fuzzy rules (of course, the centres of the fuzzified observations are the same as the original crisp values). For example, fuzzification of $(0, 1/8)$ assigns $\frac{1}{8}$ of the support length of the fuzzy terms (used in the rule base) to that of the input data object. The reason of applying fuzzification is that the test data may not be precise in practice due to factors such as measurement and readings errors. Fuzzification of the test data may better represent the collected data. Of course, fuzzification $(0, 0)$ means no fuzzification is performed.

The results of different interpolation methods with respect to various fuzzifications of the test data objects are shown in Table 8.1 and Table 8.2 for shift and zoom intermediate rule constructions (see section **??**) respectively. Note that all errors are calculated as the average of the errors in interpolating two or three nearest rules. The results

Table 8.1: Relative squared error of the interpolation inferences with shift method

Fuzzification	(0, 0)	(0, 1/8)	(0, 1/4)
General	8.45%	60.01%	56.53%
QMY	8.05%	7.62%	7.60%
Original HS (centre of core)	8.05%	7.58%	7.20%
Original HS (average)	6.92%	6.92%	6.92%
Original HS (average weighted)	6.22%	6.25%	6.28%
Enhanced HS (centre of core)	8.05%	7.81%	7.80%
Enhanced HS (average)	6.92%	6.92%	6.92%
Enhanced HS (average weighted)	6.22%	9.53%	18.86%

Table 8.2: Relative squared error of the interpolation inferences with zoom method

Fuzzification	(0, 0)	(0, 1/8)	(0, 1/4)
General	$7.41 \times 10^6\%$	$7.31 \times 10^6\%$	$7.24 \times 10^6\%$
QMY	$7.44 \times 10^6\%$	$7.39 \times 10^6\%$	$7.33 \times 10^6\%$
Original HS (centre of core)	$7.44 \times 10^6\%$	$7.42 \times 10^6\%$	$7.39 \times 10^6\%$
Original HS (average)	242.21%	242.21%	242.21%
Original HS (average weighted)	209.93%	314.00%	608.14%
Enhanced HS (centre of core)	$7.44 \times 10^6\%$	$7.39 \times 10^6\%$	$7.33 \times 10^6\%$
Enhanced HS (average)	325.59%	325.59%	325.59%
Enhanced HS (average weighted)	290.25%	295.17%	303.54%

clearly show that all shift construction based interpolation inferences (except for some cases when using the general interpolation method) outperform Mamdani inference. The reason of the poor performance for the general method is that it drops too many attributes (if either fuzzy terms of the intermediate rules or the fuzzification of input data objects exceed the input domain), resulting in massive information loss. On the contrary, as the original HS method does not need to drop any attributes, it results in very good and stable performance. QMY and the enhanced HS method are between these two and generate better performance than that produced by Mamdani. However, as the strategy of dropping attributes is not part of the general interpolation method, it is assumed so just for the comparison of interpolations involving multiple fuzzy rules. There may exist other possible approaches, in which the general does not necessarily drop attributes, thus hopefully resulting in a better performance.

The best performance is 6.22% where the original (or enhanced) HS interpolation is used and no fuzzification is made for the input data objects. This error is even less than half of the error rate of 13.29% (produced by Mamdani inference). In addition to the high performance, the interpolation methods inferences are capable of firing all data including those were not fired by the Mamdani inference. It is worth noting that the fuzzification of the test data with different support lengths does not significantly affect the prediction error of the original HS method. This ensures the stability of this method. In particular, if the average RV is used, the results are exactly the same across different support lengths. This is because the value of the average RV over a fuzzy set is exactly the same as the fuzzified crisp value created from the defuzzification method used (centre of gravity) over the same fuzzy set.

8.2.3 Shift vs. Zoom

However, the zoom intermediate rule construction method results in poor results. This is because during some rule firing, the γ value may be very small or large, which is far away from the desired stable value (1). This will make the output fuzzy term of the intermediate rule to become very unstable, leading to an enormous error in interpolations. Fortunately, this problem does not occur in the shift constructing method, which makes the shift method a more reliable choice. An example is presented to explain this. Suppose a data object

1.0, 2165, 205, 101, 1.2, 43107, 19.4, 161.8, 3, 1131931, 88

is considered to be fired by the 47 rules in the initial fuzzy rule base, two nearest fuzzy rules r_1 and r_2 are selected and they are listed as follow:

$$r_1 : null, null, null, null, null, null, null, FTerm1, null, FTerm3, FTerm4,$$

$$r_2 : null, FTerm1, null, null, null, FTerm0, null, FTerm1, null, FTerm2, FTerm4$$

where $FTermi$ $(i = \{0, \ldots, 5\})$ is the ith fuzzy term assigned for a particular attribute. Attention is drawn to attribute 6, on which the given data value is 43107, and the fuzzy term of rule r_2 is FTerm0 $(278.0, 278.0, 505552.2)$ (rule r_1 has null value on this attribute). In order to move this fuzzy term to a new position so that it has the RV value 43107, the calculated γ would be 155.06 $>>$ 1 (assume the centre of core RV is used), resulting in the average γ of all input attributes to be 39.37. This further causes the output fuzzy term of the intermediate rule to become $(2338.74, 3118.32, 3897.90)$. This is obviously wrong as the output domain space actually is $[0, 99]$. If, however, the shift is used, the δ calculated on attribute 6 is 0.017, which is close to the stable value 0. It thus leads to a reasonable output term of the intermediate rule $(59.40, 79.20, 99.00)$.

In order to provide a unique platform for the following experiments, several assumptions are made. Firstly, as the general and QMY methods implicitly make use of the centre of core RV value, the original HS and the enhanced HS methods use this RV definition as well. Secondly, the shift method (rather than the zoom one) is chosen in the following experiments due to its stability and effectiveness. Thirdly, the fuzzification of the test data is set to $(0, \frac{1}{4})$. That is, for each attribute, the fuzzification process assigns $\frac{1}{4}$ of the support length of the fuzzy terms (used in the rule base) to that of the corresponding input test data.

Now the RDFR based rule base reduction and interpolation based inferences are applied in the following two subsections, namely the reduction based on 11 attributes and that based on reduced 4 attributes. The difference is that the latter is integrated with the feature selection technique to further reduce the number of attributes from 11 to 4.

8.2.4 RDFR Based Rule Base Reduction over 11 Attributes

Since exhaustive RDFR causes too many data (6^{11} in this case), random RDFR is used. In order to sufficiently represent the model, 2000 data are chosen to be retrieved from RDFR. To better demonstrate the performance of the RDFR rule base reduction and interpolation based fuzzy inference, five experiments based on five different random 2000 data are carried out. In each experiment, PART [FW98, WF99], JRip [WF99] and ID3 [Qui86] are integrated with the RDFR reduction respectively.

8.2.4.1 PART-based RDFR Rule Base Reduction

The PART algorithm is applied to the five sets of 2000 random data. The number of rules generated in the five experiments are shown in Table 8.3. The performance

Table 8.3: The number of rules in PART based RDFR reduction with 11 attributes

#	1	2	3	4	5
Number	31	25	30	28	30

of the new rule bases is examined through three different inference methods, namely, the ordered firing, Mamdani, and the interpolation based inference. As the newly generated rule base has a default rule which only consists of a class value and is used to fire the test data if no other rules can fire, its existence may not be suitable for Mamdani and interpolation based inference. As with before, the default fuzzy rule is simply removed due to the observation that 1) the default rule is not as important as other rules in the sense that it usually covers less data than other rules does, and 2) removal of this rule will not cause the loss of class entries as the class domain is in fact numerical.

Ordered firing The ordered firing works with a predetermined threshold. In particular, each rule (in the ordered list) attempts to fire the given observation data in turn, it stops when the firing strength of itself is greater than the threshold. Fig. 8.2 shows that the errors of the ordered firing (with respect to different thresholds) are not stable, although in a small range of thresholds ($[0.4, 0.5]$) the performance seems good (with the error rates being in the range of $[10.64\%, 12.47\%]$).

Mamdani inference After the removal of the default rule, the relative squared error of the Mamdani inference is shown in Table 8.4. As can be seen, the performance is bad for every experiment.

Interpolation based inferences The average errors of the interpolation based inferences, namely the general, the QMY, the original HS, and the enhanced HS are

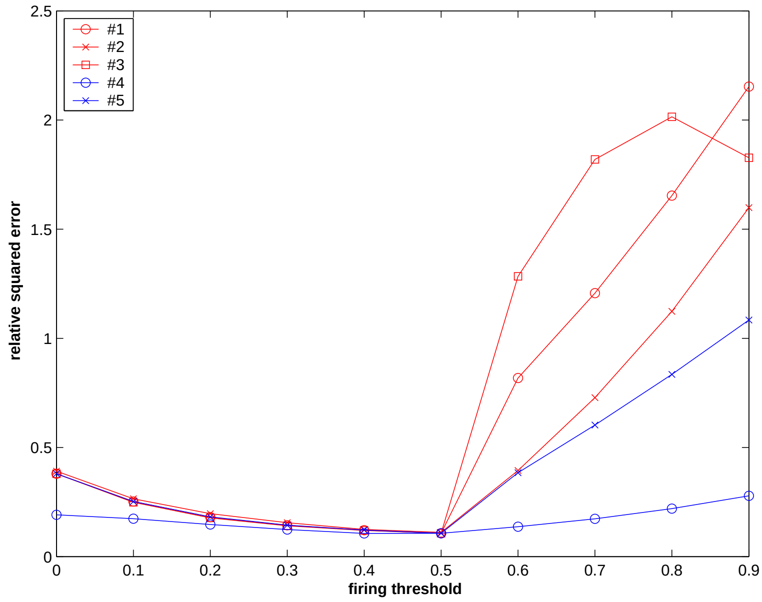

Figure 8.2: Relative squared error of the ordered firing in PART based RDFR reduction with 11 attributes

Table 8.4: Relative squared error of the Mamdani inference in PART based RDFR reduction with 11 attributes

#	1	2	3	4	5
Error	54.37%	60.84%	61.67%	35.98%	56.20%

shown in Table 8.5. As can be seen, all interpolation methods perform differently in the third experiment. This may be because the randomly retrieved dataset in such experiment does not properly represent the underlying data structure, or the PART algorithm cannot learn a proper structure from such a dataset. For all five experiments, although the original HS interpolation based fuzzy inference outperforms or is roughly equal to (only in the third experiment) others, the lowest relative squared error achieved is too high (26.42%, by the original HS in the fourth experiment). It can be concluded that none of them is a successful reduction compared to the original rule base (with an error rate of 13.29%).

Table 8.5: Relative squared error of the interpolation based inferences in PART based RDFR reduction with 11 attributes

#	1	2	3	4	5
General	80.68%	62.60%	115.37%	44.60%	62.37%
QMY	69.50%	55.67%	109.35%	73.43%	34.03%
Original HS	47.61%	34.59%	109.90%	26.42%	29.62%
Linear HS	69.68%	55.91%	109.40%	73.54%	34.10%

It can be summarised that the PART based RDFR rule base reductions do not achieve a satisfiable fuzzy model in this case. In fact, RDFR base reductions provide a *framework* for rule base simplification. The implementation of such framework has many choices. It includes which data retrieving technique is used and how many data are retrieved, which training scheme is chosen to re-train the retrieved dataset, and which fuzzy inference is adopted etc. It is not surprise that the reductions do not always achieve good simplified models.

8.2.4.2 JRip based RDFR Rule Base Reduction

This subsection applies the JRip-based RDFR rule base reduction to five sets of 2000 random data, resulting five new rule bases. The size of such rule bases are shown in Table 8.6. As can be seen, the rule number is significantly simplified from the original 47 to an average of 15. Note that experiments 3 and 4 result in the same number of rules. In fact, these two rule bases are so similar that they lead to nearly identical error rates in the following ordered firing. As with the PART-based RDFR rule base

Table 8.6: The number of rules in JRip based RDFR reduction with 11 attributes

#	1	2	3	4	5
Number	14	16	15	15	14

reduction, the performance is examined through the ordering firing, Mamdani, and interpolation based inferences.

Ordered firing The ordered firing results (see Fig. 8.3) show that all the five experiments except the second produce very consistent and stable error rates. In particular, the fifth experiment produces the best result of 8.87% when the firing threshold is set to 0.4. A maximal error of 13.23% is obtained if the threshold is set in the range $[0, 0.5]$, which can thus be treated as a safe range to test unseen data. In summary, the combination of the JRip-based RDFR reduction and the ordered firing inference offers a very good simplification – it not only simplifies the rule number from 47 to an average of 15, but also increases the prediction accuracy when a proper fire threshold is given.

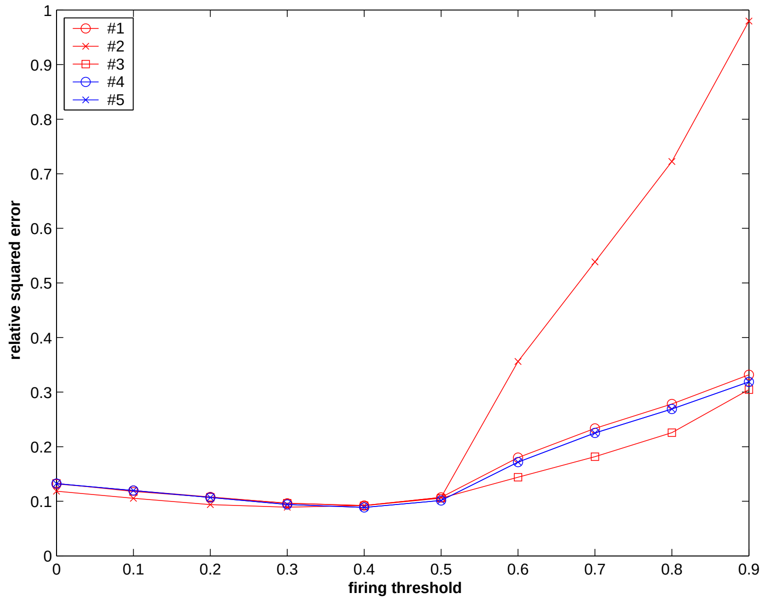

Figure 8.3: Relative squared error of the ordered firing in JRip based RDFR reduction with 11 attributes

Mamdani inference After the removal of the default rule, the relative squared error of the Mamdani inference is shown in Table 8.7. As can be seen, the performance is worse than the original (13.29%) for every experiment.

Interpolation based inferences The average errors of the general, QMY, original HS,

Table 8.7: Relative squared error of the Mamdani inference in JRip based RDFR reduction with 11 attributes

#	1	2	3	4	5
Error	27.78%	24.02%	33.21%	33.83%	33.80%

and enhanced HS methods are shown in Table 8.8. The minimal relative squared error achieved for four interpolation methods is 34.47% (by general interpolation in the first experiment). Although fuzzy inferences based on the general method and the original HS produce less error than the average predictor (by always assigning the average of the output of the training data to be the prediction), neither of them is suitable to perform fuzzy inference. The reason that QMY and the enhanced HS method perform so poorly will be explained in section 8.3.

Table 8.8: Relative squared error of the interpolation based inferences in JRip based RDFR reduction with 11 attributes

#	1	2	3	4	5
General	34.47%	36.89%	45.52%	39.89%	38.50%
QMY	5.56×10^4%	96.16%	4.03×10^4%	4.00×10^4%	4.32×10^4%
Original HS	50.99%	41.33%	55.25%	53.51%	52.95%
Linear HS	5.56×10^4%	97.58%	4.04×10^4%	3.99×10^4%	4.32×10^4%

It can be summarised that the JRip based RDFR rule base reductions do not lead to stable and good results if the Mamdani or interpolation based inference is adopted. However, they do provide promising results when the ordered firing inference is employed.

8.2.4.3 ID3-based RDFR Rule Base Reduction

It is interesting to investigate the results of feeding the retrieved five sets of 2000 data again into the ID3 training scheme. Is the new rule base produced by applying both fuzzy ID3 and crisp ID3 better than the original one (merely produced by fuzzy ID3)? Table 8.9 shows the average number of rules and average error rates of the five experiments with respect to different number of leaf nodes (T_obj, used as a criterion to terminate the training). In fact, the range of $\{50, 100, 150, 200\}$ of T_obj has been fully tested. As the former two settings (50 and 100) may generate more rules than the original of 47 (which is against the purpose of rule base simplification), only the settings of 150 and 200 are used for the results comparison.

Among the five experiments, the best results are achieved by using the original HS based fuzzy inference (as shown in Table 8.10), which has 37 rules with an error rate of 7.25% (T_obj = 150), or has 34 rules with an error rate of 9.69%(T_obj = 200).

Table 8.9: Average results of the ID3 based RDFR reduction with 11 attributes

T_obj	Rule No	Mamdani	general	QMY	original HS	enhanced HS
150	36.8	14.39%	40.53%	3.72×10^3%	10.90%	3.61×10^3%
200	29.2	18.03%	62.45%	7.20×10^5%	17.37%	4.29×10^5%

Table 8.10: Best results of the ID3 based RDFR reduction with 11 attributes

T_obj	Rule No	Mamdani	general	QMY	original HS	enhanced HS
150	37	13.19%	51.82%	8.93%	7.25%	9.14%
200	34	17.67%	53.92%	11.40%	9.69%	11.63%

8.2.5 RDFR Based Rule Base Reduction with 4 Attributes

Feature selection is widely used to filter out the irrelevant or less important attributes. It can thus help achieve more efficient and compact rule models. This subsection illustrates the integration of feature selection into the RDFR based rule base reduction. The basic idea is to apply feature selection techniques to the newly retrieved data, reducing the number of attributes for further re-training. Again, the method of correlation-based feature subset selection [Hal99, WF99] is adopted, leading to only four attributes (scall, vflt, freeswap, usr) remained. The original randomly generated five 2000 data are trimmed so that values of those four attributes are remained for each datum. Once again, five experiments are carried out and the performance is discussed. In each experiment, the PART, JRip and ID3 are integrated into the RDFR rule base simplification.

8.2.5.1 PART-based RDFR Rule Base Reduction

The PART-based RDFR rule base reduction is applied to the five sets of 2000 randomly retrieved data to generate five new rule bases. The rule numbers of such rule bases are shown in Table 8.11. As with before, the performance evaluation is made through the

Table 8.11: The number of rules in PART based RDFR reduction with 4 attributes

#	1	2	3	4	5
Number	24	19	24	25	20

ordered firing, Mamdani, and interpolation based inferences. Again, the default rule is removed when the latter two inference mechanisms are used.

Ordered firing The ordered firing results (see Fig. 8.4) show that the errors are not stable, although in a small range of thresholds ($[0.4, 0.5]$) the performance seems good (the error rates are in the range of $[14.32\%, 16.87\%]$). The best performance is 14.32%, which is achieved by four out the five experiments (with firing threshold set to 0.5).

Mamdani inference After the removal of the default rule, the relative squared error of the Mamdani inference is shown in Table 8.12. As can be seen, the performance is poor in every experiment.

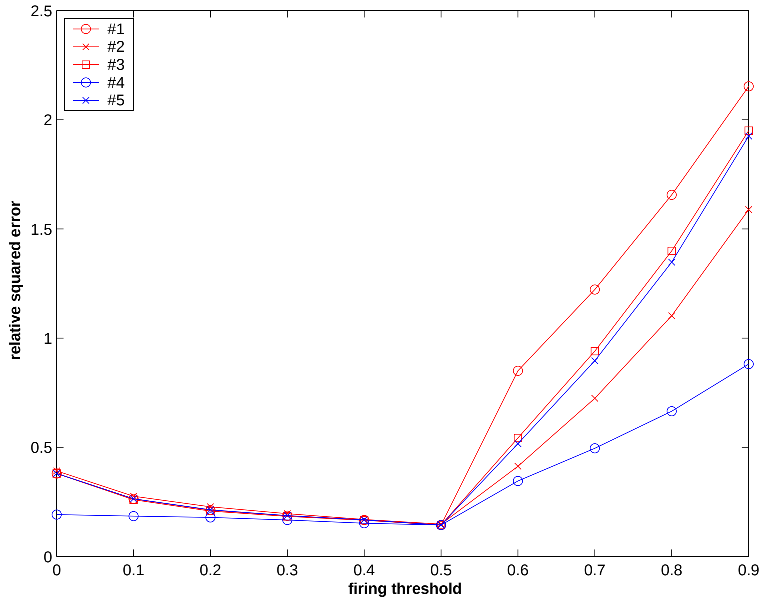

Figure 8.4: Relative squared error of the ordered fire in PART based RDFR reduction with 4 attributes

Table 8.12: Relative squared error of the Mamdani inference in PART based RDFR reduction with 4 attributes

#	1	2	3	4	5
Error	57.26%	56.24%	55.47%	44.81%	50.07%

Interpolation based inferences The average errors of the four interpolation based inferences are shown in Table 8.13. As can be seen, although the original HS interpolation outperforms the others, the minimal relative squared error achieved for four interpolation methods is too large (26.88%, by original HS in the fifth experiment). It can be concluded that none of them is a good reduction compared to the original rule model (with an error rate of 13.29%).

It can be summarised that the use of PART-based RDFR reductions does not lead to a satisfiable fuzzy model.

Table 8.13: Relative squared error of the interpolation based inferences in PART based RDFR reduction with 4 attributes

#	1	2	3	4	5
General	81.93%	55.21%	66.68%	48.83%	57.71%
QMY	69.16%	52.83%	55.51%	49.63%	54.02%
Original HS	47.78%	32.85%	34.26%	28.18%	26.88%
Linear HS	69.30%	53.18%	55.72%	49.87%	54.13%

8.2.5.2 JRip-based RDFR Rule Base Reduction

This subsection applies the JRip-based RDFR rule base reduction to five sets of 2000 randomly retrieved data. Five new rule bases are generated and the sizes of such five rule bases are shown in Table 8.14. As can be seen, the rule number is significantly simplified from the original 47 to an average of 10.4. Note that all experiments except the second result in exactly the same rule base. Once again, the performance of the

Table 8.14: The number of rules in JRip based RDFR reduction with 4 attributes

#	1	2	3	4	5
Number	10	26	10	10	10

simplified rule bases are compared through different fuzzy inferences including the ordered firing, Mamdani, and interpolation based inferences.

Ordered firing The ordered firing results (see Fig. 8.5) show that all the five experiments except the second produce consistent and stable error rates. However, the second experiment produces the best result (9.38%) when the fire threshold is set to 0.2. The maximal error of 14.32% is obtained if the threshold is in the range of $[0, 0.5]$. This gives a very good reduction in terms of rule size (from 47 to an average of 10.2), with little performance compromised.

Mamdani inference After the removal of the default rule, the relative squared error

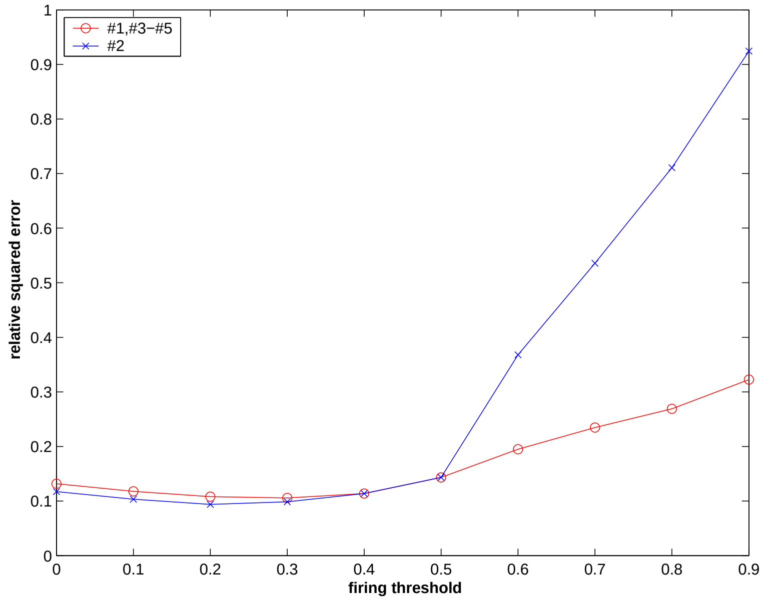

Figure 8.5: Relative squared error of the ordered firing in JRip based RDFR reduction with 4 attributes

of the Mamdani inference is shown in Table 8.15. Although the performance is not so bad, the problem is that this inference cannot handle a large amount of data (330) among the 2730 test data.

Table 8.15: Relative squared error of the Mamdani inference in JRip based RDFR reduction with 4 attributes

#	1, 3, 4 and 5	2
Uncovered data	330	330
Error	20.80%	18.60%

Interpolation based inferences The average errors of the general, QMY, original HS, and enhance HS methods are shown in Table 8.16. The minimal relative squared error achieved for four interpolation methods is 38.38% (by the general interpolation based fuzzy inference). Although the general and original HS based inferences produce less error rate than the inference of average predictor, none of them is satisfiable. The reason that the QMY and enhanced HS methods perform so poorly will be explained in section 8.3.

Table 8.16: Relative squared error of the interpolation based inferences in JRip based RDFR reduction with 4 attributes

#	1,3,4 and 5	2
General	38.38%	40.12%
QMY	3.64×10^6%	49.79%
Original HS	49.36%	39.27%
Linear HS	3.64×10^6%	51.00%

It can be summarised that the JRip-based RDFR rule base reduction does not lead to a practical solution by using Mamdani, nor does it produce stable and good results by using the interpolation based fuzzy inferences. However, it does provide promising results with the usage of the ordered firing inference.

8.2.5.3 ID3-based RDFR Rule Base Reduction

As with before, the range $\{50, 100, 150, 200\}$ of the leaf nodes has been tested. Since only the use of 50 generates more rules than the original number (47), the leaf nodes settings of 100, 150 and 200 are thus employed for the results comparison. Table 8.17 shows the average number of rules and average error rates for the five experiments with respect to the number of leaf nodes (T_obj). From this table, it is clear that the original HS based fuzzy inference outperforms the Mamdani inference, which outperforms the general interpolation based inference. However, the QMY and enhanced HS based inferences again perform very poorly.

Among the five experiments, the best results are achieved with the use of the original HS interpolation based fuzzy inference. Such results (see Table 8.18) include an error rate of 8.85% if T_obj is set 100 or 150 (with 24 rules), or an error rate of 11.25% if T_obj is set 200 (with 21 rules). All these results are more encouraging compared to those produced in section 8.2.4. The main reason is that the experiments in this subsection make use of feature selection techniques.

Table 8.17: Average results of the ID3 based RDFR reduction with 4 attributes

T_obj	Rule No	Mamdani	general	QMY	Original HS	Enhanced HS
100	34.8	16.53%	43.19%	3.72×10^3%	8.70%	3.72×10^3%
150	28.4	17.66%	43.36%	3.72×10^3%	9.32%	3.72×10^3%
200	21.8	21.07%	65.64%	4.35×10^5%	15.48%	4.35×10^5%

Table 8.18: Best results of the ID3 based RDFR reduction with 4 attributes

T_obj	Rule No	Mamdani	general	QMY	Original HS	Enhanced HS
100 or 150	24	16.72%	56.50%	11.03%	8.85%	11.14%
200	21	21.21%	58.61%	13.47%	11.25%	13.58%

8.3 Discussions

Although there are considerable fuzzy interpolation methods existing in the literature, no work has so far been done to apply the fuzzy interpolation methods to real world applications. This chapter has applied fuzzy interpolation based inferences to solve real life problems.

The work carried out in this chapter is based on the shift method to construct the intermediate rules. It has been shown that this method works well with the original HS fuzzy interpolation, but not good with other fuzzy interpolation approaches. There may exist other techniques to create the intermediate rules, through which the performance of other interpolation methods may become better.

As mentioned before, the QMY and enhanced HS interpolation methods cannot handle the certain cases where the intermediate fuzzy terms have vertical slopes. What happens if the intermediate fuzzy terms have slopes which are nearly vertical? In these cases, the rates (for QMY) or the scale criteria (for enhanced HS) become very large, leading to massive error in computing interpolations. An example is given to explain this. Consider the experiments with 4 attributes, if a test datum is given as follows:

$$3783, 717.6, 30, 0,$$

representing the values of attributes scall, vflt, freeswarp and usr (the class attribute) respectively. After fuzzification of the antecedent part of the test datum, the following vector of fuzzy terms is obtained:

$$(3525, 3783, 4041),\ (689.2, 717.6, 746.0),\ (-0.5 \times 10^5, 30, 0.5 \times 10^5),$$

where each element represents a triangular fuzzy set. Given such an observation, the intermediate fuzzy rule constructed by the shift method is

$$null,\ (444.6, 717.6, 990.6),\ (16, 30, 4.5 \times 10^5).$$

Fig. 8.6 shows the data object and the intermediate fuzzy rule. As the intermediate rule has a null value on the first attribute, this attribute is ignored in performing the interpolation.

The third attribute causes trouble for both QMY and enhanced HS interpolations. As can be seen, the intermediate rule has a nearly vertical slope on the third attribute. In this case, the QMY method results in the left ratio being 3337.97, which is far greater than the stable value of 1 (which normal ratios should be close to). The interpolation contribution of this attribute leads to the final result $(-1.90 \times 10^4, 33.51, 35.57)$, which is fuzzified to the crisp output of -6.18×10^3. Obviously, this output is far away from the actual output (0). Similarly, the enhanced HS method results in the left scale criterion being 3337.97 $>>$ 1 (the stable value that normal scale criteria should be close to). It further leads to the output $(-1.86 \times 10^4, 33.51, 35.57)$ in fuzzy form and -6.18×10^3 in crisp form.

For the general interpolation, as the fuzzy term of the observation on this attribute exceeds the range of the domain space ($[2, 2243184]$), this attribute is simply ignored while computing the interpolation, which leads to a crisp output of 33.81. Ignoring

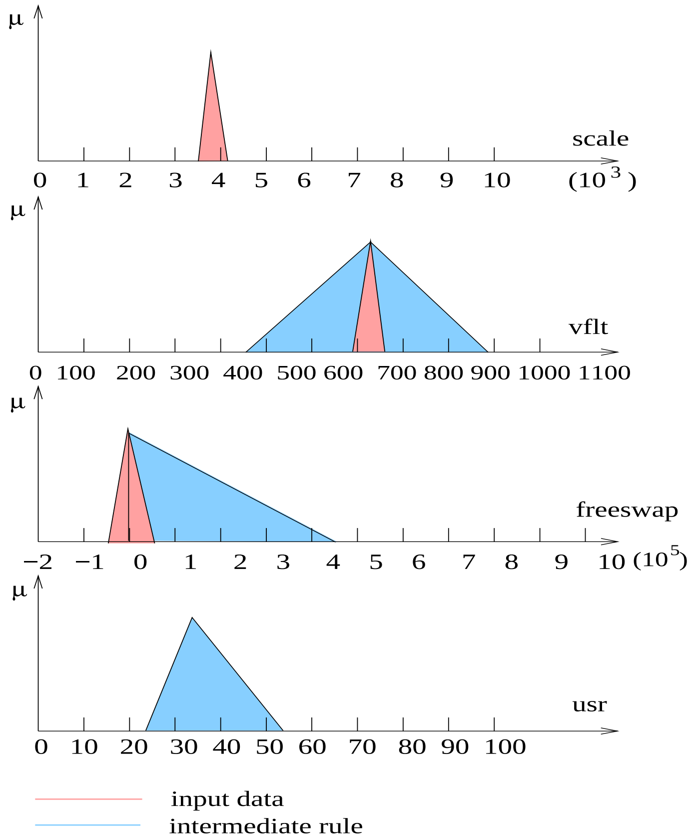

Figure 8.6: An example showing why various interpolation methods perform differently

attribute three makes the interpolation to be still valid (avoiding the difficulty in interpolation using QMY and enhanced HS interpolations). However, it inevitably leads to certain loss of information, resulting in less accurate conclusions. Yet the original HS interpolation handles this case as usual without any loss of information. The result is 33.44, which although is not so close to the actual output (0), is far more accurate than those obtained by using other interpolation methods.

8.4 Summary

As a novel approach, RDFR based rule base simplification provides a flexible and effective framework to simplify rule bases (crisp or fuzzy). Three training schemes including PART, JRip and ID3 have been integrated into this framework to solve real world problems.

This chapter has shown not only the success of the RDFR rule base reduction, but also the potential of interpolation based fuzzy inferences. Their major advantage is that they are capable of handling sparse rule bases. The comparison between different interpolation based fuzzy inferences have shown that the original HS interpolation outperforms the others (although the enhanced HS has the advantage of preserving piecewise linearity, unfortunately, it cannot obtain as good performance as the original one). The main reason is that the original HS method is robust enough to handle the vertical slope cases as described in section 8.3.

In all experimental studies there has been no attempt to optimise fuzzification. It can be expected that the results obtained with optimization would be even better than those already observed. In addressing real world applications, this optimization should be done via domain heuristics or by exploring fuzzy clustering algorithms in order to further improve the performance of the systems.

Chapter 9

Conclusion

This chapter concludes the thesis. Firstly, a summary of the research presented in this thesis is given. Secondly, possible future work is outlined, including several further developments for the RDFR rule base simplification method as well as those for the family of HS interpolative reasoning methods.

9.1 Thesis Summary

This section summarises the main work which includes a novel rule base simplification method and a family of fuzzy interpolation methods. The combination of these two approaches results in very good reductions of fuzzy rule bases as described in chapter 8.

9.1.1 RDFR Rule Base Simplification Method

Rule model simplification techniques are desired to alleviate the *curse of dimensionality* and to maintain models' effectiveness and transparency. This thesis has proposed a novel simplification method by means of a procedure called *retrieving data from rules (RDFR)*. It first retrieves a set of new data from an original rule base. Then it retrains the new data using certain rule induction schemes to build a more compact rule model, while maintaining a satisfactory performance. This proposed method has four advantages. 1) It can reduce rule bases without using the original training data, and it is capable of handing the case in which both a rule base and some training data are given. 2) It builds a flexible framework in which any rule induction or reduction methods can be integrated. 3) It implements the approaches of similarity merging [CCT96, KB95, SBKL98] and inconsistency removal [XL02]. 4) It makes use of rule weights (if applicable). Illustrative examples and realistic applications have been provided to demonstrate the success of this work.

9.1.2 HS Fuzzy Interpolations

This thesis has proposed a generalised, scale and move transformation-based, interpolative reasoning method (original HS method) which can handle interpolation of complex polygonal, Gaussian and other bell-shaped fuzzy membership functions. The method works by first constructing a new intermediate rule via manipulating two adjacent rules (and the given observations of course), and then converting the intermediate inference result into the final derived conclusion, using the scale and move transformations. This has been further developed into the enhanced HS method. It can preserve the piecewise linearity property for any polygonal fuzzy membership functions. The extension to interpolation (and extrapolation) involving multiple variables and multiple rules is accommodated in detail.

The original HS method not only inherits the common advantages of fuzzy interpolative reasoning – allowing inferences to be performed with simple and sparse rule bases, but also has another two advantages: 1) It provides a degree of freedom to choose various RV definitions for different application requirements. 2) It can handle the interpolation of multiple rules, with each rule having multiple antecedent variables associated with arbitrary polygonal fuzzy membership functions. In addition to all the advantages the OHS has, the enhanced HS method has less computation cost than OHS (see chapter 6), and preserves the piecewise linearity property for any polygonal fuzzy functions (see chapter 6). It is worth stressing that the piecewise linearity property is essential to ignore artificial characteristic points in performing the interpolations. Unfortunately, the enhanced HS method does not perform as well as the original HS one in practice as it cannot properly handle the vertical slope cases (see section 8.3). This is set as the main future work to be resolved.

The original and enhanced HS methods lead to a big family of interpolative reasoning methods. This is because of 1) the flexibility in choosing different RVs in implementation, 2) the order swap of scale and move transformations, and 3) the alternative choices for the order of computing the scale rates (or move rates). For example, in the original HS method, the scale rates are calculated from the bottom to the top of the fuzzy sets, the alternative solution may calculate in the reverse way: from the top to the bottom.

9.1.3 Complex Model Simplification

As a novel approach, RDFR rule base simplification provides a framework to effectively and efficiently simplify the rule bases (either crisp or fuzzy). This method has been applied to the computer activity dataset [RNe96] which includes 8192 cases, with each having 22 continuous attributes. The scaled-up application (chapter 8) has shown not only the success of the RDFR based rule reduction, but also the potential of the interpolation based fuzzy inferences. The major advantage of the interpolation based inferences is that they are capable of handling sparse rule bases. The comparison has shown that the original HS interpolation outperforms Mamdani, general, QMY and the enhanced HS interpolation approaches.

In all experimental studies there has been no attempt to optimise the fuzzifications employed. It can be expected that the results obtained with optimization would be even better than those already observed. In solving a real world problem, this optimization should be done via heuristics or exploring clustering algorithms in order to further improve the performance of the systems.

9.2 Future Work

This section presents important further work to improve the RDFR rule base simplification method and the family of the HS fuzzy interpolation mechanisms.

9.2.1 RDFR Rule Base Simplification Method

Different retrieving methods are needed to carefully investigate with respect to different weighted rules. The number of retrieved data should reflect the importance of the given rule in terms of its weight. Thus the principle should be: the greater weight a fuzzy rule has, the more data are retrieved from this rule.

Also, this method only applies to *non-structure-complete* rules. The retrieving techniques to coping with *structure-complete* rules require further research. That is, numeric data rather than the fuzzy linguistic terms based data will be retrieved. New fuzzification partitions should be employed. This may risk destroying the semantic meaning of the original predefined fuzzy sets, but it may open a new door to form more reasonable fuzzy partitions, thereby leading to a more efficient way of modelling the given problems.

9.2.2 HS Fuzzy Interpolations

Although the family of the HS interpolation methods have been significantly developed, there is still room to improve the present work. In particular, the piecewise linearity is worth further analyzing from the mathematical perspective. Since fuzzy sets can be represented as points in high dimensional Cartesian spaces [YK00], a fuzzy interpolation can be represented as the mapping from one point in a high dimensional space to one point in another (with the two spaces having the same dimensionality which is equal to the number of the characteristic points of the considered fuzzy sets). Due to the preservation of the piecewise linearity, the enhanced HS method may be used in the mathematics literature to solve high dimension spaces interpolation (or mapping) problems.

In addition, more development is desirable for the enhanced HS method. Although it perfectly preserves the piecewise linearity property, it cannot produce as good performance as that given by the original HS method in the scaled-up applications. This is due to its less robustness in handling the interpolation cases which involve vertical slopes in considered fuzzy sets. Further effort to improve its robustness seems necessary.

Bibliography

[AD91] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *Proc. on Artificial Intelligence*, pages 547 – 552, 1991.

[BB92] D. G. Burkhardt and P. P. Bonissone. Automated fuzzy knowledge base generation and tuning. In *Proc. IEEE Int. Conf. on Fuzzy Systems*, pages 179 – 188, 1992.

[BG96] P. Baranyi and T. D. Gedeon. Rule interpolation by spatial geometric representation. In *Proc. Inform. Processing Management Uncertainty Knowledge-Based Syst.*, pages 483–488, 1996.

[BGK95] P. Baranyi, T. D. Gedeon, and L. T. Kóczy. A general method for fuzzy rule interpolation: specialized for crisp, triangular, and trapezoidal rules. In *Pro. Eur. Conf. Fuzzy and Intelligent Techniques*, pages 99 – 103, 1995.

[BGK96] P. Baranyi, T. D. Gedeon, and L. T. Kóczy. A general interpolation technique in fuzzy rule bases with arbitrary membership functions. In *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, pages 510–515, 1996.

[BK96a] P. Baranyi and L. T. Kóczy. A general and specialized solid cutting method for fuzzy rule interpolation. *Journal BUSEFAL, Automne, URA-CNRS*, 1996.

[BK96b] P. Baranyi and L. T. Kóczy. Multi-dimensional fuzzy rule inter- and extra-polation based on geometric solution. In *Proc. Int. Power Electronic and Motion Control Conf.*, volume 3, pages 443 – 447, 1996.

[BKG04] P. Baranyi, L. T. Kóczy, and T. D. Gedeon. A generalized concept for fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 12(6):820 – 837, 2004.

[BL97] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1), 1997.

[BM00] B. Bouchon-Meunier. Analogy and fuzzy interpolation in fuzzy setting. In *Proc. Int. Conf. Fuzzy Sets Theory and Its Applications*, page 7, 2000.

[BMDM+99] B. Bouchon-Meunier, J. Delechamp, C. Marsala, N. Mellouli, M. Rifqi, and L. Zerrouki. Analogy and fuzzy interpolation in the case of sparse rules. In *Proc. Joint Conf. Eurofuse-Soft and Intelligent Computing*, pages 132 – 136, 1999.

[BMDM+01] B. Bouchon-Meunier, D. Dubois, C. Marsala, H. Prade, and L. Ughetto. A comparative view of interpolation methods between sparse fuzzy rules. In *Proc. International Fuzzy Systems Association Congr.*, pages 2499–2504, 2001.

[BMK+98] P. Baranyi, S. Mizik, L. T. Kóczy, T. D. Gedeon, and I. Nagy. Fuzzy rule base interpolation based on semantic revision. In *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, volume 2, pages 1306 – 1311, 1998.

[BMMR00] B. Bouchon-Meunier, C. Marsala, and M. Rifqi. Interpolative reasoning based on graduality. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 483–487, 2000.

[BRV98] R. Babuska, J. A. Roubos, and H. B. Verbruggen. Identification of mimo systems by input-outputs its fuzzy models. In *Proc. IEEE International Conference on Fuzzy Systems*, volume 1, pages 657 – 662, 1998.

[BTYK99a] P. Baranyi, D. Tikk, Y. Yam, and L. T. Kóczy. Investigation of a new α-cut based fuzzy interpolation method. Technical Report CUHK-MAE-9906, 1999.

[BTYK99b] P. Baranyi, D. Tikk, Y. Yam, and L. T. Kóczy. A new method for avoiding abnormal conclusion for α-cut based rule interpolation. In *Proc. IEEE Int. Conf. on Fuzzy Systems*, pages 383–388, 1999.

[Buc93] J. J. Buckley. Sugeno type controllers are universal controllers. *Fuzzy Sets and Systems*, 53(3):299–304, 1993.

[BV95] R. Babuska and H. B. Verbruggen. A new identification method for linguistic fuzzy models. In *Proc. IEEE International Conference on Fuzzy Systems*, volume 2, pages 905 – 912, 1995.

[Cas95] J. L. Castro. Fuzzy logic controllers are universal approximators. *IEEE Transactions on Systems, Man and Cybernetics*, 25(4):629–635, 1995.

[CCG91] S. Chen, C. F. N. Cowan, and P. M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2):302–309, 1991.

[CCT96] C. T. Chao, Y. J. Chen, and C. C. Teng. Simplification of fuzzy neural systems using similarity analysis. *IEEE Transactions on Systems, Man and Cybernetics Part B*, 26(2):344–354, 1996.

[CLL01] S. M. Chen, S. H. Lee, and C. H. Lee. A new method for generating fuzzy rules from numerical data for handling classification problems. *Applied Artificial Intelligence*, 15(6):645–664, 2001.

[CLRS90] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 1990.

[CZ97] J. L. Castro and J. M. Zurita. An inductive learning algorithm in fuzzy systems. *Fuzzy Sets and Systems*, 89(2):193–203, 1997.

[DHR93] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Berlin, Germany: Springer, 1993.

[DL97] M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1:131–156, 1997.

[DP92] D. Dubois and H. Prade. Gradual inference rules in approximate reasoning. *Information Science*, 61:103 – 122, 1992.

[DP99] D. Dubois and H. Prade. On fuzzy interpolative reasoning. *International Journal of Systems*, 28:103 – 114, 1999.

[DPG95] D. Dubois, H. Prade, and M. Grabisch. *Gradual rules and the approximation of control laws*. New York: Wiley, 1995.

[DSM89] L. Ding, Z. Shen, and M. Mukaidono. A new method for approximate reasoning. In *Proc. Nineteenth International Symposium on Multiple-Valued Logic*, pages 179 – 185, 1989.

[DSM92] L. Ding, Z. Shen, and M. Mukaidono. Principle for approximate reasoning – based on revision principle. In *2nd International Conference Fuzzy Logic and Neural networks*, pages 305 – 308, 1992.

[FW98] E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. In *Proc. of the Fifteenth International Conference on Machine Learning*, pages 144–151, 1998.

[GK96] T. D. Gedeon and L. T. Kóczy. Conservation of fuzzyness in rule interpolation. In *Proc. Symp. New Trends in the Control of Large Scale Systems*, volume 1, pages 13–19, 1996.

[GL83] G. H. Golub and C. G. Van Loan. *Matrix computations*. Johns Hopkins Univ. Press, 1983.

[GP01] M. K. Güven and K. M. Passino. Avoiding exponential parameter growth in fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 9(1):194–199, 2001.

[Gui01] S. Guillaume. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, 9(3):426–443, 2001.

[Hal99] M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, Department of Computer Science, University of Waikato, 1999.

[HB99] O. Huwendiek and W. Brockmann. Function approximation with decomposed fuzzy systems. *Fuzzy Sets and Systems*, 101(2):273 – 286, 1999.

[HBM98] S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998. Available at `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[HCL98] W. H. Hsiao, S. M. Chen, and C. H. Lee. A new interpolative reasoning method in sparse rule-based systems. *Fuzzy Set and Systems*, 93(1):17–22, 1998.

[HM95] J. Hohensohn and J. M. Mendel. Two-pass orthogonal least-squares algorithm to train and reduce fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 1, 1995.

[HS03] Z. H. Huang and Q. Shen. A new fuzzy interpolative reasoning method based on center of gravity. In *Proc. of the International Conference on Fuzzy Systems*, volume 1, pages 25–30, 2003.

[HS04a] Z. H. Huang and Q. Shen. Fuzzy interpolation with generalized representative values. In *Proc. of the 2004 UK Workshop on Computational Intelligence*, pages 161–171, 2004.

[HS04b] Z. H. Huang and Q. Shen. Scale and move transformation-based fuzzy interpolative reasoning: a revisit. In *Proc. of the International Conference on Fuzzy Systems*, volume 2, pages 623–628, 2004.

[HS05a] Z. H. Huang and Q. Shen. Fuzzy model simplification and inference via fuzzy interpolative and extrapolative reasoning. preprint, 2005.

[HS05b] Z. H. Huang and Q. Shen. Model simplification via retrieving data from rules. preprint, 2005.

[HS05c] Z. H. Huang and Q. Shen. Preserving piece-wise linearity in fuzzy interpolation. In *Proc. of the 2005 UK Workshop on Computational Intelligence*, 2005.

[HS05d] Z. H. Huang and Q. Shen. Transformation based interpolation with generalized representative values. In *Proc. of the International Conference on Fuzzy Systems*, pages 821 – 826, 2005.

[HS06] Z. H. Huang and Q. Shen. Fuzzy interpolation reasoning via scale and move transformations. *IEEE Trans. on Fuzzy Systems*, 14(2):340– 359, 2006.

[Jan98] C. Z. Janikow. Fuzzy decision trees: issues and methods. *IEEE Transaction on Systems, Man and Cybernetics*, 28(1):1–14, 1998.

[Jen01] S. Jenei. Interpolating and extrapolating fuzzy quantities revisited – a axiomatic approach. *Soft Computing*, 5, 2001.

[Jen04] R. Jensen. *Combining rough and fuzzy sets for feature selection*. PhD thesis, School of Informatics, University of Edinburgh, 2004.

[Jin99] Y. Jin. Fuzzy modelling of high-dimensional systems: Complexity reduction and interpretability improvement. *IEEE Trans. Fuzzy Systems*, 29(1):13–24, 1999.

[JKK02] S. Jenei, E. P. Klement, and R. Konzel. Interpolation and extrapolation of fuzzy quantities – the multiple-dimensional case. *Soft Computing*, 6(3 – 4):258 – 270, 2002.

[JKTV97] I. Joó, L. T. Kóczy, D. Tikk, and P. Várlaki. Stability of interpolative fuzzy kh-controllers. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 93 – 97, 1997.

[Jol86] I. T. Jolliffe. *Principal component analysis*. Springer Series in Statistics, Springer- Verlag, Berlin, 1986.

[JS04a] R. Jensen and Q. Shen. Fuzzy-rough attribute reduction with application to web categorization. *Fuzzy Sets and Systems*, 141(3):469–485, 2004.

[JS04b] R. Jensen and Q. Shen. Semantics-preserving dimensionality reduction: rough and fuzzy-rough based approaches. *IEEE Transactions on Knowledge and Data Engineering*, 16(12):1457–1471, 2004.

[JSM97] J. S. R. Jang, C. T. Sun, and E. Mizutani. *Neuro-fuzzy and soft computing*. Englewood Cliffs, NJ: Prentice Hall, 1997.

[KB95] U. Kaymak and R. Babuška. Compatible cluster merging for fuzzy modelling. In *Proc. IEEE Int. Conf. on Fuzzy Syst.*, volume 2, pages 897–904, 1995.

[KC96] S. Kawase and Q. Chen. On fuzzy reasoning by kóczy's linear rule interpolation. Technical report, 1996.

[KF92] R. Krishnapuram and C. P. Freg. Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recognition*, 25(4):385–400, 1992.

[KH93a] L. T. Kóczy and K. Hirota. Approximate reasoning by linear rule interpolation and general approximation. *Internat. J. Approx. Reasoning*, 9(3):197–225, 1993.

[KH93b] L. T. Kóczy and K. Hirota. Interpolation in structured fuzzy rule bases. In *IEEE International Conference on Fuzzy Systems*, volume 1, pages 402 – 406, 1993.

[KH93c] L. T. Kóczy and K. Hirota. Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Inform. Sci.*, 71(1):169–201, 1993.

[KH93d] L. T. Kóczy and K. Hirota. Ordering, distance and closeness of fuzzy sets. *Fuzzy Sets and Systems*, 59(3):281 – 293, 1993.

[KH97] L. T. Kóczy and K. Hirota. Size reduction by interpolation in fuzzy rule bases. *IEEE Trans, Syst., Man, Cybern.*, 27(1):14–25, 1997.

[KHG97] L. T. Kóczy, K. Hirota, and T. D. Gedeon. Fuzzy rule interpolation by the conservation of relative fuzziness. Technical Report 97/2, 1997.

[KHM91] L. T. Kóczy, K. Hirota, and L. Muresan. Rule interpolation by α-level sets in fuzzy approximate reasoning. *Journal of BUSEFAL*, 46, 1991.

[KHM00] L. T. Kóczy, K. Hirota, and L. Muresan. Interpolation in hierarchical fuzzy rule bases. In *Proceedings of International Conference on Fuzzy Systems*, volume 1, pages 471 – 477, 2000.

[KJS02] O. Kaynak, K. Jezernik, and A. Szeghegyi. Complexity reduction of rule based models: a survey. In *Proc. IEEE Int. Conf. on Fuzzy Syst.*, volume 2, pages 1216–1221, 2002.

[KK92] S. G. Kong and B. Kosko. Adaptive fuzzy systems for backing up a truck-and-trailer. *IEEE Trans. on Neural Networks*, 3(5):211–223, 1992.

[KK93] L. T. Kóczy and Sz. Kovács. On the preservation of the convexity and piecewise linearity in linear fuzzy rule interpolation. Technical Report 93-94/402, 1993.

[KK94a] L. T. Kóczy and Sz. Kovács. The convexity and piecewise linearity of the fuzzy conclusion generated by linear fuzzy rule interpolation. *Journal of BUSEFAL*, 1994.

[KK94b] L. T. Kóczy and Sz. Kovács. Linearity and the cnf property in linear fuzzy rule interpolation. In *Proceeding of International Conference on Fuzzy Systems*, volume 2, pages 870 – 875, 1994.

[KK94c] L. T. Kóczy and Sz. Kovács. Shape of the fuzzy conclusion generated by linear interpolation in trapezoidal fuzzy rule bases. In *European Congress on Intelligent Techniques and Soft Computing*, pages 1666 – 1670, 1994.

[KK97a] Sz. Kovács and L. T. Kóczy. Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rulebase. In *Proceeding IEEE International Conference on Engineering Systems*, pages 63 – 68, 1997.

[KK97b] Sz. Kovács and L. T. Kóczy. Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rulebase as a practical alternative of the classical cri. In *Proceeding of International Fuzzy Systems Association World Congress*, pages 144 – 149, 1997.

[KK97c] Sz. Kovács and L. T. Kóczy. The use of the concept of vague environment in approximate fuzzy reasoning. *Bratislava, Slovakia: Tatra Mountains, Math. Inst. Slovak Acad. Sco.*, 12, 1997.

[KM98] M. F. Kawaguchi and M. Miyakoshi. Fuzzy spline interpolation in sparse fuzzy rule bases. In *Proc. Int. Conf. Soft Computing*, pages 664 – 667, 1998.

[KM00a] M. F. Kawaguchi and M. Miyakoshi. A fuzzy rule interpolation technique based on b-splines in multiple input systems. In *Proc. Int. Conf. on Fuzzy Systems*, pages 488 – 492, 2000.

[KM00b] M. F. Kawaguchi and M. Miyakoshi. Fuzzy spline functions and nonlinear interpolation of if-then rules. In *Proc. Int. Conf. Soft Computing*, pages 785 – 792, 2000.

[KMK97] M. F. Kawaguchi, M. Miyakoshi, and M. Kawaguchi. Linear interpolation with triangular rules in sparse fuzzy rule bases. In *Proc. IFSA World Congr.*, pages 138 – 143, 1997.

[KR92] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of Ninth National Conference on Artificial Intelligence*, volume 1, pages 129 – 134, 1992.

[KS96] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceeding of International Conference on Machine Learning*, pages 284–292, 1996.

[LM98] H. Liu and H. Motoda. Feature transformation and subset selection. *IEEE Intelligent Systems*, 13(2):26–28, 1998.

[Lot00] A. Lotfi. Fuzzy inference system toolbox for matlab (fismat). 2000. Available at `http://www.lotfi.net/research/fismat/index.html`.

[LS96a] H. Liu and R. Setiono. Feature selection and classification – a probabilistic wrapper approach. In *Proceedings of the Ninth International Conference on Industrial and Engineering Applications of AI and ES.*, pages 419 – 424, 1996.

[LS96b] H. Liu and R. Setiono. A probabilistic wrapper approach to feature selection – a filter solution. In *Proceedings of the International Conference on Industrial and Engineering Applications of AI and ES*, pages 284 – 292, 1996.

[MA75] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man Machine Studies*, 7, 1975.

[MBKK99] S. Mizik, P. Baranyi, P. Korondi, and L. T. Kóczy. Comparison of α-cut, modified α-cut, vkk and the general fuzzy interpolation for the widely popular cases of triangular sets. In *Pro. EFDAN*, pages 165 – 172, 1999.

[MDS90] M. Mukaidono, L. Ding, and Z. Shen. Approximate reasoning based on revision principle. In *Proc. North American Fuzzy Information Processing Society*, volume 1, pages 94 – 97, 1990.

[Men95] J. M. Mendel. Fuzzy logic systems for engineering: a tutorial. In *Proceedings of the IEEE*, volume 83, pages 345 – 377, 1995.

[Mit97] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[Miz01] S. Mizik. Fuzzy rule interpolation techniques in comparison. *MFT Periodika 2001-04, Hungarian Society of IFSA, Hungary*, 2001. Available at `http://www.mft.hu`.

[MJ92] J. Mao and A. K. Jain. Artificial neural networks for nonlinear projection of multivariate data. In *Proc. IEEE Int. Joint Conf. Neural Networks*, volume 3, pages 59–69, 1992.

[MJ95] J. Mao and A. K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Trans. on Neural Networks*, 6(2):296–317, 1995.

[MM96] G. C. Mouzouris and J. M. Mendel. Designing fuzzy logic systems for uncertain environments using a singular-value-qr decomposition methods. *IEEE Int. Conf. Fuzzy Systems*, 1, 1996.

[MSK99] S. Mizik, D. Szabó, and P. Korondi. Survey on fuzzy interpolation techniques. *Proc. IEEE Int. Conf. Intelligent Engineering Systems*, 1999.

[MZ82] M. Mizumoto and H. J. Zimmermann. Comparison of fuzzy reasoning methods. *Fuzzy Sets and Systems*, 8, 1982.

[NMM96] K. C. Nisbet, B. Mulgrew, and S. McLaughlin. Reduced state methods in nonlinear prediction. *Signal Processing*, 48(1):37–49, 1996.

[NW90] D. Nguyen and B. Widrow. The truck backer-upper: An example of self-learning in neural network. *IEEE Contr. Syst. Mag.*, 10(2):18–23, 1990.

[PE98] N. R. Pal and V. K. Eluri. Two efficient connectionist schemes for structure preserving dimensionality reduction. *IEEE transaction on Neural Networks*, 9(6):1142–1154, 1998.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[Ped92] W. Pedrycz. *Fuzzy Control and Fuzzy Systems*. 2nd Ed. New York: Wiley, 1992.

[PEM02] N. R. Pal, V. K. Eluri, and G. K. Mandal. Fuzzy logic approaches to structure preserving dimensionality reduction. *IEEE Transaction on Fuzzy Systems*, 10(3):277 – 286, 2002.

[QMY96] W. Z. Qiao, M. Mizumoto, and S. Y. Yan. An improvement to kóczy and hirota's interpolative reasoning in sparse fuzzy rule bases. *Internat. J. Approx. Reasoning*, 15(3):185–201, 1996.

[Qui86] J. R. Quinlan. Induction on decision trees. *Machine Learning*, 1(1):81–106, 1986.

[Qui87] J. R. Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27(3):221–234, 1987.

[RNe96] C. E. Rasmussen, R. M. Neal, and etc. Data for evaluating learning in valid experiments (delve), 1995-1996. Available at `http://www.cs.toronto.edu/~delve/`.

[Rog71] A. Rogers. *Matrix methods in urban and regional analysis*. San Francisco: Holden-Day, 1971.

[RR01] A. Riid and E. Rustern. Fuzzy logic control: truck backer-upper problem revised. In *IEEE International Conference on Fuzzy Systems*, volume 1, pages 513–516, 2001.

[RZ93] G. V. S. Raju and J. Zhou. Adaptive hierarchical fuzzy controller. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4):973–980, 1993.

[RZK90] G. V. S. Raju, J. Zhou, and R. A. Kisner. Fuzzy logic control for steam generator feedwater control. In *Proc. Amer. Contr. Conf.*, pages 1491 – 1493, 1990.

[RZK91] G. V. S. Raju, J. Zhou, and R. A. Kisner. Hierarchical fuzzy control. *International Journal Control*, 54(5):1201–1216, 1991.

[Sam69] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409, 1969.

[SBKL98] M. Setnes, R. Babuska, U. Kaymak, and H. R. N. Lemke. Similarity measures in fuzzy rule base simplification. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):376–386, 1998.

[SDM88] Z. Shen, L. Ding, and M. Mukaidono. Fuzzy resolution principle. In *Proc. of the Eighteenth International Symposium on Multiple-Valued Logic*, pages 210 – 215, 1988.

[SDM93] Z. Shen, L. Ding, and M. Mukaidono. Methods of revision principle. In *Proc. of the 5th International Fuzzy Systems Association Congress*, pages 246 – 249, 1993.

[SGB93] M. Sugeno, M. F. Griffin, and A. Bastian. Fuzzy hierarchical control of an unmanned helicopter. In *International Fuzzy Systems and Applications conference*, pages 179–182, 1993.

[Sim00] D. Simon. Design and rule base reduction of a fuzzy filter for the estimation of motor currents. *International Journal of Approximate Reasoning*, 25(2):145–167, 2000.

[SK88] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1):15–33, 1988.

[SL97] R. Setiono and H. Liu. Neural network feature selector. *IEEE Transactions on Neural Networks*, 8(3):645 – 662, 1997.

[SM95] Y. Shi and M. Mizumoto. Some considerations on kóczy's interpolative reasoning method. In *IEEE International Conference on Fuzzy Systems*, volume 4, pages 2117–2122, 1995.

[Tao01] C. W. Tao. Comments on “reduction of fuzzy rule base via singular value decomposition”. *IEEE Tran. on Fuzzy Systems*, 9(4):675 – 677, 2001.

[TB00] D. Tikk and P. Baranyi. Comprehensive analysis of a new fuzzy rule interpolation method. *IEEE Tran. on Fuzzy Systems*, 8(3):281 – 296, 2000.

[TBGM01] D. Tikk, P. Baranyi, T. D. Gedeon, and L. Muresan. Generalization of the rule interpolation method resulting always in acceptable conclusion. *Bratislava, Slovakia: Tatra Mountains, Math. Inst. Slovak Acad. Sco.*, 21, 2001.

[TBYk99] D. Tikk, P. Baranyi, Y. Yam, and L. T. kóczy. Stability of a new interpolation method. In *Proc. IEEE conference on System, Man and Cybernetics*, volume 3, pages 7 – 9, 1999.

[Tik99] D. Tikk. *Investigation of fuzzy rule interpolation and the universal approximator property of fuzzy controllers*. PhD thesis, Department of Telecommunications and Telematics, 1999.

[TkM97] D. Tikk, L. T. kóczy, and B. Moser. Stability of interpolative fuzzy kh-controllers, multi-dimensional case. In *Proc. Symp. System Modelling, Fault Diagnosis and Fuzzy Logic Control*, volume 8, 1997.

[Tor52] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17, 1952.

[TS85] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1):116–132, 1985.

[TZ88] I. B. Turksen and Z. Zhong. An approximate analogical reasoning approach based on similarity measures. *IEEE Transaction on Systems, Man, and Cybernetics*, 18(6):1049 – 1056, 1988.

[UOHT94] M. Umano, H. Okamoto, I. Hatono, and H. Tamura. Fuzzy decision trees by fuzzy id3 algorithm and its application to diagnosis systems. In *IEEE World Congress on Computational Intelligence*, volume 3, pages 2113 – 2118, 1994.

[VKK92] G. Vass, L. Kalmar, and L. T. Kóczy. Extension of the fuzzy rule interpolation method. In *Int. Conf. Fuzzy Sets Theory Applications*, pages 1–6, 1992.

[Wan98] L. X. Wang. Universal approximation by hierarchical fuzzy systems. *Fuzzy Sets and Systems*, 93(2):223 – 230, 1998.

[WF99] I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, 1999.

[WGF00] K. W. Wong, T. D. Gedeon, and C. C. Fung. Using modified α-cut based fuzzy interpolation in petrophysical properties prediction. In *Proc. 4th Japan-Australia Joint Workshop on Intelligent and Evolutionary Systems*, pages 136 – 142, 2000.

[WGT00] K. W. Wong, T. D. Gedeon, and D. Tikk. An improved multidimensional α-cut based fuzzy interpolation technique. In *Proc. Int. Conf. Artificial Intelligence in Science and Technology*, pages 29 – 32, 2000.

[WL95] L. Wang and R. Langari. Building sugeno-type models using fuzzy discretization and orthogonal parameter estimation techniques. *IEEE Trans. on Fuzzy Syst.*, 3(5):454–458, 1995.

[WM79] D. Willaeys and N. Malvache. Use of fuzzy model for process control. In *IEEE Proc. Int. Conf. Cybern. Soc*, pages 942 – 946, 1979.

[WM92a] L. X. Wang and J. M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, 3(5):807–814, 1992.

[WM92b] L. X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, 22(6):1414–1427, 1992.

[XL02] N. Xiong and L. Litz. Reduction of fuzzy control rules by means of premise learning - method and case study. *Fuzzy Sets and Systems*, 132(2):217 – 231, 2002.

[Yam97] Y. Yam. Fuzzy approximation via grid point sampling and singular value decomposition. *IEEE Transactions on Systems, Man and Cybernetics*, 27(6):933–951, 1997.

[YBTK99] Y. Yam, P. Baranyi, D. Tikk, and L. T. Kóczy. Eliminating the abnormality problem of α-cut based interpolation. In *Proc. 8th IFSA World Congress*, volume 2, pages 762 – 766, 1999.

[YBY99] Y. Yam, P. Baranyi, and C. Yang. Reduction of fuzzy rule base via singular value decomposition. *IEEE Trans. Fuzzy Systems*, 7(2):120–132, 1999.

[YK98] Y. Yam and L. T. Kóczy. Cartesian representation for fuzzy interpolation. In *IEEE Conference on Decision and Control*, volume 3, pages 2936–2937, 1998.

[YK00] Y. Yam and L. T. Kóczy. Representing membership functions as points in high-dimensional spaces for fuzzy interpolation and extrapolation. *IEEE Transaction on Fuzzy Systems*, 8(6):761–772, 2000.

[YK01] Y. Yam and L. T. Kóczy. Fuzzy interpolation with cartesian representation and extensibility functions. In *Pro. International Fuzzy Systems Association Congr.*, pages 2852 – 2857, 2001.

[YKN00] Y. Yam, V. Kreinovich, and H. T. Nguyen. Extracting fuzzy sparse rules by cartesian representation and clustering. In *Pro. International Conference on Systems, Man and Cybernetics*, pages 3778 – 3783, 2000.

[YMQ95] S. Yan, M. Mizumoto, and W. Z. Qiao. Reasoning conditions on kóczy's interpolative reasoning method in sparse fuzzy rule bases. *Fuzzy Sets and Systems*, 75(1):63–71, 1995.

[YS95] Y. Yuan and M. J. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, 69(2):125–139, 1995.

[YW96] J. Yen and L. Wang. An svd-based fuzzy model reduction strategy. In *IEEE International Conference on Fuzzy Systems*, volume 2, pages 835–841, 1996.

[YW97] J. Yen and L. Wang. Simplification of fuzzy rule based systems using orthogonal transformation. In *IEEE International Conference on Fuzzy Systems*, volume 1, pages 253–258, 1997.

[YW99] J. Yen and L. Wang. Simplifying fuzzy rule-based models using orthogonal transformation methods. *IEEE transactions on Systems, Man, and Cybernetics*, 29(1):13–24, 1999.

[YWB00] Y. Yam, M. L. Wong, and P. Baranyi. Interpolation as mappings between cartesian spaces. In *Pro. Int. Conf. Intelligent Technologies*, volume 3, pages 459 – 466, 2000.

[Zad65] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8, 1965.

[Zad73] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transaction on System, Man and Cybernetics*, 3, 1973.

[ZK04] X. J. Zeng and J. A. Keane. Approximation capability analysis of hierarchical takagi-sugeno fuzzy systems. In *IEEE International Conference on Fuzzy Systems*, volume 3, pages 1227–1232, 2004.