Curvature-based Sparse Rule Base Generation for Fuzzy Rule Interpolation

Yao Tan

Supervisors: Dr Longzhi Yang (principal)
Dr Hubert Shum (second)
Dr Fei Chao (external)

Faculty of Engineering and Environment
University of Northumbria at Newcastle

This thesis is submitted for the degree of
Doctor of Philosophy

February 2020

Dedication

I would like to dedicate this thesis to my loving parents Xingde Tan and Yongrong Cheng.

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others. All material in this thesis which is not of my own work has been identified and properly attributed.

Any ethical clearance for the research presented in this thesis has been approved. Approval has been sought and granted by the Faculty Ethics Committee / University Ethics Committee on [25/01/2016].

I declare that the Word Count of this Thesis is 43,568 words. Signature:

Yao Tan
February 2020

Acknowledgements

I would like to express my enormous gratitude to my supervisors, Dr Longzhi Yang, Dr Hubert P. H. Shum and Dr Fei Chao, for their unfailing motivation and guidance throughout the course of my research. Additionally, I am indebted to the following colleagues or individuals for their support during this period: Dr Jie Li, Dr Yang Long, Dr Martin Wonders , Dr Ling Shao, Zheming Zuo, Noe Elisa, Daniel Organisciak, for some inspiration or cooperation for this research; and Dr Li Zhang for her encouragement. I would like to also thank all other researchers in the CICV Group, both past and present, for stimulating discussion, insight, and helpful advice.

I would also like to acknowledge my gratitude to the staff—academic, administrative, technical, and support of the Department of Computer Science at Northumbria University for providing such a positive and nurturing environment in which to conduct my research. Moreover, the completion of the research would not have been possible without the generous financial support of not only the department, but also the Chengdu Yubo Information Technology Co. Ltd. which provided the Collaborative Studentship.

Many of the arguments made here were first aired as conference papers. Through these channels, my initial idea was discussed, and further developed and extended. I record my thanks to these generous interlocutors.

Lastly, the completion of this PhD would not have been possible without the support and encouragement of my family: my parents Xingde Tan and Yongrong Cheng, my sister Ying Tan, my dear wife Yujin Zheng, my lovely son Feng Tan, and all my relatives and friends. My heartfelt thanks to them all.

Abstract

Fuzzy logic has been successfully widely utilised in many real-world applications. The most common application of fuzzy logic is the rule-based fuzzy inference system, which is composed of mainly two parts including an inference engine and a fuzzy rule base. Conventional fuzzy inference systems always require a rule base that fully covers the entire problem domain (i.e., a dense rule base). Fuzzy rule interpolation (FRI) makes inference possible with sparse rule bases which may not cover some parts of the problem domain (i.e., a sparse rule base). In addition to extending the applicability of fuzzy inference systems, fuzzy interpolation can also be used to reduce system complexity for over-complex fuzzy inference systems. There are typically two methods to generate fuzzy rule bases, i.e., the knowledge-driven and data-driven approaches. Almost all of these approaches only target dense rule bases for conventional fuzzy inference systems. The knowledge-driven methods may be negatively affected by the limited availability of expert knowledge and expert knowledge may be subjective, whilst redundancy often exists in fuzzy rule-based models that are acquired from numerical data. Note that various rule base reduction approaches have been proposed, but they are all based on certain similarity measures and are likely to cause performance deterioration along with the size reduction.

This project, for the first time, innovatively applies curvature values to distinguish important features and instances in a dataset, to support the construction of a neat and concise sparse rule base for fuzzy rule interpolation. In addition to working in a three-dimensional problem space, the work also extends the natural three-dimensional curvature calculation to problems with high dimensions, which greatly broadens the applicability of the proposed approach. As a result, the proposed approach alleviates the 'curse of dimensionality' and helps to reduce the computational cost for fuzzy inference systems. The proposed approach has been validated and evaluated by three real-world applications. The experimental results demonstrate that the proposed approach is able to generate sparse rule bases with less rules but resulting in better performance, which confirms the power of the proposed system. In addition to fuzzy rule interpolation, the proposed curvature-based approach can also be readily used as a general feature selection tool to work with other machine learning approaches, such as classifiers.

The proposed approach makes several contributions. For the first time, it utilises the curvature values in sparse rule base generation to support fuzzy inference systems. It provides an objective tool to distinguish the important things in a dataset; thus efficiently constructs a sparse rule base with a smaller number of important instances and features. In addition, it originally extends the three-dimensional curvature idea into high-dimensional problems; thus highly increases its applicability. To some extent, it also alleviates the 'curse of dimensionality' and helps to reduce the computational cost for sparse rule base generation by using a one-time process to carry out curvature calculation off-line. The proposed approach has been validated and evaluated by three real-world applications. The accuracy of the classification result by the proposed method are comparable or outperform the existing methods, using only several important instances with their several important features. Furthermore, in the zero-shot learning image classification application, the proposed method makes the image representation more interpretable and discriminative. The results demonstrated the power of the proposed system in reference to the conventional methods from both aspects of system effectiveness and efficiency.

Table of contents

List of figures

List of tables

List of Algorithms

Chapter 1

Introduction

1.1 Fuzzy Logic and Fuzzy Inference Systems

Fuzzy logic was firstly proposed in 1965 which provides an approximate reasoning approach based on partial membership, rather than simply true or false as used in the traditional Boolean Logic. Fuzzy sets and fuzzy logic theory provide an efficient way of handling vague information that arises due to the lack of sharp distinctions or boundaries between pieces of information. With the ability to effectively represent and reason human natural language, fuzzy logic theory is considered as an advanced methodology in the field of control systems. It has been widely used in many real-world applications such as in science, engineering, business, psychology, medicine and other fields. Some early commercial applications of fuzzy systems include: fuzzy automatic transmissions and fuzzy anti-skid braking systems developed by Nissan, auto-focusing cameras by Canon, digital image stabilisers for camcorders by Matsushita, hand-writing recognition systems by Hitachi, hand-printed character recognition systems by Sony, voice recognition systems by Ricoh and Hitachi, stock-trading portfolio systems used in Tokyo stock market, Sendai station subway control systems in Japan, and so on [5–19].

Fuzzy logic is a mathematical approach to problem solving. It performs exceptionally well in producing exact results from imprecise or incomplete data. Fuzzy logic differs from classical logic in that its statements are no longer simply true or false. In traditional logic a variable takes on a value with a certainty degree or truth measure of either 0 or 1; in fuzzy logic, a variable can assume a value with any degree between 0 and 1, representing the extent (membership) to which an element belongs to a given concept. Human brains can reason with uncertainties, vagueness, and judgements. Computers can only manipulate precise valuations. Fuzzy logic is an attempt to combine the two [20].

In real life, fuzzy logic provides the means to compute with words. Using fuzzy logic, experts are no longer forced to summarise and express their knowledge in a language that machines or computers can understand. What traditional expert systems failed to achieve may be realised through the use of fuzzy expert systems. Fuzzy logic comprises fuzzy sets and fuzzy (linguistic) variables. It is a way of representing non-statistical uncertainty and performing approximate reasoning, which includes operations that implement knowledge-based inferences. A fuzzy (linguistic) variable, as its name suggests, is a variable whose values are words rather than numerical numbers, for example, S (small), M (medium), L (large), L (low), H (high), or even 'very hot', 'rather old' and 'quite slow'.

A fuzzy set is distinct from a crisp set in that it allows its elements to have a degree of membership. The core of a fuzzy set is its membership function that defines the relationship between a value in the domain of set and its degree of membership. Membership of an element x to a fuzzy set A, denoted as $\mu_A(x)$, can vary from 0 (full non-membership) to 1 (full membership), i.e., it can assume all values in the interval [0,1]. The value of $\mu_A(x)$ describes a degree of membership of x in A. Clearly, a fuzzy set is a generalisation of the concept of a classical set whose membership function takes on only two values 0 or 1, as shown in Fig. 1.1. Each fuzzy set defines a portion of the domain of variable. However, this portion is not uniquely defined. Fuzzy sets overlap as a natural consequence of their elastic boundaries.

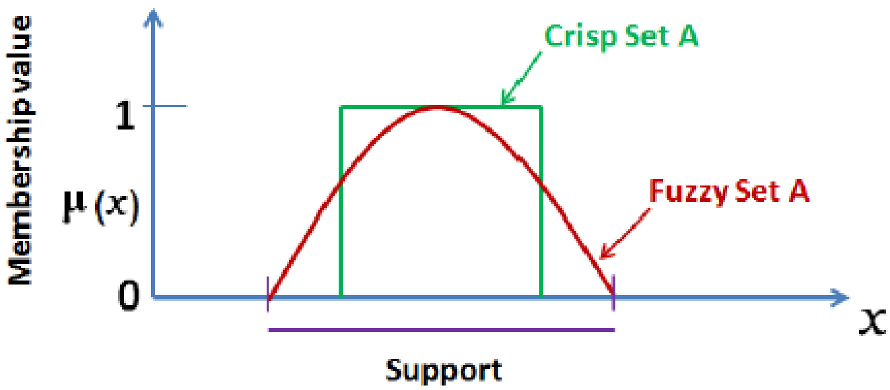

Fig. 1.1 Fuzzy set

Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, time series prediction, robotics, and computer vision [5–19]. The most common fuzzy model is the rule-based fuzzy inference systems, which is mainly composed of two parts: a rule base (or knowledge base) and an inference engine, as shown as the Fig. 1.2. The rule base is usually generated in two ways. The first way is directly transferring human expert knowledge into rules. The second way is using machine learning approaches to extract rules from data. No matter which approach is employed to generate the rule base, the resulting rule base has to cover the entire input domain.

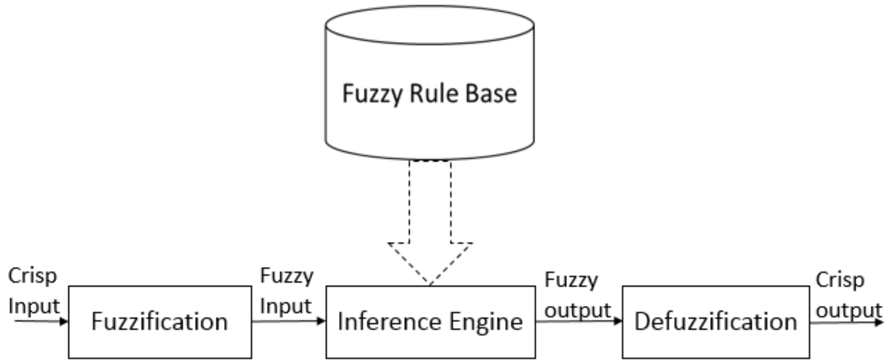

Fig. 1.2 The rule-based fuzzy inference system

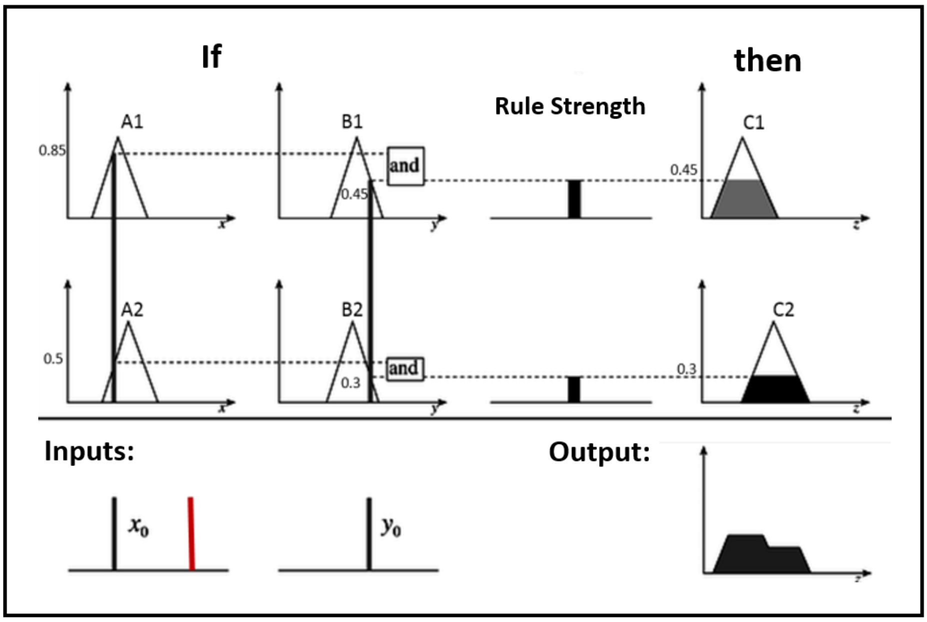

Fig. 1.3 Limitation of conventional fuzzy inference systems

A common feature of all the classical fuzzy inference systems is that they are only applicable to problems with dense rule bases, by which the entire input domain must be fully covered (the input space is completely covered by the rule premises). When a given observation does not overlap with any rule antecedent in the rule base (which usually termed as a sparse rule base), no rule can be fired, and thus no result can be generated.

This is an example of Mamdani inference with two inputs and one output, as shown as the Fig. 1.3. Suppose that there are two rules. The first rule is

$$\textbf{If } x = A_1 \textit{ and } y = B_1, \textbf{ then } z = C_1, \tag{1.1}$$

and the second rule is

$$\textbf{If } x = A_2 \textit{ and } y = B_2, \textbf{ then } z = C_2. \tag{1.2}$$

Given an input with x is equal to x_0 and y is equal to y_0. The Mamdani inference firstly calculates the matching degree of the inputs. For example, from the input x_0, the matching degree of the antecedent of first rule is about 0.85, and the matching degree of x_0 with the antecedent of second rule is about 0.5. Similarly, the matching degree of the input for variable y can be calculated, which is about 0.45 for the first rule and about 0.3 for the second rule.

From this the firing degree of each rule is computed using a T-norm. Particularly for Mamdani inference, the T-norm operator is implemented using 'minimum', or 'and'. For example, for the first rule, the minimum of 0.85 and 0.45 is 0.45, then the firing strength or rule strength is 0.45. From this, the consequence is computed as the shadow part of line 1. Using the same way, the consequence from the second rule is the shadow part of line 2. The final consequence is the combination of these two intermediate results through the logic union operation. In this example, the final result is the combined shadow part of line 3.

From the above example, it is clear that no rule will be fired if a given input is not covered by any rule antecedents in the rule base. In other words, the rule base has to cover the entire input domain such that a logic consequence can be generated from any input. In order to address this problem, the following section 1.2 Fuzzy rule interpolation (FRI) was proposed.

1.2 Fuzzy Rule Interpolation

As shown as in Fig. 1.4, a tomato problem was initially proposed by Mizumoto and Zimmerman [21]. Suppose there are only two rules: if the tomato is green, then it is unripe; and if the tomato is red, then it is ripe; Even though there is no rule if the tomato is yellow then what it is (conclusion like ripe or unripe), the result can still be achieved by FRI based on its neighbouring rules.

FRI not only can solve the issues when inputs are not covered by any rule antecedents. It can also be employed to reduce the complexity of complex fuzzy models by excluding those rules that can be approximated by their neighbouring ones. A number of important fuzzy rule interpolation methods have been proposed in the literature [22–26], which have been successfully applied to deal with real-world problems [27–33]. Some other FRI methods are also developed and discussed, such as [34–39].

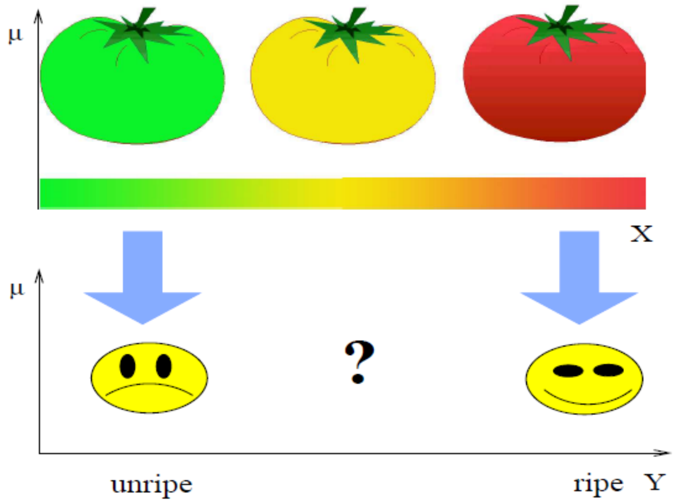

Fig. 1.4 Fuzzy rule interpolation example [1]

Fuzzy rule interpolation alleviates the problem of lack of expertise or data for rule base generation, as FRI enables the performance of inference upon sparse rule bases. Fuzzy rule interpolation strengthens the power of fuzzy inference. When observations do not overlap with any rule antecedent values, traditional fuzzy inference systems will not be applicable, as no rule can be fired. However, fuzzy rule interpolation can still generate a conclusion through a sparse rule base, thus improve the applicability of fuzzy models. Also, with the help of fuzzy interpolation, the complexity of an inference system can be reduced by omitting those rules which may be approximated with their neighbouring ones.

1.3 Rule Base Generation and Reduction

The fuzzy rule base (set of fuzzy rules) is the core part of a fuzzy inference system that contains knowledge that is utilised by the reasoning mechanism of the system. If the rule base covers the entire input domain then one of the possible reasoning mechanisms would be compositional rule of inference (CRI). However, if the rule base does not cover the entire input domain then fuzzy rule interpolation (FRI) may offer a potential solution. This thesis mainly focus on curvature-based sparse rule base generation for FRI.

Although a dense fuzzy rule base is not required by FRI, a sparse rule base is still needed. Fuzzy rule base generation has been intensively studied in literature and is usually implemented in one of two ways: data-driven (extracting rules from data) [40–42] and knowledge-driven (generating rules from human expert knowledge) [43]. Both approaches may suffer from the 'curse of dimensionality', which means the number of rules will grow exponentially while the number of input variables and the number of fuzzy terms increasing. In addition, the knowledge-driven method may be further negatively affected by the limited availability of expert knowledge. Data-driven rule base generation was proposed to minimise the involvement of human expertise. The success of data-driven approaches is built upon a large quantity of training data. In this work only the data-driving methods are mainly discussed.

No matter which kind of approach (knowledge-driven or data-driven) is used to generate rule bases, all the above approaches usually only target dense rule bases that are used for traditional fuzzy inference approaches [44–48]. In addition, redundancy often exists in fuzzy rule-based models that are acquired from numerical data. This results in unnecessary structural complexity and reduces the interpretability of the system.

In order to reduce the complexity of such rule bases, various rule base reduction approaches have been developed to minimise the redundancy [49–61]. According to [62] two types of rule redundancy can be distinguished: overlap redundancy and interpolation redundancy. In this work, only the interpolation redundancy is treated, considering that choosing a uniformly partitioned universe of discourse does not lead to high levels of overlapping between the fuzzy rules. Most of existing rule base generation and reduction approaches are based on certain similarity measures; therefore, they are likely to cause performance deterioration along with the size reduction of the rule base. Usually a fuzzy clustering technique associated with some fuzzy partition are used to extract the initial fuzzy rule-base and find out the optimal number of fuzzy rules.

1.4 Existing Issues and Motivation

Fuzzy rule interpolation (FRI) [63] was initially proposed to address above limitation due to its ability to work with a spare rule base. When system inputs or observations do not overlap with any rule antecedent values, traditional fuzzy inference systems are not applicable as no rule can be fired. However, fuzzy rule interpolation can still generate a conclusion through a sparse rule base, thereby improving the applicability of fuzzy models. Fuzzy rule interpolation provides a tool for specifying an output fuzzy set even when one or all of the input spaces are sparse.

The most common application of fuzzy logic is the rule-based fuzzy inference system, which is composed of mainly two parts: an inference engine, and a fuzzy rule base. The fuzzy rule base is usually generated in two ways. The first way is directly transferring human expert knowledge into rules. The second way is using machine learning approaches to extract rules from data. In early-stage, a fuzzy rule base is traditionally built upon human expertise, which greatly limits the system modelling as experts may not always be available. Nowadays the data-driven rule base generation was proposed to minimise the involvement of human expertise, in an effort to automate the generation of rule bases during the system modelling process. They both may suffer from the 'curse of dimensionality', which means as many attributes and attribute values need to be considered, the number of rules grows exponentially with the number of input variables and the number of fuzzy terms. And almost all of these approaches are targeting dense rule bases for traditional fuzzy inference approaches, which may be low efficient. In addition, the knowledge-driven method may further be negatively affected by the limited availability of expert knowledge. And expert knowledge may be subjective, even in the same situation it can get different solutions as different experts have different opinions on what is important. On the other hand, redundancy often exists in fuzzy rule-based models that are acquired from numerical data. This work mainly concerns about the data-driven methods for fuzzy rule base generation, proposing a novel data-driven rule base generation approach for FRI, which is able to efficiently generate a sparse rule base from data.

Furthermore, most of existing rule base generation and reduction approaches are based on certain similarity measures; therefore, they are likely to cause performance deterioration along with the size reduction of the rule base. That means, existing methods lack a more accurate and efficient way to distinguish important rules (or important instances and their important features).

Therefore, to solve about issues, more study should be invested in how to keep a good balance between model accuracy, efficiency and transparency, i.e., how to generate a sparse rule base which is very accurate, computationally efficient, and linguistically tractable. So the aim of project is to develop a novel fuzzy rule base generation/reduction approach for fuzzy rule interpolation and apply it to real-world applications.

1.5 Curvature-based Rule Base Generation

Different to the conventional fuzzy rule base generation approaches, the proposed work discriminates rules by calculating their curvature values. Note that curvature values are only workable in three-dimensional spaces (or a rule with two antecedents and one consequence) and thus cannot be directly used for higher-order problems. As a solution, for any given higher-order problem, the proposed approach firstly decomposes the higher-order space into a number of three-dimensional spaces, and then approximates the importance of the higher-order spaces by aggregating the curvature values of the corresponding decomposed three-dimensional ones. From this, the most important rules are selected to form a raw rule base, which is then optimised using a general optimisation approach, such as the genetic algorithm.

Note that the proposed method provides an important key to solve the issues of ‘curse of dimensionality’. The difficulty of feature selection lies in how to find good clues to select feature variables that can most efficiently represent the intrinsic characteristics of the data. Its essence is a complex combinatorial optimization problem. Irrespective of all desirable features of fuzzy rule interpolation techniques the success of a particular method depends on its computational complexity, which should be minimised. For example, if there are n feature variables, each feature variable has two possible states when we are modelling: “selected" and “omitted". Then the number of elements in the sets of feature combination dimensions is 2^n. If n = 3, the traditional exhaustive methods need 8 combinations. However, use the proposed curvature based method the curse of dimensionality is greatly reduced as only need 3 combinations. That means, it reduced the traditional feature combination dimensions from 2^n to $C_n^2 = \frac{n!}{2!(n-2)!}$, when n is large the reduction number is very huge. For example, If n = 20, the proposed method reduces the combinations dimensions from 1048576 to only 190, which is less than the 0.02% of traditional one.

The proposed approach is validated and evaluated by one experiment and three applications; the results demonstrate that the proposed approach is promising. The contribution of this PhD project are summarised below:

1. Applying curvature values for rule base generation and thus leading to a novel curvature-based rule base generation approach;
2. Providing an objective tool to distinguish important instances and features, and thus to ease the subjectiveness of expert knowledge;
3. Alleviating the ‘curse of dimensionality’ by generating sparse rule bases with a smaller number of rules;
4. Applying the proposed curvature-based fuzzy rule base generation approach to three real-wold applications with promising results demonstrated.

1.6 Structure of Thesis

This subsection outlines the structure of the remainder of the thesis.

Chapter 2 introduces the theoretical underpinnings of fuzzy inference systems, fuzzy rule interpolation, rule base generation and previously curvature utilisation, which this work is built upon. It provides a comprehensive review of typical FRI methods that have been developed in the last two decades.

Chapter 3 presents the proposed curvature-based sparse rule base generation approach. The inference problems with two inputs and one output (referred to as the basic case) is considered first, followed by the general situation with multiple inputs (referred to as the general case).

Chapter 4 details the experimentation for demonstration and validation. The proposed method was evaluated using a synthetic dataset and three real-world applications, i.e., an indoor environment localisation problem, a student knowledge level evaluation problem, and the zero-shot learning image classification problem. The experiment demonstrates the working procedure of the proposed approach, while the applications show the power of the proposed approach in solving real-world problems.

Chapter 5 concludes the thesis and suggests probable future developments. The thesis has proposed a curvature-based sparse rule base generation method to support fuzzy rule interpolation. It first fuzzy partitions the problem domain either into a number of hypercubes for problems with dense datasets, or by representing each data instance as a hypercube if only a small dataset is available. From this, the hypercubes are discriminated by effectively using the curvature values and each important hypercube is represented as a fuzzy rule. The capabilities and potential of the developed applications have been experimentally validated, and compared with conventional work. The proposed approach has very comparative results in the experiments, which shows the potential of the proposed approach for a wider range of real-world applications. A number of initial thoughts about the directions for future research are also represented.

Appendices:

Appendix A lists the publications arising from the work presented in this thesis, containing both published conference papers, and the proceeding journal publication.

Appendix B summaries the acronyms employed throughout this thesis.

Chapter 2

Background

This chapter provides a literature review on fuzzy inference and fuzzy inference systems, fuzzy rule interpolation, rule base generation and reduction, and previously curvature utilisation. Fuzzy inference mimics the crucial ability of the human mind to summarise data and focus on decision-relevant information. Fuzzy inference is one of the most advanced technologies in control field. It has been widely applied to solve real-world problems due to its simplicity and effectiveness in representing and reasoning on human natural language. The most common application of fuzzy logic is the rule-based fuzzy inference system, which is composed of mainly two parts: an inference engine, and a fuzzy rule base. A number of inference engines have been developed, with the Mamdani method and the TSK method being the most widely used. The traditional fuzzy inference approaches require the entire domains of input variables to be fully covered by rules in a given rule base. Otherwise, no rule can be fired when a given observation does not overlap with any rule antecedent. Fuzzy rule interpolation strengthens the power of fuzzy inference. When given observations have no overlap with any rule antecedent values, fuzzy interpolation through a sparse rule base can still obtain certain conclusions and thus improve the applicability of fuzzy models.

The rest of this chapter is organised as follows. Section 2.1 explains the fuzzy inference systems or fuzzy rule-based systems. Section 2.2 briefly discusses fuzzy rule interpolation (FRI), which helps to understand why the proposed curvature based method works. Section 2.3 briefly discusses rule base generation and reduction, which is an important part of the proposed method. Section 2.4 discusses existing curvature utilisation, and introduces two types of methods for curvature calculation.

2.1 Fuzzy Inference and Fuzzy Inference Systems

2.1.1 Fuzzy Inference

Inference is the process of deriving logical conclusions from premises known or assumed to be true or partially true. When conclusions are derived from fuzzy linguistic variables using fuzzy set operators (AND, OR, NOT), then the process is called the approximate reasoning or fuzzy inference. Fuzzy inference is more effective and useful for those systems where a system cannot be defined in precise mathematical terms/models due to uncertainties, unpredicted dynamics and other unknown phenomena. In the real world much of the knowledge is naturally unclear, confusing, ambiguous, imprecise, and vague. Fuzzy inference mimics the crucial ability of the human mind to summarise data and focus on decision-relevant information.

Fuzzy inference is a convenient way to map a fuzzy input space onto a fuzzy output space. This mapping is implemented by using fuzzy If-Then rules. These rules (together with the fuzzy sets that the rules used) serve to partition the input and output spaces into fuzzy regions [64]. A fuzzy rule can be expressed in the form: **If** a set of conditions are satisfied **Then** a set of consequences can be inferred. The **If** part is called the antecedent and the **Then** part is called the consequent. The antecedent describes to what degree the rule applies, while the consequent assigns a fuzzy function to each of one or more output variables [65]. A fuzzy **If-Then** rules provide an easy means to express and capture human 'rule of thumb' type knowledge, because they are expressed with linguistic terms. A collection of fuzzy rules (rule base) can be derived from subject matter experts or extracted from data through a rule induction process. The rule base is the key component of a fuzzy inference system and such a system is also known as a fuzzy rule-based system (FRBS). Fuzzy inference is only possible in a dense rule base when the rule base is large enough to cover the complete input space, i.e., every input condition should be covered by at least one rule. The compositional inference and compatibility-modification inference are two main classical approaches to fuzzy inference [66]. Based on these basic inference principles, many inference methods have been proposed such as: Zadeh inference [67, 68], Mamdani inference [69, 70], Larsen inference [71], Sugeno inference [72, 73], and Tsukamoto inference [74, 75].

2.1.2 Fuzzy Inference Systems

The inference engines have been defined by different inference approaches, the most important two are the Mamdani model [76] and the TSK model [73]. Compared to the Mamdani approach, the TSK approach is more convenient when the crisp outputs are required. Although the TSK model is able to generate crisp output, the Mamdani fuzzy inference method is more intuitive and suitable for handling human natural language inputs, which is an implementation of the extension principle [77]. As fuzzy outputs are usually led by the Mamdani approach, a defuzzification approach, such as the centre of gravity method [78], has to be employed to map fuzzy outputs to crisp values for general system use. The TSK approach uses polynomials to generate the inference consequence, and it therefore is able to directly produce crisp values as outputs, which is often more convenient to be employed when the crisp values are required. Some interpretability measures are discussed in [79]. Precise Fuzzy Modelling (PFM) is mainly developed by means of Takagi-Sugeno system, which is focused on the accuracy and associated with the data-driven approach, resulting in very robust and accurate models [80]. Linguistic Fuzzy Modeling (LFM) is mainly developed by means of linguistic (or Mamdani) system, which is focused on the interpretability and associated with the expert-driven approach, resulting in very interpretable models [81]. Some hybrid learning models were proposed to get the interpretability–accuracy trade-off in fuzzy modeling, thus accuracy improvements are proposed in LFM whilst interpretability improvements are proposed in PFM, as reviewed in [82].

Fuzzy inference systems, which tend to mimic the inference behaviour of human, are a vital part of many successful knowledge-based systems in many field [17, 83]. The basic concepts which underlie fuzzy systems are fuzzy If-Then rules (it is assumed that the reader should know the fundamentals of fuzzy logic). Fuzzy inference describes systems by establishing relations between the relevant variables in the form of if-then rules. Fuzzy If-Then rules are of the general form: **If** antecedent(s) **Then** consequent(s), where antecedent and consequent are fuzzy propositions that contain linguistic variables. A fuzzy If-Then rule is exemplified by **If** the quality score is less than or equal to 12 **Then** the quality category is 'Not Reliable' [84], or **If** the temperature is high **Then** the fan-speed should be high [85].

2.1.3 Inference Engine

There are mainly two types of fuzzy engines, Madamni [76] and TSK [73], with different forms of the consequent part in the if-then rules. A multi-input and single-output (MISO) fuzzy model is represented as a collection of fuzzy rules in the following form:

$$\begin{aligned} R_i : \textbf{IF } x_1 = A_{i1} \text{ and } x_2 = A_{i2} \text{ and } x_n = A_{in}, \\ \textbf{THEN } y_i = z_i(x), \end{aligned} \tag{2.1}$$

where $z_i(x)$ takes different forms depending on the fuzzy inference engines. In the Mamdani engine, it is linguistic labels represented by fuzzy sets: $z_i(x) = B_i$; or in the TSK engine, it is often a linear function: $z_i(x) = b_{i0} + \sum_{j=1}^{s} b_{ij}x_j$.

The Mamdani engine is a particular implementation of the Compositional Rule of Inference, which is more intuitive and commonly utilised to deal with human natural language. Compared to the Mamdani engine, the TSK engine is more convenient when the crisp outputs are required. The main advantage of the TSK engine is that the resulting global model can be non-linear (of high order) while the local models can be as simple as necessary, since the local regions are fuzzily defined. Generally the local sub-models can be non-linear, however, usually linear (first-order) or singleton (zero-order) sub-models are considered. Although the TSK engine is able to generate crisp output, Mamdani engine is more intuitive and suitable for dealing with human natural language inputs using max-min operators during the inference, ie. Mamdani engine has higher interpretability.

Subsections below will introduce the two types of typical rule-based fuzzy inference systems, one is the Mamdani engine and the other is the TSK engine:

2.1.2.1 Mamdani Engine

From the introduction of fuzzy sets by Zadeh in 1965, fuzzy logic has become a significant area of interest for researchers in artificial intelligence. In particular, Mamdani was the pioneer who investigated the use of fuzzy logic for interpreting the human derived control rules, and therefore his work was considered a milestone application of this theory. The original Mamdani fuzzy inference system was proposed as the first attempt to control a steam engine and boiler combination by a set of linguistic control rules obtained from experienced human operators. A fuzzy system with two inputs x and y (antecedents) and a single output z (consequence) is described by a linguistic If-Then rule in Mamdani form :

$$\textbf{If } x = A \textit{ and } y = B, \textbf{ then } z = C, \tag{2.2}$$

where A and B are fuzzy sets in the antecedent and C is a fuzzy set in the consequence. Fig. 2.1 is an illustration of how a two-rule Mamdani fuzzy inference system derives the overall output z when subjected to two crisp inputs x and y. If using min and max for the T-norm and T-conorm operators, respectively, and using the original max-min composition, then the resulting fuzzy reasoning is shown in Fig. 2.1, where the inferred output of each rule is a fuzzy set scaled down by its firing strength via max. Other variations are possible if using different T-norm and T-conorm operators. For example, using product and max for T-norm and T-conorm operators results in the max-product composition. In general, to compute the

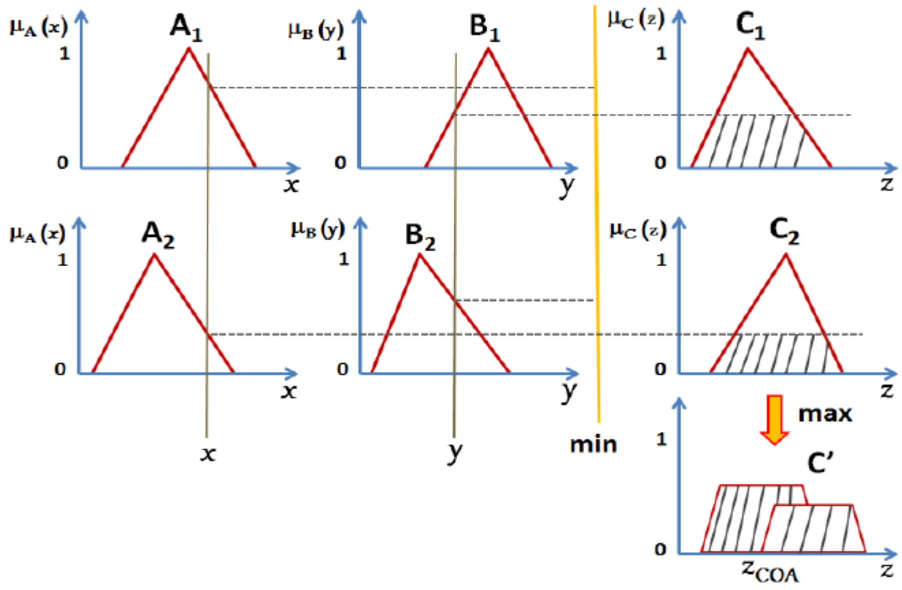

Fig. 2.1 Mamdani approach

output for the given input observation, a Mamdani inference system follows the following steps:

1. Determining a set of fuzzy rules
2. Fuzzifying the inputs using the input membership functions
3. Combining the fuzzified inputs according to the fuzzy rules to establish a rule strength
4. Finding the consequence of the rule by combining the rule strength and the output membership function
5. Combining the consequences to get an output distribution
6. Defuzzifying the output distribution (this step is involved only if a crisp output is needed)

2.1.2.2 TSK Engine

The TSK fuzzy model was proposed by Takagi, Sugeno, and Kang in 1985, and a typical fuzzy rule for the TSK model is of the following form:

$$\textbf{IF } u \textit{ is } A \textit{ and } v \textit{ is } B \textbf{ then } w = f(u,v) \,, \tag{2.3}$$

where A and B are fuzzy sets regarding antecedent variables x and y respectively, and $f(u,v)$ is a crisp function (usually polynomial), which determines the crisp value of the consequence. For instance, assume that a rule base for TSK model is comprised of two rules:

$$\begin{aligned} R_i : \textbf{IF} \quad & x\ is\ A_i\ and\ y\ is\ B_i\ \textbf{THEN}\ z = f_i(x,y) = a_i x + b_i y + c_i \\ R_j : \textbf{IF} \quad & x\ is\ A_j\ and\ y\ is\ B_j\ \textbf{THEN}\ z = f_j(x,y) = a_j x + b_j y + c_j, \end{aligned} \tag{2.4}$$

where a_i, a_j, b_i, b_j, c_i, and c_j are constants in the polynomial equation in the consequent part of the rule. The consequences of rules R_i and R_j deteriorate to constants $c_i$and c_j when $a_i = a_j = b_i = b_j = 0$. TSK model is usually employed to crisp inputs and outputs problems. Given an observation with singleton values as input (x_0, y_0), the working progress of this approach is demonstrated in Fig. 2.2. The inferred output from the given observation from rules R_i and R_j are $f_i(x_0, y_0)$ and $f_j(x_0, y_0)$, respectively. The overall output is then taken as the weighted average of outputs from all rules, where the values of weights are the firing strengths of corresponding rules. Suppose that $\mu_{A_i}(x_0)$ and $\mu_{B_i}(y_0)$ represent the matching degrees between input (x_0, y_0) and rules R_i and R_j, respectively. The firing strength (weight) of rule R_i, α_i, is calculated as:

$$\alpha_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0), \tag{2.5}$$

where $\wedge$ is a t-norm, which usually implemented as a minimum operator. Obviously, if a given input (x_1, y_1) does not overlap with any rule antecedent, the matching degree between this input and rules R_i and R_j are $\mu_{A_i}(x_1)$ and $\mu_{B_i}(y_1)$ are equal to 0, then no rule will be fired. Therefore, no consequence can be derived for such case. As the final result of the consequent variable z is a crisp value, the defuzzification progress then can be saved, which in turn reduces the overall computational efforts.

2.2 Fuzzy Rule Interpolation

A fuzzy rule interpolation method usually maintains the similarity between the antecedent fuzzy sets and that between the consequent fuzzy sets. This means the similar observations lead to similar results [86–90]. Various interpolation methods have been developed [34–37, 39], which can be categorised into two classes with several exceptions (such as type II fuzzy interpolation [91, 92]). These two classes are alpha-cuts based fuzzy rule interpolation and analogy based fuzzy rule interpolation.

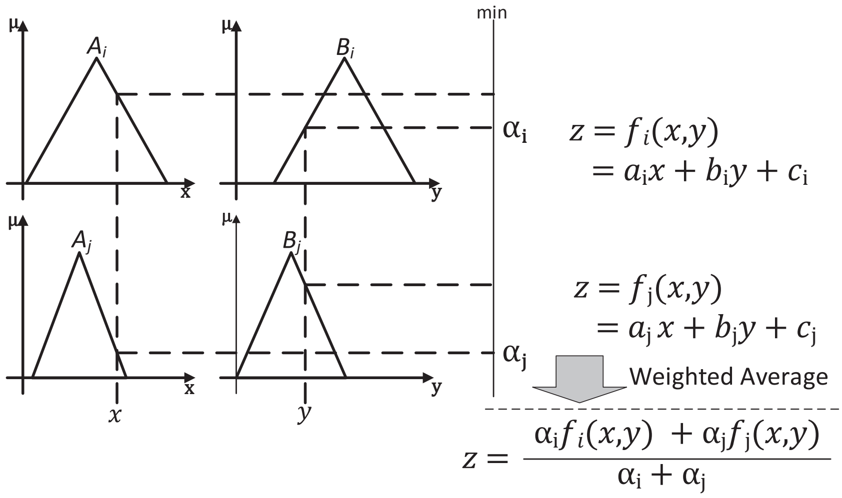

Fig. 2.2 TSK approach

2.2.1 Alpha-cuts Based Fuzzy Rule Interpolation

The first class of approaches directly interpolates rules whose antecedent variables are identical to the observed (α-cuts based). The most typical approach in this group is the very first proposed fuzzy interpolation [93–95], denoted as the Koczy-Hirota (KH) approach, which was based on the concept that the approximated conclusion divides the distance between the consequent sets of the used rules in the same proportion as the observation does the distance between the antecedents of those rules. This method is developed by using two fundamental principles. The first is the definition of the fuzzy distance using classical Shepard interpolation extension [96], and the second is the fact that fuzzy sets can be decomposed into and composed from α-cuts i.e. the Decomposition and Resolution Principles [97, 98]. This method can also be applied to multidimensional environment using the Minkowski distance.

KH approach is implemented in the following steps: 1. Determine the α-cut sets and infimum and supremum values for all the available fuzzy sets. 2. Determine the lower and upper distances through infimum and supremum values of two α-cut sets by Euclidean distance (single dimension) and Miskowski distance (multi dimension). 3. Determine the infimum and supremum values of conclusion fuzzy set by applying the linear interpolation formula. According to these principles, each fuzzy set can be represented by a series of α-cuts ($\alpha \in (0,1]$). Given a certain α, the α-cut of the consequent fuzzy set is calculated from the α-cuts of the observation and all the fuzzy sets involved in the rules used for interpolation. Knowing the α-cuts of the consequent fuzzy set for all $\alpha \in (0,1]$, the consequent fuzzy set can be assembled by applying the Resolution Principle. The closed form fuzzy interpolation was proposed [26]. It not only can be represented in a closed form but also guarantees that the interpolated results are valid fuzzy sets. Another approach is based on the areas of fuzzy sets and uses the weighted average method to infer the FRI results [99]. Approaches such as[26, 100–102] also belong to this group.

KH approach has many advantages, for example it is approximately linear between the α levels. Its computational complexity is light because it calculates the α-cut set for the conclusion and therefore, it may be suitable for real time application. It is more suitable for triangle and trapezoidal shaped fuzzy sets because these can be easily described by few characteristics points that the α-cut process. It is initially developed for Single Input Single Output (SISO) fuzzy systems, but it can be extended for the case of Multiple Input Single Output (MISO) fuzzy systems using Minkowski distances. It preserves the original rule base and supports multiple rules.

However, KH approach is restricted to convex and normal fuzzy (CNF) sets. It is shown to rely on implicative rules, viewed as constraints. It does not always provide the normal conclusion and does not maintain the piece-wise linearity of the resultant conclusion. Theoretically, an infinite number of α-cuts are required for a correct result. However, it considers only two α levels which may cause error in the result. Sometimes the bounds of the results are not in the expected order, because the interpolation weights for left-hand sides and for the right-hand sides are not related to each other. It does not always produce the convex and normal conclusion even if the given fuzzy sets are convex and normal [86, 103].

In particular, the Stabilised-KH approach extends the original KH approach, which is based on a certain interpolation of a family of distances between fuzzy sets in the rules and in the observation [104]. Unlike the original KH approach, it does not consider the two closest neighbouring rules. Instead, it takes all the rules and computes the conclusion based on the consequent parts weighted by the distances. This approach is outlined below. Suppose that a sparse rule base is composed of n rules $R_i, i \in \{1,2,...,n\}$ that are represented as follows:

$$R_i : \textbf{IF } x \text{ is } A_i \text{ and } y \text{ is } B_i \textbf{ THEN } z \text{ is } C_i. \tag{2.6}$$

Suppose each variable value A is represented as a triangular fuzzy set and conveniently denoted as (a_1,a_2,a_3), where a_2 is the core and (a_1,a_3) is the support. Given an observation

(A^*, B^*), the result C^* can be calculated by

$$
\begin{aligned}
min\, C^*_\alpha &= (\frac{\sum_{j=1}^n inf\{C_{j\alpha}\}\ \frac{1}{d_L(A^*_\alpha,\ A_{i\alpha})}}{\sum_{j=1}^n \frac{1}{d_L(A^*_\alpha,\ A_{i\alpha})}} + \\
&\quad \frac{\sum_{j=1}^n inf\{C_{j\alpha}\}\ \frac{1}{d_L(B^*_\alpha,\ B_{i\alpha})}}{\sum_{j=1}^n \frac{1}{d_L(B^*_\alpha,\ B_{i\alpha})}})/2, \\
max\, C^*_\alpha &= (\frac{\sum_{j=1}^n sup\{C_{j\alpha}\}\ \frac{1}{d_U(A^*_\alpha,\ A_{i\alpha})}}{\sum_{j=1}^n \frac{1}{d_U(A^*_\alpha,\ A_{i\alpha})}} + \\
&\quad \frac{\sum_{j=1}^n sup\{C_{j\alpha}\}\ \frac{1}{d_U(B^*_\alpha,\ B_{i\alpha})}}{\sum_{j=1}^n \frac{1}{d_U(B^*_\alpha,\ B_{i\alpha})}})/2,
\end{aligned}
\tag{2.7}
$$

where $A_{i\alpha}$ and A^*_α represent the α-cut of A_i and A^* respectively; $d_L(A_{i\alpha}, A^*_\alpha)$ and $d_U(A_{i\alpha}, A^*_\alpha)$ represent the lower and upper Euclidean distance between A^* and A_i, respectively. The values of $d_L(A_{i\alpha}, A^*_\alpha)$ and $d_U(A_{i\alpha}, A^*_\alpha)$ are illustrated in Fig. 2.3 and calculated as follows:

$$
\begin{aligned}
d_L(A_{i\alpha}, A^*_\alpha) &= d(inf\{A_{i\alpha}\}, inf\{A^*_\alpha\}), \\
d_U(A_{i\alpha}, A^*_\alpha) &= d(sup\{A_{i\alpha}\}, sup\{A^*_\alpha\}).
\end{aligned}
\tag{2.8}
$$

From this, the conclusion C^* can be generated by assembling all C_α, ($\alpha \in (0, 1]$) based on the resolution principle.

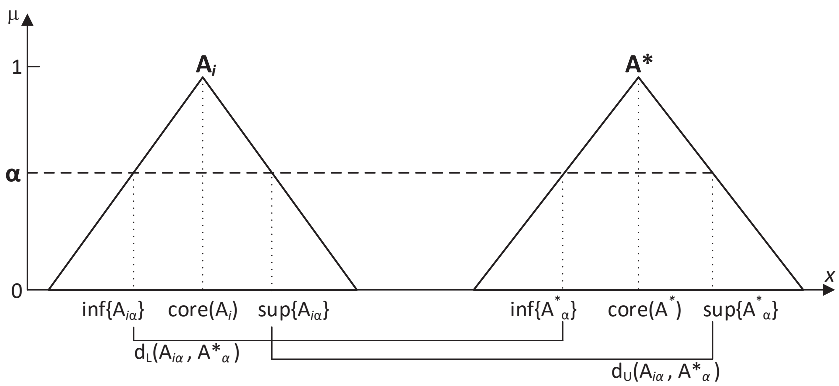

Fig. 2.3 Illustration of lower and upper distances

2.2.2 Analogy Based Fuzzy Rule Interpolation

The second class of the FRI approaches is based on the analogical reasoning mechanism [105] and therefore, referred to as “analogy based fuzzy rule interpolation" (intermediate-rule based). Instead of directly inferring conclusions, this class works by first creating an intermediate rule such that its antecedent is as “close" (given a fuzzy distance metric) to the given observation as possible (given a fuzzy distance metric or other measures based on certain similarity principles). Then, a conclusion is derived from the given observation by firing the generated intermediate rule through the analogical reasoning mechanism. That is, the shape distinguishability between the resultant fuzzy set and the consequence of the intermediate rule is analogous to the shape distinguishability between the observation and the antecedent of the generated intermediate rule. A number of ways to create an intermediate rule and then to infer a conclusion from the given observation by that rule have been developed, such as the weighted fuzzy interpolative reasoning [22], and another approach based on the areas of fuzzy sets and uses the weighted average method to infer the FRI results [99]. The most important methods in this class are the HS approach based on scale and move transformation [23] and its extensions [106, 107]. The HS approaches 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. The HS approaches not only guarantee the uniqueness, normality, and convexity of the interpolated fuzzy sets, but can also handle the interpolation of multiple antecedent variables with different types of fuzzy membership function. 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.

In particular, the scale and move transformation-based FRI with triangle fuzzy sets is introduced in detail as follows. Suppose that two fuzzy rules R_i and R_j $(i, j \in \mathbb{N})$ are given as:

R_i : **IF** X_1 is A_{1i} and X_2 is A_{2i} and ... and X_m is A_{mi}, **THEN** Y is B_i,
R_j : **IF** X_1 is A_{1j} and X_2 is A_{2j} and ... and X_m is A_{mj}, **THEN** Y is B_j,
where each fuzzy set $A_{kl}, (k = \{1,2,...,m\}, l = \{i, j\})$ has a triangular membership function and conveniently denoted as (a_{kl}, b_{kl}, c_{kl}). Given observations $(A_1^*, A_2^*, ..., A_k^*)$, the calculation process of the conclusion using FRI is summarised as the following steps.

Step 1: Calculate the *representative value* of each given antecedent variable A_{kl} using Equation 2.9, and do the same for the given observation.

$$Rep(A_{kl}) = \frac{a_{kl} + b_{kl} + c_{kl}}{3} \tag{2.9}$$

Step 2: Calculate the *relative placement factor* λ_k, by Equation 2.10 based on the relative location of the observation regarding to the two antecedents, and then calculate the average $\lambda_{average}$ for later use.

$$\lambda_k = \frac{d(Rep(A_{ki}), Rep(A_k^*))}{d(Rep(A_{kj}), Rep(A_k^*))}, \ (k = 1, 2, ..., m) \tag{2.10}$$

Step 3: Based on λ_k calculated above, obtain the antecedents of the new intermediate rule $A_k^{'}$ by Equation 2.11.

$$A_k^{'} = (1 - \lambda_k)A_{ki} + \lambda_k A_{kj} \tag{2.11}$$

Step 4: By comparing the size of $A_k^{'}$ and A_k^*, obtain the *Scale Rate* S_i using Equation 2.12.

$$S_k = \frac{c_k^{'} - a_k^{'}}{c_k^* - a_k^*}, \tag{2.12}$$

Step 5: Obtain the average of *Scale Rate* $S_{average}$ based on the $S_1 ... S_m$ calculated above.
Step 6: Apply scale rate S_k to $A_k^{'}$ to obtain $A_k^{''}$ by Equation 2.13, 2.14, 2.15 ($k = 1, 2, ..., m$).

$$a_k^{''} = \frac{a_k^{'}(1 + 2S_k) + b_k^{'}(1 - S_k) + c_k^{'}(1 - S_k)}{3} \tag{2.13}$$

$$b_k^{''} = \frac{a_k^{'}(1 - S_k) + b_k^{'}(1 + 2S_k) + c_k^{'}(1 - S_k)}{3} \tag{2.14}$$

$$c_k^{''} = \frac{a_k^{'}(1 - S_k) + b_k^{'}(1 - S_k) + c_k^{'}(1 + 2S_k)}{3} \tag{2.15}$$

Step 7: By comparing the shapes of A_k^* and $A_k^{''}$, obtain *Move Transformation Rate* M_k, then calculate the average of move transformation rate $M_{average}$ for later use.

$$M_k = \begin{cases} \frac{3(a_k^{''} - a_k^*)}{b_k^* - a_k^*}, & if\ a_k^{''} \geq a_k^* \\ \frac{3(a_k^{''} - a_k^*)}{c_k^* - b_k^*}, & if\ c_k^{''} \leq a_k^* \end{cases} \tag{2.16}$$

Step 8: To compute $B^{'}$ from the rule consequences using Equation 2.11 based on the $\lambda_{average}$ calculated above.
Step 9: Finally, the fuzzy set B^* of the conclusion can then be estimated by apply $S_{average}$ and $M_{average}$ on B'.

The 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 fuzzy membership functions, such as 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.

Note that 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 [108–110]: the more similar between the observation and an antecedent, the more similar conclusion must be concluded to the corresponding consequent set. As traditional generated rule bases are almost all based on certain similarity measures, therefore they are likely to cause performance deterioration along with the size reduction of the rule base.

2.2.3 Adaptive Fuzzy Rule Interpolation

Common to all above fuzzy interpolation techniques is the fact that interpolations are carried out in a linear manner. However, when dealing with realistic problems this is not always feasible and may lead to inconsistencies in inferred rules and reasoning results after a certain amount of interpolations. Therefore, adaptive fuzzy rule interpolation was proposed to adaptively address this [24, 25, 111–113], and successfully applied to a number of real-world problems [114]. This system has to main components: 1) a diagnostic system that is implemented by the use of the classical candidate generation procedure of the general diagnostic engine (GDE) [115] through the exploitation of inconsistent interpolative (intermediate) results that are recorded in an assumption-based truth-maintenance system (ATMS) [116], and 2) a corrective system that is developed from the fuzzy extension of the conventional numerical piece-wise linear interpolation theory [117] and its application in approximate computation [118]. The method works, first, by the extraction of the entire set of interpolated rules, which depend on the same pair of neighbouring rules in the generated candidate list. Then, it imposes a group of equations and inequations, which not only constrain the modified propositions and ensure their propagation to be consistent but guarantee the original similarity-based reasoning in fuzzy interpolation to be followed as well. Finally, the approach corrects the culprit interpolated rules by solving the set of simultaneous equations and inequations. Another method was proposed based on similarity measures of polygonal fuzzy sets and novel move and transformation techniques for solving contradictory fuzzy interpolative reasoning results [119].

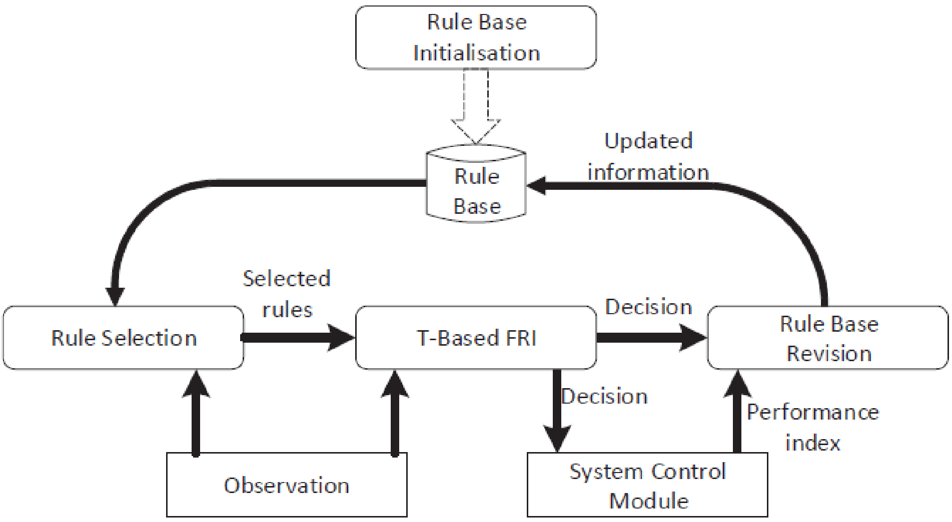

Fig. 2.4 Adaptive Fuzzy Rule Interpolation [2]

A novel rule base generation and adaptation system for FRI was also introduced in [2], and the system framework is outlined in Fig. 2.4, which comprises of mainly four parts: rule base initialisation, rule selection for interpolation, transaction-based FRI, and rule base revision. Firstly, an initial set of rules is generated from limited a priori knowledge. For a given observation, the system then selects the 'best' two rules from the current rule base for interpolation, based on a particular set of metrics, including the usage frequency/experience information, the previous performance index and the distances between the given observation and the rule antecedents. From this, a new rule is interpolated based on the selected 'best' rules from the given observation. Afterwards, the performance index will be utilised to support the rule base updating, whenever it is available from the feedback system. Rule Selection was presented a novel rule base generation and adaptation approach for FRI, which is able to adaptively generate and revise the rule base with limited training data and/or expert knowledge for control problems, as long as a performance index is available to indicate if the inference result is acceptable or not. Some work focuses on automated rule base construction in an effort to enhance the potential of the Takagi-Sugeno fuzzy regression models for situations where only limited training data are available [120].

2.2.4 Dynamic Fuzzy Rule Interpolation

The HS approaches have been also extended as dynamic fuzzy rule interpolation [3, 30, 121–123]. A initial dynamic fuzzy rule interpolation approach [3] has been firstly introduced to better exploit the interpolation results provided by a given FRI method. It has presented an initial attempt towards building an intelligent framework for dynamic FRI: the interpolated rules are analysed, aggregated, and promoted into the original sparse rule base. Fig. 2.5 illustrates the overall operation for this framework. Initially, there is a set of original (sparse) rules $\mathbb{R}$. While running the FRI system, an interpolation method such as HS approach continuously fills a pool of interpolated rules $\mathbb{R}'$. The antecedent domains of $\mathbb{R}'$ are then partitioned into a set of hypercubes $\mathbb{H}$, so that certain regions that have accumulated a good number of interpolated rules can be determined. A clustering algorithm is then employed to group the similar rules together, for each 'filled' hypercubes, whilst the most informative clusters are also identified. Finally, an aggregation process is applied to the selected groups of rules, in order to construct and promote new rules to becoming member of $\mathbb{R}$.

However, the above approach relies heavily on the use of the standard k-means clustering algorithm. Yet, for many application problems, it is difficult to predict the value of k (the number of clusters) [124]. Therefore the initial approach was improved by employing a genetic algorithm (GA) based clustering technique [125, 126] in place of k-means clustering. In these works, the collection of the interpolated rules is pre-partitioned into hypercubes (multidimensional blocks), in order to reduce the search complexity of the GA process. The non-empty hypercubes are then identified and used as the input into GA. After a certain number of generations, GA identifies a 'best' chromosome (cluster arrangement) based on a given fitness function such as the Dunn Index [127]. Here, a chromosome is viewed as a combination of strong and weak clusters, where the weak clusters are merged into the closest strong clusters in order to obtain the final result. In the end, the densest clusters that have accumulated a sufficient number of candidate rules are selected for rule aggregation and promotion.

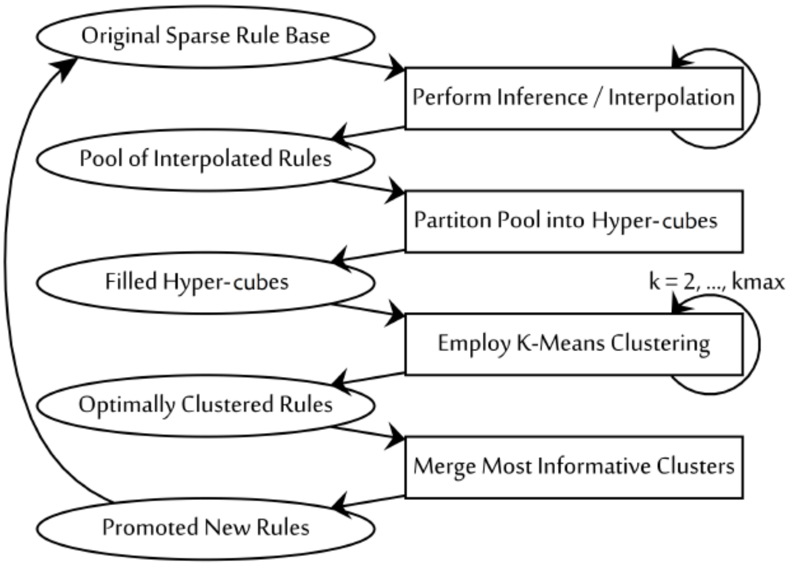

Fig. 2.5 Dynamic Fuzzy Rule Interpolation [3]

2.3 Rule Base Generation and Reduction

2.3.1 Rule Base Generation

Most fuzzy rule base generation methods are based on grid-type fuzzy partition [8, 11, 42, 58, 82, 128–133].These methods firstly divide the input and output spaces into fuzzy regions: given a set of examples with multiple inputs (m) and single output, denoted as $(x_j^k; y^k)$ where j =1, . . ., m and k =1, . . ., n. Define the universe of discourse of each input variable as $[x_j^-; x_j^+]$ and the output variable as $[\ y^-; y^+]$ and then divide each universe of discourse into N regions. The minimal and maximal values of each variable are often used to define its universe of discourse. That is $[x_j^-; x_j^+]$=$[min(x_j), max(x_j)]$. They are also considered to be the centre of the left end term and the right end term, respectively. That is, $c_{1j} = min(x_j)$ and $c_{nj} = max(x_j)$. Accordingly, the other linguistic term centre of each region, c_{ij}, can be computed as follows:

$$C_{ij} = min(x_j) + \ i(max(x_j)) - \ min(x_j))/(N-1), \ where\ i = 2,...,N-1. \tag{2.17}$$

Then these methods secondly generate fuzzy rules from given examples, by determining its membership degrees in every fuzzy partition of the input and output spaces, and associating each example a rule with the linguistic labels best covered by the example. Based on a collection of n-inputs-dimensions data examples, a multi-input and single-output (MISO) fuzzy model is represented as a collection of fuzzy rules in the following form:

$$R_i : \textbf{IF } x_1 = A_{i1} \text{ and } x_2 = A_{i2} \text{ and } x_n = A_{in}, \ \textbf{THEN } y_i = z_i(x), \tag{2.18}$$

where R_i is the ith rule, A_{ij} are linguistic labels represented by fuzzy sets of the universes of discourse in the $[x_j^-;x_j^+]$, with its linguistic term centre as C_{ij}. According to the different form of the consequences in the if-then rules, there are two types of fuzzy models, Madamni or TSK. The $z_i(x)$ takes the following forms: linguistic labels represented by fuzzy sets of the universes of discourse in the $[y_j^-;y_j^+]$, $z_i(x) = B_i$, which is Mamdani model; or linear function, $z_i(x) = b_{i0} + \sum_{j=1}^{s} b_{ij}x_j$, which is TSK model.

Finally the rule base can be optimised or adjusted to avoid conflicting rules. The most common way to optimise the membership functions is to change their definition parameters. For example, triangular shaped membership functions are usually considered due to their simplicity:

$$\mu(x) = \begin{cases} \frac{x-l}{c-l}, \ if \ l \le x < c \\ \frac{r-x}{r-c}, \ if \ c \le x \le r \\ 0, \ otherwise \ , \end{cases} \tag{2.19}$$

changing the basic parameters of left, centre, right values will vary the shape of the fuzzy set (l,c,r) associated to the membership function, thus influencing the fuzzy rule base performance. The same yields for other shapes of membership functions (trapezoidal,gaussian, sigmoid, etc.).

Most fuzzy rule base generation methods are based on grid-type fuzzy partition. They divide a given problem space into a number of fuzzy regions, each representing a fuzzy rule that is used to construct the final rule base. From this, the raw rule base is optimised by a general optimisation approach, such as the genetic algorithm. As an important benchmark, the Wang and Mendel's (WM) method in [41] provides a fast and non-iterative way to learn the linguistic rules from data and has been proven with many successful applications [134, 135]. It performs rule induction by considering that the fuzzy partitions and the membership functions are already set, with domains even partitioned (uniformly partitioned). The detailed procedures of the benchmark WM model are described as follows:

Step 1: Divide the input and output spaces into fuzzy regions. Given a set of examples with multiple inputs (m) and single output, denoted as $(x_j^k;y^k)$ where j =1, . . ., m and k =1, . . ., n. Define the universe of discourse of each input variable as $[x_j^-;x_j^+]$ and the output variable as $[\ y^-;y^+]$ and then divide each universe of discourse into N regions. The minimal and maximal values of each variable are often used to define its universe of discourse. That is $[x_j^-;x_j^+]$=$[min(x_j), max(x_j)]$. They are also considered to be the centre of the left end term and the right end term, respectively. That is, $c_{1j} = min(x_j)$ and $c_{Nj} = max(x_j)$. Accordingly, the other term centre, c_{ij}, can be computed as follows:

$$C_{ij} = min(x_j) + \ i(max(x_j)) - \ min(x_j))/(N-1), \ where \ i = 2,...,N-1. \tag{2.20}$$

Step 2: Generate fuzzy rules from given examples. Firstly, determine the membership degrees of each example belonging to each fuzzy term defined for each region, variable by variable (including the output variable). Secondly, associate each example with the term having the highest membership degree variable by variable, denoted as md_j. Finally, obtain one rule for each example using the term selected in the previous step. The rules generated are “and" rules and the antecedents of the IF part of each rule must be met simultaneously for the consequence of the rule to occur. Letting Tx_j be a term selected for variable x_j of an example, a rule could look like:

$$IF \ x_1 \ is \ Tx_1 \ (with \ md_1) \ and \ x_2 \ is \ Tx_2 \ (with \ md_2) \ and... \ and \ x_m \ is \ Tx_m \ (with \ md_m) \\ THEN \ y \ is \ Ty \ (with \ md_y). \tag{2.21}$$

Step 3: Assign a degree to each rule. The rule degree is computed as the product of the membership degree of all variables (m inputs $x_1 to x_m$ and 1 output y). Let D^k be the degree of the rule generated by example k, mathematically,

$$D^k = \prod_{j=1,...,m \ and \ y} md_j^k. \tag{2.22}$$

Step 4: Create a combined fuzzy rule base. When the number of examples is high, it is quite possible that the same rule could be generated for more than one example. These rules are redundant rules. In addition, rules with the same **if** part but a different **then** part could also be generated. These rules are conflicting rules. The redundant and conflicting rules must be removed to maintain the integrity of the rule base. This is achieved by keeping only the rule with the highest degree for each fuzzy region: this rule is deemed most useful. Up to this step, the fuzzy rule base is complete; however, the usefulness of the rule base must be shown using some fuzzy inference methods, as introduced in the next step.

Step 5. Determine a mapping based on the combined fuzzy rule base. To predict the output of an unseen example denoted as x_j, the centroid defuzzification formula is used. Accordingly, the predicted output, y, is computed as

$$y = \frac{\sum_{r=1}^{R} amd^r c^r}{\sum_{r=1}^{R} amd^r}, \tag{2.23}$$

where $amd^r = \prod_{j=1,...,m} md_j^r, c^r$ is the centre value of the consequent term of rule r, and R denotes the total number of combined rules.

Another successful method based on grid-type fuzzy partition is the 'cooperative rules' (COR) strategy, developed by Cordón and Herrera and refined in [130]. The COR method extends non-iterative methods such as WM, by creating a large pool of possible rule-bases using search heuristics. It can be divided into two steps: the generation of the pool of all possible rule bases, and the heuristic selection of the final rule base. The pool is generated by allowing two or more consequences for the same pair of antecedents. Then the heuristic selection uses each chromosome to represent a possible rule base from the rule base pool, which is evaluated on the set of training examples. Some recent studies proposed the selection-reduction (SR) method [136], which uses an iterative procedure to learn the system rule base. Here, unlike the COR approaches, iterations are performed in a greedy way, yielding a low-complexity and time-deterministic method. In [131, 137], to improve the system accuracy, fuzzy rules consider linguistic terms that are defined in flexible fuzzy partitions with different granularity levels, being changed from even partitions into uneven partitions.

In the non-grid fuzzy partition type, fuzzy clustering is another well recognised paradigm to generate the initial fuzzy model [138]. Each cluster represents a certain operation region of the system, and the number of clusters equals the number of rules. One of the most widely used approaches in fuzzy modelling is based on the fuzzy C-means (FCM) algorithm such as [139]. The FCM algorithm partitions a collection of n data points $(X = x_1, x_2, ..., x_n)$ into c fuzzy clusters, where the number c of clusters must be predetermined. Methods like cluster validity measures or compatible cluster merging can be applied to find a suitable number of clusters. Other algorithms such as evolutionary algorithms have also been used for building compact fuzzy rules [140], both in grid fuzzy partition or non-grid partition types.

A number of ways are available in the implementation of the aggregation operator, including max and min operators [69, 141], arithmetic averaging (AA) [142–145], weighted arithmetic averaging (WAA) [142], ordered weighted averaging (OAA) [143], generalised ordered weighted average (GOWA) [146], fuzzy weighted averaging (FWA) [147], geometric averaging [148], weighted geometric averaging (WGA) [149, 150], induced ordered weighted averaging (IOWA) [144], weighted ordered weighted averaging (WOWA) [145], and weighted order statistic averaging (WOSA) [151]. These operators have been used in a wide range of applications, such as database systems [152, 153], fuzzy logic controllers [154, 155], and decision making [143, 156, 157]. The most commonly used approach in these aggregation approaches is WAA, which is also applied in this work. work. This approach first weights all the given arguments, and then aggregates all these weighted arguments into a collective one.

More focus should be invested in how to keep good balance between model accuracy, efficiency and transparency, that means, how to generate a sparse rule base which is very accurate, computationally efficient and linguistically tractable.

2.3.2 Rule Base Simplification

Almost all the existing rule base generation approaches may suffer from the redundancy problem and the 'curse of dimensionality'. These issues can be addressed by reducing the constructed fuzzy rules through feature selection and instance selection. Empirical studies shows that some variables or features are not sufficiently important to be included in the realisation of the fuzzy model during the fuzzy rule base generation process, as some features may be redundant or barely relevant. Thus, to implement feature selection before constructing the fuzzy models may reduce the fuzzy rule search space and increases the accuracy of the model, as introduced in many references such as [158–161]. In general, the feature selection algorithms can be classified into three categories: filters, wrappers, and embedded methods [162]. The filter method selection criterion is independent of the learning algorithm. In contrast to the wrapper method, the selection criterion is dependent on the learning algorithm and uses its performance index as the evaluation criterion. The embedded method incorporates feature selection as part of the training process. Many feature selection methods combined with constructing fuzzy models were developed as the orthogonal transformation approach [163–166], the computational geometric approach [167], the evolving of Takagi–Sugeno fuzzy model [168], pareto multi-objective cooperative co-evolutionary algorithm [169], clonal selection algorithm [38], and more methods are reviewed and compared in medical data mining [170], and in synthetic data [171].

Instance selection is to choose the relevant data in training data set which are important to reflect the knowledge for solving the concerned problems. Commonly, several instances are stored in the training set but some of them are not useful for classifying as there may be noise and redundancy. Therefore, it is possible to get acceptable classification rates ignoring non-useful cases. The WM method was also completed in [172] (WMC) to reduce the influence of noisy data in original WM method. Like in feature selection, according to the strategy used for selecting instances, the instance selection algorithms can be mainly classified into two categories: wrappers and filters [173]. Most of the methods are wrappers. They are mainly based on the k-NN rule such as [174] [175], and several methods are based on coevolutionary fuzzy system approach [176], memetic algorithms [177], or class conditional instance selection based on variance [178]. In contrast, the filters methods are not based on a classifier to distinguish the instances to be selected or discarded from the training set, for example, they may be based on clustering for instance selection, or prototype selection by relevance. Filter methods may have better performance with respect to wrappers, and as an additional characteristic they spend less runtime than the wrappers [179].

Some instance selection methods such as tabu search (TS), clonal selection algorithm (CSA), prototype selection by relevance (PSR) are reviewed in [173], and many methods particularly in partial least squares regression (PLSR) are also reviewed in [180]. Instance selection needs to choose the important data from training data set, but before the proposed work it lacks an easy way to efficiently and reasonably distinguish the relevant data is important or not.

Instead of to perform feature selection and instances selection separately, hybrid method focuses on the integration of feature selection and instances selection in the construction of the fuzzy models. Both processes are applied simultaneously to the initial data set, in order to obtain a suitable subset of feature and data to construct the parameters for the fuzzy model. Some hybrid methods have been proposed, such as scalable simultaneous evolution [181], clustering and evolving fuzzy rules [182], cooperative coevolution with the nearest neighbour rule [183], [184], rule similarity checking and gradient learning [185], fuzzy complex numbers [186], or particle swarm optimization [161]. Several sparse fuzzy system generation methods have been proposed [8, 187–192]. One of their important features is that in most of the cases the resulting rule base is sparse, which ensures a reduced system complexity. Clustering are often used in these methods. However, some methods just simply increase the number of the fuzzy rules by inserting new rules in the required positions [190, 191].

Existing rule base reduction techniques consist of mainly 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.

Most of existing rule base generation and reduction approaches are based on certain similarity measures [58, 193–195]; therefore, they are likely to cause performance deterioration along with the size reduction of the rule base. They usually reduce redundant information that is found in the form of similar fuzzy sets in the rule base [195]. It simplifies the rule base by merging fuzzy sets having a similarity higher than a given threshold. Merging creates a common fuzzy set that replaces the occurrence of the merged ones in the rule base, which reduces the number of fuzzy sets that are used in the model, and thus simplifies the rule base. The compatible cluster merging algorithm for rule base generation [193] is based on the work of Krishnapuram and Freg [196]. Then Chao, Chen and Teng [194] utilised fuzzy similarity based merging. They first derive simple triangular approximate equations from Gaussianshaped fuzzy sets. Then the measurement between two triangular fuzzy sets is proposed, and similar linguistic terms are merged into one. This method indirectly results in decreasing of the number of rules. The work reported in [195] 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 [194]. 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 [195]: one is set-theoretic-based and the other is geometric-based. The former set-theoretic-based similarity measure is sensitive among different pairs of joint fuzzy sets, while the latter representative value geometric-based similarity measure is sensitive among different pairs of disjoint fuzzy sets.

Set-theoretic-based measurements are the most suitable for capturing similarity among overlapping fuzzy sets. Based on the set-theoretic operations of intersection and union, the similarity between fuzzy sets is defined as 2.24. The work of [194, 195] implement set-theoretic-based similarity measurements.

$$S(A,A') = \frac{A \cap A'}{A \cup A'} \,. \tag{2.24}$$

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. As an example of geometric-based similarity merging methods, Jin makes use of the distance concept rather than the set operations [197]. Based on different measurement standards, various similarity measures have been proposed in literature to calculate the degree of similarity between two fuzzy sets, such as [198–200]. Note that, in order to generate reasonable measurement of similarity, the corresponding variable domain is required to be normalised first. This work also focuses on the geometric-based similarity.

Given two triangle fuzzy sets on the variable with normalised domain, $A = (a_1, a_2, a_3)$ and $A' = (a'_1, a'_2, a'_3)$, where $0 \leq a_1 \leq a_2 \leq a_3 \leq 1$, and $0 \leq a'_1 \leq a'_2 \leq a'_3 \leq 1$, the degree of similarity $S(A,A')$ between fuzzy sets A and A' can be calculated as follows [198]:

$$S(A,A') = 1 - \frac{\sum_{i=1}^{3} |a_i - a'_i|}{3} . \tag{2.25}$$

The larger value of $S(A,A')$ means that there is more similarity between fuzzy sets A and A'. This method is also the most wildly applied method. It requires a normalisation progress for the concerned variable domain.

A graded mean integration representation distance-based similarity degree measurement does not need such normalisation. This geometric-based similarity measure is summarised as [201]:

$$S(A,A') = \frac{1}{1+d(A,A')} , \tag{2.26}$$

where $d(A,A') = |P(A) - P(A')|$, $P(A)$ and $P(A')$ are the graded mean integration representation of A and A', respectively [201]. In particular, the graded mean integration representation $P(A)$ and $P(A')$ can be defined as:

$$\begin{aligned} P(A) &= \frac{a_1 + 4a_2 + a_3}{6} , \\ P(A') &= \frac{a'_1 + 4a'_2 + a'_3}{6} . \end{aligned} \tag{2.27}$$

Usually the larger value of $S(A,A')$ means higher degree of similarity between fuzzy sets A and A'.

The above two approaches may not provide correct similarity degrees in certain situations, (such as two generalised fuzzy sets, which may not be normal), although they are usually able to produce acceptable results and have been widely applied. A generalised triangle fuzzy set regarding variable x can be represented as $A = (a_1, a_2, a_3, \mu(a_2))$, where $\mu(a_2)$ ($0 < \mu(a_2) \leq 1$) is the membership of element a_2, and $\mu(a_2) \geq \mu(a), \forall a \in D_x$, D_x is the domain of variable x, as illustrated in Fig 2.6. If $\mu(a_2) = 1$, the generalised triangle fuzzy set deteriorates to normal a fuzzy set which is usually denoted as $A = (a_1, a_2, a_3)$. A centre of gravity method (COG) has been proposed to work with generalised fuzzy sets [200]. The concept COG is that of an average of the masses factored by their distances from a reference point. The COG is an important property since it reflects both the location and the shape of a fuzzy set [202]. The COG metric is used to determine the closeness of two sets based on their COG reference points. The process to calculate the COG-based similarity degree measure is summarised as below.

Step 1: Determine the point of centre of gravity for each triangle fuzzy set. Given a generalised triangle fuzzy set A, its COG $G(a^*, \mu(a^*))$ is shown in Fig. 2.6, which can be calculated by:

$$a^* = \frac{a_1 + a_2 + a_3}{3} , \tag{2.28}$$

$$\mu(a^*) = \frac{\mu(a_1) + \mu(a_2) + \mu(a_3)}{3} . \tag{2.29}$$

As $\mu(a_1) = \mu(a_3) = 0$, $\mu(a^*)$ can be simplified to:

$$\mu(a^*) = \frac{\mu(a_2)}{3} . \tag{2.30}$$

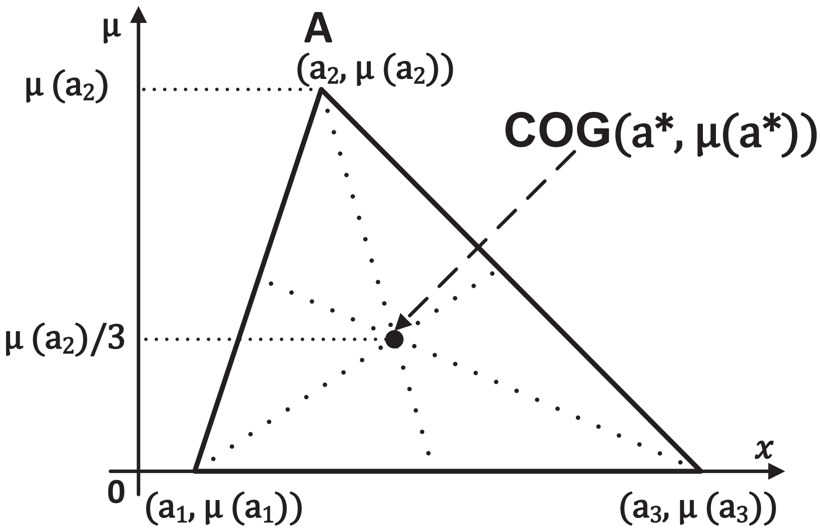

Fig. 2.6 An example triangular fuzzy set and its COG [4]

Step 2: Calculate the similarity degree $S(A, A')$ between fuzzy sets A and A' by:

$$S(A,A') = \left(1 - \frac{\sum_{i=1}^{3} |a_i - a'_i|}{3}\right) \cdot \left(1 - |a^*_A - a^*_{A'}|\right)^{B(Supp_A, Supp_{A'})} \cdot \frac{min(\mu(a^*_A), \mu(a^*_{A'})}{max(\mu(a^*_A), \mu(a^*_{A'})} , \tag{2.31}$$

where a_A^* and $a_{A'}^*$ are calculated by Equation 2.28, $\mu(a_A^*)$ and $\mu(a_{A'}^*)$ are obtained from Equation 2.30, and $B(Supp_A, Supp_{A'})$ is defined as follow:

$$B(Supp_A, Supp_{A'}) = \begin{cases} 1, \; if \; Supp_A + Supp_{A'} \neq 0 \\ \\ 0, \; if \; Supp_A + Supp_{A'} = 0 \; , \end{cases} \tag{2.32}$$

where $Supp_A$ and $Supp_{A'}$ are the supports of the fuzzy sets A and A' respectively, which in turn are calculated as:

$$\begin{aligned} Supp_A &= a_3 - a_1 \; , \\ Supp_{A'} &= a'_3 - a'_1 \; . \end{aligned} \tag{2.33}$$

In the above equation, $B(Supp_A, Supp_{A'})$ is used to determine whether COG distance $(1 - |a_A^* - a_{A'}^*|)$ needs to be considered. For instance, if fuzzy sets A and A' both are the crisp values, (i.e., $Supp_A = Supp_{A'} = Supp_A + Supp_{A'} = 0$), the COG distance will not be considered for the degree of similarity measure; otherwise, the COG distance will be considered. In this measure, again the larger value of $S(A, A')$ means that the two fuzzy sets A and A' are more similar.

2.3.3 Rule Base Adaptation

There is a need for effective methods for tuning the membership functions so as to minimise the output error measure or maximise performance index. Considering the data-driven methods for rule base generation, usually a raw rule base is generalised from the data firstly, by which the fuzzy partition of variable domains and the number of rules are determined. For instance, neuron-fuzzy system has been employed to initialise membership functions and to extract fuzzy rules from a large data set [203]. Secondly, the raw rule base is adapted or fine-tuned using optimisation algorithms, with Genetic Algorithm (GA) being a popular choice [204] and other hybrid choice [205, 206]. Knowing that the success of data-driven approaches is built on a large amount of training data and complex calculations, the system may perform poorly if the quantity of the available data is not able to reach the minimum requirement for the appropriate use of neural networks or GAs.

Tuning fuzzy rule-based systems for linguistic fuzzy modeling is an interesting and widely developed task. It involves adjusting some of the components of the knowledge base without completely redefining it. One of the most promising research topics in fuzzy modelling relates with the quest of a good trade-off between interpretability and accuracy. It is usually implemented by tuning the surface structure using linguistic hedges or by tuning the deep structure by changing the basic membership functions parameters and using nonlinear scaling factors [207]. Genetic tuning of fuzzy rule is often used for such rule base adaption [208, 209, 207].

The adaption for TSK model is relatively simple because the adaption has no need to consider the linguistic meaning for adjusted rules. In [210], a recursive approach for adaptation of fuzzy rule-based model structure has been developed and tested. It uses on-line clustering of the input–output data with a recursively calculated spatial proximity measure. The proposed algorithm is instrumental for on-line identification of Takagi–Sugeno models, exploiting their dual nature and combined with the recursive modified weighted least squares estimation of the consequent-part parameters of the model. The procedure for the on-line rule-base adaptation can be summarised as follows:

1. Initialisation of the rule-base structure (antecedent part of the rules).

2. Reading the next data sample at the next time step.

3. Recursive calculation of the potential of each new data sample.

4. Recursive update of the potentials of the established centres because of the influence of the new data.

5. Possible modification or innovation of the rule-base structure based on the comparison between the potential of the new data sample and the potential of the existing rules' centres (continues for the next time step from step 2).

The adaptive-network-based fuzzy inference system (ANFIS) [211] serves as a basis for constructing a set of fuzzy If-Then rules with appropriate membership functions to generate the stipulated input-output pairs. It can construct an input-output mapping based on both human knowledge in the form of fuzzy If-Then rules and stipulated input-output data pairs.

2.4 Curvature Value Calculation

Curvature is an important concept in the field of geography, which is conventionally used to investigate the water flow over a landscape. It is used in 2D images or maps, and 3D digital elevation models. It has been used to match finger print, describe shapes of Devanagari characters [212], and localise the nose tip point of 3D-face images [213]. It is used in graphical and engineering applications that contain more irregular discrete data, such as feature recognition, segmentation, rendering, and face recognition [214–217].

There are mainly two types of methods for curvature calculation. The first type is based on the directional derivative with meshes or curve fitting [218], such as the profile curvature, the streamline curvature, and the planform curvature. The second type is based on relation of points and surface, usually works on a moving least-squares (MLS) surface [219], such as the Gaussian curvature, the mean curvature, the maximum principal curvature and the minimum principal curvature. The maximum and minimum principal curvatures can be derived from the Gaussian curvature and the mean curvature. The first type is relatively simple and is usually used in meshes or curve fitting situations with more regular data, in comparison, the second type is more suitable for irregular data with complex model.

2.4.1 Directional Derivative

The first type of curvature calculation methods is based on a directional derivative [218]. A directional derivative represents the steepest downward gradient for a given direction. It refers to the rate at which any given scalar field F(x,y), changes as it moves in the direction of some unit vector, $\widehat{n}$:

$$D_(\widehat{n})(\mathrm{F}) = \bigtriangledown \mathrm{F} \cdot \widehat{n}, \tag{2.34}$$

where F(x,y) is a scalar field, $\widehat{n}$ is a unit vector. This expression can be used to define several different kinds of curvature, among which the profile curvature is a typical one. It is the rate at which the surface slope, S, changes as it moves in the direction of the unit vector $-(\bigtriangledown f/\mathrm{S})$, i.e., in Eq. 2.34, the scalar field F is as S, the unit vector $\widehat{n}$ is as $-(\bigtriangledown f/\mathrm{S})$, and $\bigtriangledown f$ is the gradient of this surface. Note S is the slope defined as the magnitude of the gradient vector:

$$S(x,y) = |\bigtriangledown| = \sqrt{f_x^2 + f_y^2}, \tag{2.35}$$

where the subscripts x and y indicate the partial derivatives of the surface $f(x,y)$.

As illustrated in Fig. 2.7, eight directions are defined to represent the directions clockwise from the centre of the sub-region to the four corners and the central points of the four edges. Based on these directions, different directional derivative values ($D_i, i = 1,2,...,8$) can be calculated, with the profile curvature going along the steepest downward gradient. That is, the profile curvature value takes the maximum absolute value of the eight directional curvature values: $V_p = \max(D_1,D_2,D_3,D_4,D_5,D_6,D_7,D_8)$. Note that during the calculation, the unit vector $\widehat{n}$ in the red directions is $\sqrt{2}$ times of the one in the green directions. After the calculation, a zero curvature value indicates that the surface is flat or linear, while the positive or negative value indicates that the surface is upwardly concave or convex. As the current work only interests in the linearity of the surface, the biggest absolute curvature value D_i of the eight directional curvature values is utilised. The profile curvature can also be directly calculated using the partial derivatives with following formula derived from Eq. 2.34.

$$V_p = -\frac{(f_x^2 f_{xx} + 2f_x f_y f_{xy} + f_y^2 f_{yy})}{S^2}. \tag{2.36}$$

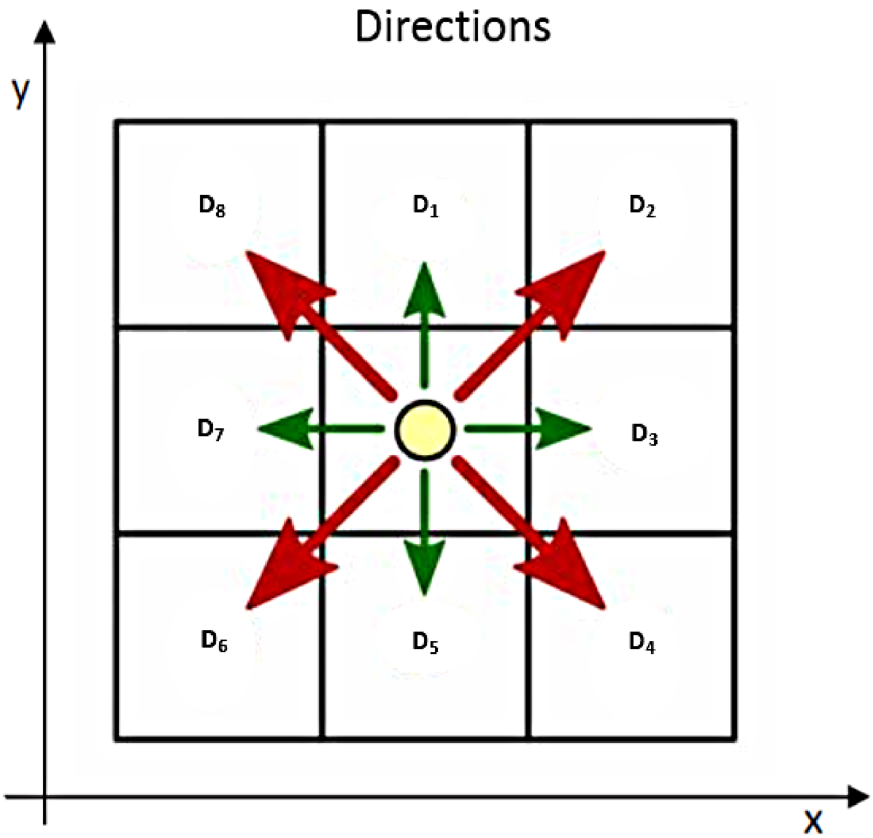

Fig. 2.7 Directions for profile curvature value approximation

2.4.2 MLS surface

The second type of curvature calculation methods is based on a MLS surface. It defines a MLS surface S as the stationary set of a projection operator ψ_p , i.e., $S = \{(x \in R^3 | \, \psi_p(x) = x\}$. As a smooth surface defined by a projection process [220, 221], MLS surfaces have been proposed and proven not only to be successful in 3D modeling and rendering, but also been applicable for point set de-noising, up-sampling, down-sampling, offsetting and so on.

Based on a more general definition of this projection [222, 223], a mathematical proof of the convergence of the projection procedure is presented in [224, 225].

Except for the previously surfaces methods such as curve fitting [226, 227], polygonal meshes [228, 229], high-order surface based methods by locally fitting a surface or by least square fitting of a local parametric quadric surface [230, 231], an new implicit MLS surface method was proposed in [219]. It explicitly defines the MLS surface as the local minimal of an energy function e(y,a) along the directions given by a vector field n(x). Here, y is a position vector and a is a direction vector. Following this, the MLS surface S can be implied or implemented by determining the vector field n and the energy function e. Suppose a normal vector v_i is assigned to each point of $\mathrm{q}_i \in R^3$ of an input-point set Q, then the vector field n is:

$$\mathrm{n}(\mathrm{x}) = \frac{\sum_{\mathrm{q_i} \in \mathrm{Q}} \mathrm{v_i} \theta(\mathrm{x}, \mathrm{q_i})}{||\sum_{\mathrm{q_i} \in \mathrm{Q}} \mathrm{v_i} \theta(\mathrm{x}, \mathrm{q_i})||}, \tag{2.37}$$

where

$$\theta(\mathrm{x}, \mathrm{q_i}) = e^{\frac{-||\mathrm{x}-\mathrm{q_i}||^2}{h^2}} \tag{2.38}$$

is a Gaussian weighting function, in which h is a Gaussian scale parameter that determines the width of the Gaussian kernel. If the normal vector v_i is not readily available, as can be the case in some applications, it can easily compute a normal vector for any point with the normalised weighted average of the normals of its nearby sample points. The energy function $e : R^3 \times R^3 \to R$ can be defined as

$$e(\mathrm{y}, \mathrm{n}(\mathrm{x_j})) = \sum_{\mathrm{q_i} \in \mathrm{Q}} \left((\mathrm{y} - \mathrm{q_i})^T \mathrm{n}(\mathrm{x_j})\right)^2 \theta(\mathrm{y}, \mathrm{q_i}). \tag{2.39}$$

It has been proven in [224] that the MLS surface is actually the implicit surface function:

$$g(\mathrm{x}) = \mathrm{n}(\mathrm{x})^T \left(\left. \frac{\partial e(\mathrm{y}, \mathrm{n}(\mathrm{x}))}{\partial \mathrm{y}} \right|_{\mathrm{y}=\mathrm{x}} \right), \tag{2.40}$$

where $\mathrm{n} : R^3 \to R^3$ is the vector field defined in Eq. 2.37 and $e : R^3 \times R^3 \to R$ is the energy function defined in Eq. 2.39. Applying the curvature formulas for implicit surfaces given in [232], the Gaussian and mean curvatures of this implicitly defined MLS surface can be readily obtained as:

$$K_{Gaussion} = -\frac{Det \begin{pmatrix} H(g(\mathrm{x})) & \nabla^T g(\mathrm{x}) \\ \nabla g(\mathrm{x}) & 0 \end{pmatrix}}{||\nabla g(\mathrm{x})||^4}, \tag{2.41}$$

$$K_{Mean} = -\left(\frac{\nabla g(\mathrm{x}) \cdot H(g(\mathrm{x})) \cdot \nabla^T g(\mathrm{x})}{||\nabla g(\mathrm{x})||^3} - \frac{||\nabla g(\mathrm{x})||^2 \cdot Trace(H)}{||\nabla g(\mathrm{x})||^3}\right), \tag{2.42}$$

where $\nabla g(\mathrm{x}) = \begin{pmatrix} \frac{\partial g(\mathrm{x})}{\partial x} & \frac{\partial g(\mathrm{x})}{\partial y} & \frac{\partial g(\mathrm{x})}{\partial z} \end{pmatrix}^T$
is the gradient of $g(\mathrm{x})$,

and $H(g(\mathrm{x})) = \nabla(\nabla(g(\mathrm{x}))) = \begin{pmatrix} \frac{\partial g(\mathrm{x})}{\partial x \partial x} & \frac{\partial g(\mathrm{x})}{\partial x \partial y} & \frac{\partial g(\mathrm{x})}{\partial x \partial z} \\ \frac{\partial g(\mathrm{x})}{\partial x \partial y} & \frac{\partial g(\mathrm{x})}{\partial y \partial y} & \frac{\partial g(\mathrm{x})}{\partial y \partial z} \\ \frac{\partial g(\mathrm{x})}{\partial x \partial z} & \frac{\partial g(\mathrm{x})}{\partial y \partial z} & \frac{\partial g(\mathrm{x})}{\partial z \partial z} \end{pmatrix}^T$ is the Hessian matrix of $g(\mathrm{x})$. Notice that $Det(\mathrm{A})$ and $Trace(\mathrm{A})$ denote the determinant and the trace of the matrix A respectively.

2.5 Summary

This chapter reviews fuzzy inference and fuzzy inference systems, fuzzy rule interpolation, rule base generation and reduction, and previously curvature utilisation. The most common application of fuzzy logic is the rule-based fuzzy inference system, which is composed of mainly two parts: an inference engine, and a fuzzy rule base. Fuzzy rule base is constructed using any available information, which can be in the form of measured data or might come from some computational simulation. Traditional fuzzy rule base generation approaches usually only target dense rule bases that are used for traditional fuzzy inference approaches and most of them are based on certain similarity measures, therefore they are likely to cause performance deterioration along with the size reduction of the rule base.

This work presents a novel data-driven rule base generation approach for FRI, which directly generates sparse rule bases from datasets by effectively using curvature values traditionally utilised in geography. The proposed work not only can use curvature values to help perform feature selection and instance selection separately, but combine them simultaneously. It designed a curvature-based mechanism to effectively select the important data (also with the important features), thus to maintain very high accuracy even when there are only several rules left in the sparse fuzzy rule base.

Chapter 3

Curvature-based Sparse Rule Base Generation

It is important to keep a good balance between model accuracy, efficiency and transparency, as discussed in Section 1.4. Sparse rule base generation with the support of FRI is one solution which is performance-wise accurate, computationally efficient, and linguistically tractable. Therefore, a novel data-driven rule base generation approach for FRI is proposed, which directly generates sparse rule bases from datasets by using curvature values [233]. It should provide an objective tool to distinguish what is important in datasets and is helpful for solving existing issues in Section 1.4. In this chapter, an initial investigation into the feasibility of a curvature-based sparse rule base generation approach for FRI is carried. This chapter details the mechanism and the implementation of the proposed method. The whole work flow of the proposed method is summarised as in Fig. 3.1.

The proposed approach can directly generate sparse rule bases from datasets by effectively using curvature values traditionally utilised in geography. Note that curvature is an important concept in the field of geography, which is conventionally used to investigate the water flow over a landscape. Therefore, the curvature values are only workable in three-dimensional spaces (or a rule with two antecedents and one consequence) and thus cannot be directly used for higher-order problems. As a solution, for any given higher-order problem, the proposed approach firstly decomposes the higher-order space into a number of three-dimensional spaces, and then approximates the importance of the higher-order spaces by aggregating the curvature values of the corresponding decomposed three-dimensional ones. From this, the most important rules are selected to form a raw rule base, which is then optimised using a general optimisation approach, such as the genetic algorithm. Different to the conventional fuzzy rule base generation approaches (which usually only target dense rule bases that are used for traditional fuzzy inference approaches and most of them are based on certain

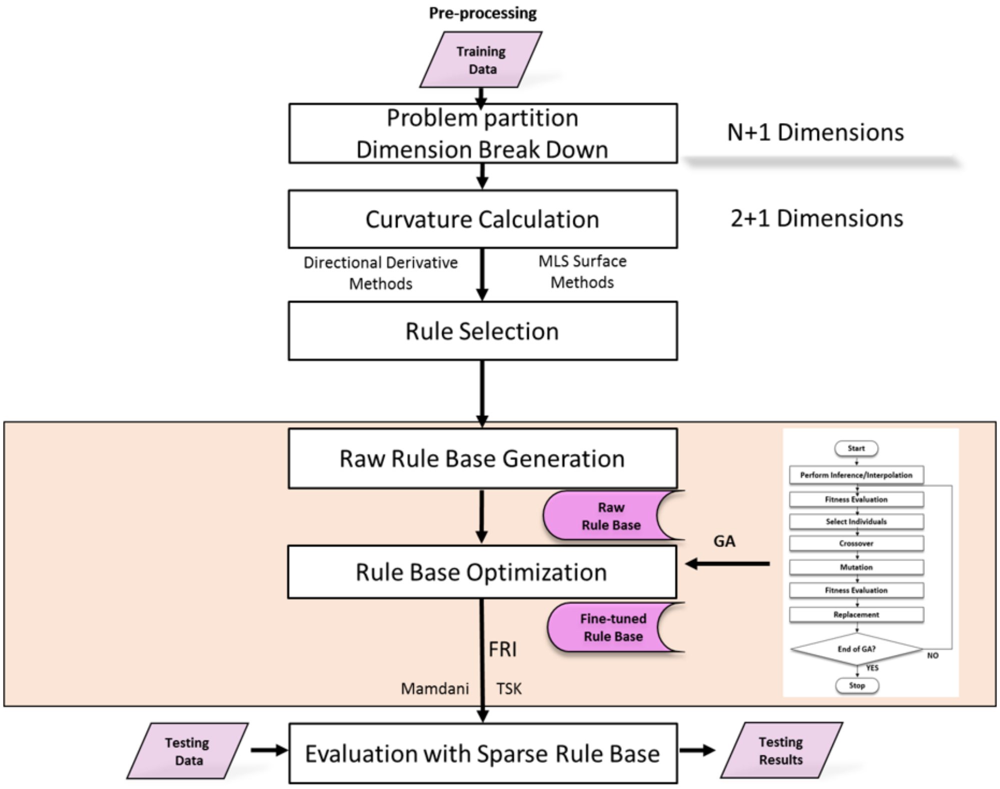

Fig. 3.1 Work flow of the proposed work

similarity measures, therefore they are likely to cause performance deterioration along with the size reduction of the rule base), the proposed approach discriminates rules by calculating their curvature values.

For better understanding, the rest of the chapter is organised as follows. Section 3.1 explains the key idea and the novel contribution of the curvature utilisation for fuzzy rule selection, which is the backbone of the proposed curvature-based sparse rule base generation method. Following this, in Section 3.2, the inference problems with two inputs and one output (referred to as the basic case) is considered first. In Section 3.3, the general situation with multiple inputs (referred to as the general case) is discussed. Finally, Section 3.5 summarises this chapter.

3.1 Curvature-Based Fuzzy Rule Selection

The proposed data-driven rule base generation approach for FRI is based on the utilisation of curvature, which is presented in this section. By artificially viewing an inference problem (such as classification, diagnosis or prediction) with two inputs and one output as a geometry object, the curvature values as introduced in Section 3.1 can be used to represent the linearity of the object surface. This then reveals the extents to which the geometric object deviates from being 'flat' or 'straight'. Considering that most of the existing FRI approaches are essentially fuzzy extensions of crisp linear interpolation [113], the 'flat' or 'straight' parts of the geometry object can be easily approximated by its surroundings, and therefore can be omitted. Given a training dataset with two input features and one output feature, the data instances that represent higher curvature values are more important in summarising and generalising the pattern entailed by the dataset. Therefore, they can be used to construct a sparse rule base or to simplify an existing complex rule base.

3.2 Basic Case with Two Inputs

A sparse rule base generation approach based on above motivation for a problem with two inputs and one output is presented below.

3.2.1 Problem Domain Partition

The partition approaches used in conventional fuzzy rule base generation methods, such as the partitioning and clustering approaches reported in [234], can also be used in this work. Specifically, if the training dataset is sparse, non-grid partition is applied; otherwise, grid partition is used. Given a dataset with two input features (x_1, x_2) and one output feature (y), denote the universe of discourse of the inputs to be $[\underline{x_1}, \overline{x_1}]$, $[\underline{x_2}, \overline{x_2}]$, and that of the output to be $[\underline{y}, \overline{y}]$. If the dataset is very sparse, each data instance in the dataset is used to represent a region. Therefore, the number of regions is equivalent to the number of the data instances in the dataset. If the dataset is dense, the domain is evenly divided into $n_1 * n_2$ regions. The values of n_1 and n_2 for a given problem are usually empirically determined. Large values of n_1 and n_2 lead to a large rule base, which requires more memory and greater computation efforts. Smaller values of n_1 and n_2 lead to a more compact rule base, which may only offer poorer performance. The general partition equation for more general case with multiple inputs (more than two inputs as in basic case) is summarised in 3.3.1.

3.2.2 Curvature-based Region Selection

Curvature values represent the 'straightness' or 'flatness' of a surface, which serves as an important description of intrinsic surface characteristics. Given that most of the existing FRI approaches are essentially fuzzy extensions of crisp linear interpolation, the 'flat' or 'straight' parts of a pattern can be easily approximated by its surroundings, as shown as the blue dots in the Fig. 3.2, which can be omitted. The parts with higher curvature values need to be explicitly represented by fuzzy rules, and the corresponding rules will be selected to initialise the rule base, as shown as the red ones in the figure. Thus, the important rules for FRI can be identified by the curvature values. Therefore, curvature is intuitively employed as the criterion to select the most important regions and hence to generate the most important rules in the implementation of FRI systems. The curvature value of a region is positively proportional to the importance of the region. So the criterion of instance selection can be summarised as: the higher the curvature value is, the more important the instance is. Based on this selection criterion, there are two ways to select the instances: by a curvature value threshold or by a pre-set number of needed rules for the sparse rule base. A predefined curvature threshold leads to a certain number of rules. Reversely, a predefined rule size implies a certain curvature threshold. The curvature threshold or the number of rules is problem-specific and are usually determined based on empirical study.

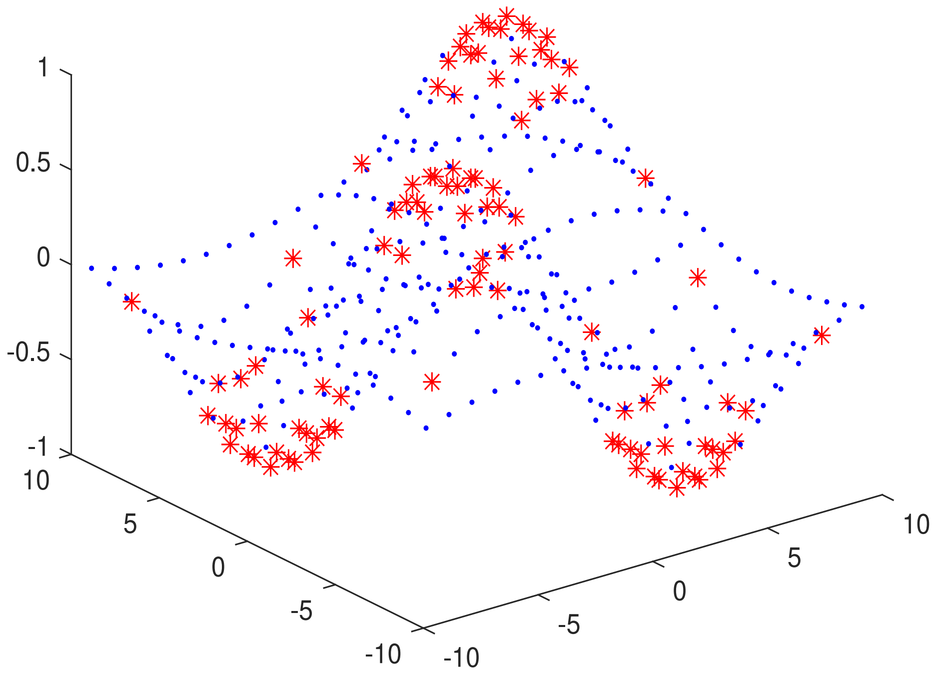

Fig. 3.2 Example for the curvature-based region selection

As discussed in Section 3.2.1, if the dataset is very sparse, a non-grid partition is applied. The curvature of each partitioned space representing a data instance can be directly calculated using the MLS-based curvature calculation approach as introduced in Section 3.1. Important data instances can then be selected using either a rule size threshold or curvature value threshold, and the selected important data instances can be directly used for rule base initialisation as introduced in the next subsection.

If a given dataset is dense, grid-partition is used. In order to balance cost and performance, a hierarchical partition and region selection approach is proposed herein to support the rule base generation. The approach is implemented in a recursive manner, and the pseudo-code of the approach is shown in Figure 3.3 and Algorithm 1. In this algorithm, if the curvature value of a region is greater than the activating threshold θ and is less than the ceiling threshold $\theta * (1+p)$, this region will be selected to generate a rule. If the curvature value of the region is less than the threshold θ, the region will be discarded. However, if the curvature value of the region is very large (i.e., larger than $\theta * (1+p)$), the region cannot accurately be represented by one rule. In this case, this region is further partitioned with the activating threshold θ being updated as $\theta * (1+p)$. Following this, the procedure is recursively applied to each of the further partitioned regions until no region needs to be further partitioned. Note that the curvature threshold is used in this algorithm; the rule size threshold can be applied in a similar and straightforward way, which is thus omitted here.

3.2.3 Rule Base Initialisation

Each selected region is expressed as a fuzzy rule. If the region is led by the non-grid approach, the corresponding data instance is used to represent the region. Denote the fuzzified value of the representative of each region as (A_1, A_2, B); the generated corresponding rule for the region can be expressed as:

$$\textbf{If } x_1 = A_1 \text{ and } x_2 = A_2, \textbf{ then } y = B. \tag{3.1}$$

For simplicity, only isosceles triangular membership functions are utilised in this work. In other words, each fuzzy set A_1, A_2, or B can be defined by a pair (c, s) with c representing the normal point and s representing the support of the fuzzy set, as commonly used in the literature [235]. In this work, the data instance that represents the region is intuitively used to represent the normal point of the triangular membership function, while a fixed support is initially applied to every fuzzy values of a data instance.

If the dataset is dense (and accordingly grid-partition is employed), each region usually covers multiple data instances, and a representative data point is required to represent the region for fuzzy rule generation. This is implemented by firstly aggregating all the data

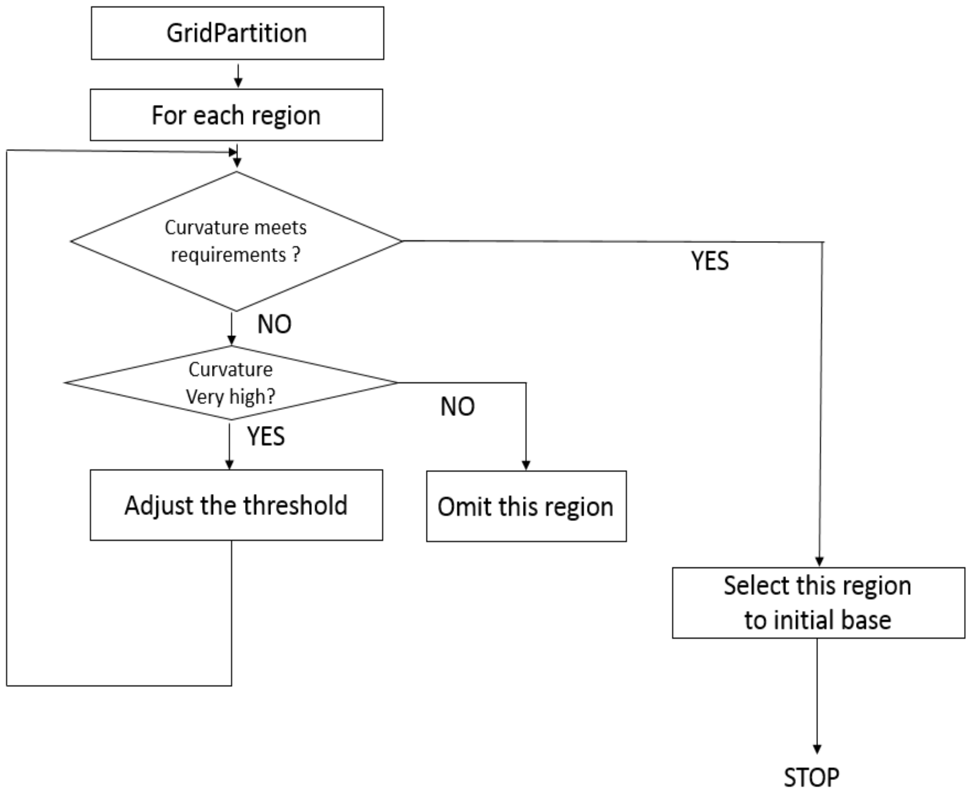

Fig. 3.3 Flowchart for the curvature-based region selection

instances into one artificially made data instance, and then such an artificially made data instance is fuzzified using the same approach as discussed above.

Many regression methods are discussed in 2.3. For simplicity, as the curvature values of the data instances already imply their importance, the curvature values are intuitively employed as a means to rank their weights in the WAA method. Suppose a given region is formed by n data instances, the artificially made data instance representing the selected region can be calculated as follows:

$$c = WAA(a_1, a_2, ..., a_n) = \sum_{i=1}^{n} w_i a_i, \tag{3.2}$$

where a_i $(1 \leq i \leq n)$ represents the ith data instance included in the selected region, and w_i $(1 \leq i \leq n)$ is the weight ranked by its curvature value:

$$w_i = \frac{C_i}{\sum_{i=1}^{n} C_i}, \tag{3.3}$$

where C_i represents the curvature value of the i^{th} data instance.

This work makes great contribution of using curvature for fuzzy rule selection, not brings the idea but also perfectly implements it, while no other literature ever considering about it.

3.3 General Case with Multiple Inputs

The majority of real-world applications consists of more than two inputs. Thus, the approach proposed in the last section needs to be extended. Given that traditional curvature values only work with three-dimensional data, the most challenging part to evaluate the importance of a high-dimensional instance is that there is no exiting approach to be directly applied for calculating the 'curvature' value of a high-dimensional instance. However, a higher-dimensional complex problem can be regarded as a collection of three dimensional problems with two inputs and one output (i.e., multiple basic cases). With the curvature-based approach discussed in Section 3.2, any high-dimensional problems can thus be addressed by applying the basic case solutions multiple times.

3.3.1 Problem Domain Partition

Suppose that a complex problem P_{n+1} $(n > 2)$ contains n input features $\mathbb{X} = \{x_1, x_2, \cdots, x_n\}$ and one output feature y, and that the universe of discourse of the input features is $[\underline{x_1}, \overline{x_1}]$, $[\underline{x_2}, \overline{x_2}]$, ..., $[\underline{x_n}, \overline{x_n}]$, whilst that of the output is $[\underline{y}, \overline{y}]$. Similar to the problem domain partition for the basic cases, each data instance in the dataset represents a hypercube in the problem domain. Therefore, the number of hypercubes will be equivalent to the number of the data instances in the dataset if the dataset is very sparse. In this case, each hypercube is represented as a fuzzy rule.

If the dataset is dense, grid partition is applied. In this situation, the input domain is evenly partitioned into $m_1 * m_2 * ... * m_n$ hypercubes, where m_i, $1 \leq i \leq n$, represents the number of partitions in the variable domain of x_i. The values of m_i are usually empirically determined by experts or statistically calculated by clustering methods such as k-means.

Traditional fuzzy partition methods represent each hypercube as a fuzzy rule. Hence, any given complex problem P_{n+1} will lead to a rule base with $m_1 * m_2 * ... * m_n$ fuzzy rules [236].

For simplicity, let h be the number of data instances if the given dataset is sparse, and $h = m_1 * m_2 * ... * m_n$ if the given dataset is dense. Accordingly, the generated hypercubes can be collectively represented as $\mathbb{H} = \{H_1, H_2, \cdots, H_h\}$. Note that some of these rules may not be necessary or can be represented by their neighbouring ones. Therefore, the importance of the hypercubes needs to be discriminated such that a sparse rule base can be generated based on the most significant hypercubes.

Here the input and output spaces are divided into fuzzy regions or hypercubes. Given a set of examples with multiple inputs (m) and single output, denoted as $(x_j^k; y^k)$ where j =1, . . ., m and k =1, . . ., n. Define the universe of discourse of each input variable as $[x_j^-; x_j^+]$ and the output variable as $[\, y^-; y^+]$ and then divide each universe of discourse into N intervals. The minimal and maximal values of each variable are often used to define its universe of discourse. That is $[x_j^-; x_j^+]$=$[min(x_j), max(x_j)]$. They are also considered to be the centre of the left end term and the right end term, respectively. That is, $c_{1j} = min(x_j)$ and $c_{Nj} = max(x_j)$. Accordingly, the other term centre of each interval, c_{ij}, can be computed as follows:

$$C_{ij} = min(x_j) + \ i(max(x_j)) - \ min(x_j))/(N-1), \ where \ i = 2, ..., N-1. \tag{3.4}$$

The selected regions or hypercubes can be represented by fuzzy rules as in 3.3.5.

3.3.2 Representing Hypercube by Cubes

The traditional curvature value is only applicable in a geometric space with three dimensions. Hence, there is no equation to directly calculate a 'curvature' value for a high-dimensional instance, and thus to directly use the value for representing its importance. However, based on the curvature values of its decomposed cubes, the importance of a high dimensional hypercube can still, to some extent, be identified. In order to distinguish important hypercubes, every high-dimensional hypercube H_i, $1 \leq i \leq h$, is broken down into $c = C_n^2 = \frac{n!}{2!(n-2)!}$ cubes. This is done by considering all the combination of two input features and the output feature. Therefore, the collection of hypercubes and the corresponding decomposed cubes can be represented as:

$$\mathbb{H} = \Big(H_1, H_2, \cdots, H_h\Big) = \begin{pmatrix} C_{11} & C_{21} & C_{31} & \cdots & C_{h1} \\ C_{12} & C_{22} & C_{32} & \cdots & C_{h2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C_{1c} & C_{2c} & C_{3c} & \cdots & C_{hc} \end{pmatrix}, \tag{3.5}$$

where C_{ij} represents the jth cube of the ith hypercube H_i, $1 \leq i \leq h$ and $1 \leq j \leq c$. Note that cubes C_{1j}, C_{2j}, $\cdots$, C_{hj} share the same two input features for any $1 \leq j \leq c$.

The importance of a hypercube can be collectively determined by its decomposed cubes. That is, the importance of hypercube H_i can be determined by the curvature values of $C_{i1}, C_{i2}, \cdots, C_{ic}$. The curvature value v_{ij} of each artificially created cube C_{ij} can be calculated using the approach detailed in Section 3.1. Collectively denoting the values of all cubes as V, the calculated results of all cubes can then be represented as follows:

$$V = \begin{pmatrix} v_{11} & v_{21} & v_{31} & \cdots & v_{h1} \\ v_{12} & v_{22} & v_{32} & \cdots & v_{h2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{1c} & v_{2c} & v_{3c} & \cdots & v_{hc} \end{pmatrix}. \tag{3.6}$$

3.3.3 Hypercube Selection

The importance of each hypercube can be represented by the summation of the curvature values of its decomposed cubes. In particular, given a set of hypercubes $\mathbb{H}$ and the required number of rules m in the to-be-generated rule base or an accumulated curvature threshold θ, the algorithm selects a set of the significant hypercubes as the output $\mathbb{H}'$, using a way similar to the approach presented in Section 3.2.2. Note that if the parameter θ is given, the algorithm works on the same principle as Algorithm 1; thus, the details are omitted here.

If the parameter m is given instead of using θ, the hypercube selection process is summarised in Figure 3.4 and Algorithm 2. The most significant m instances or hypercubes can be selected simply by taking the first m hypercubes with the highest accumulated curvature values after ranking them in descending order, as expressed in Line 8. The accumulated curvature value regarding a hyper cube H_i, represented as $H_i.weight$, can be calculated as the summation of all its related decomposed cubes (as expressed in Line 5).

Here the summation rather than maximum is used , which to some extent eases the influence of noise. Furthermore, some hypercubes which has no significant highest values in some cube 3D dimensions may still be very important because it may have relative higher curvature values in many cube 3D dimensions. However, in some experiments, the utilisation of maximum probably produce better results, which should be further investigated.

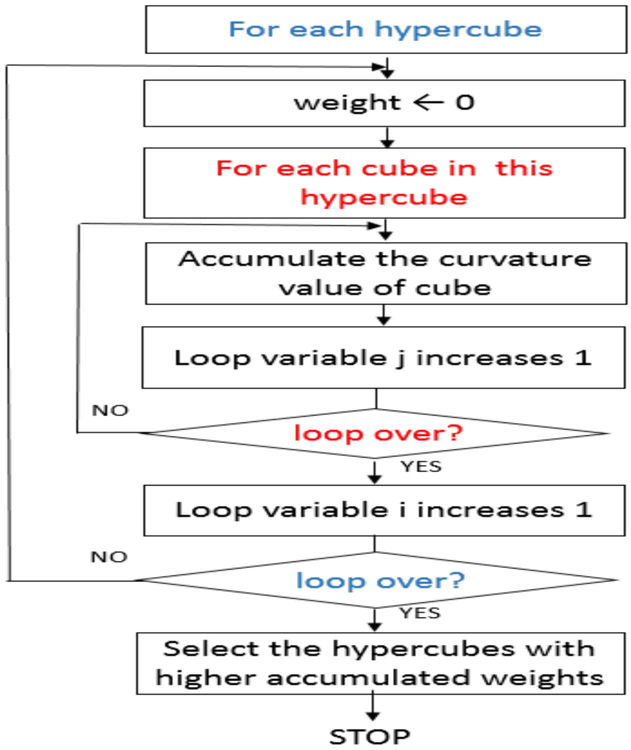

Fig. 3.4 Flowchart for the hypercube selection

3.3.4 Feature Discrimination

In addition to the fact that some hypercubes are more important than the others, some dimensions in a selected hypercube may also be more significant than the others as demonstrated by many feature selection approaches [237–241]. Therefore, selective dimensionally reduced hypercubes can be used to generate a more compact rule base with fewer rule antecedents. This is summarised in Figure 3.5 and Algorithm 3. It takes the output of $HypercubeSelection()$ as its input, in addition to the number of selected features b and the training data T. The output of the algorithm is a subset of input features $\mathbb{X}'$, which represent the most significant input features from the entire set of input features $\mathbb{X}$. The number of selected input features "b" is selected empirically, usually no more than 20% of total feature number.

The algorithm artificially generates c set of two-input and one-output fuzzy rule bases to evaluate the importance of the pair of associated features. Recall that since $C_{1j}, C_{2j}, \cdots C_{hj}$ share the same input features in Eq. 3.5, they jointly form the jth artificial rule base. Every

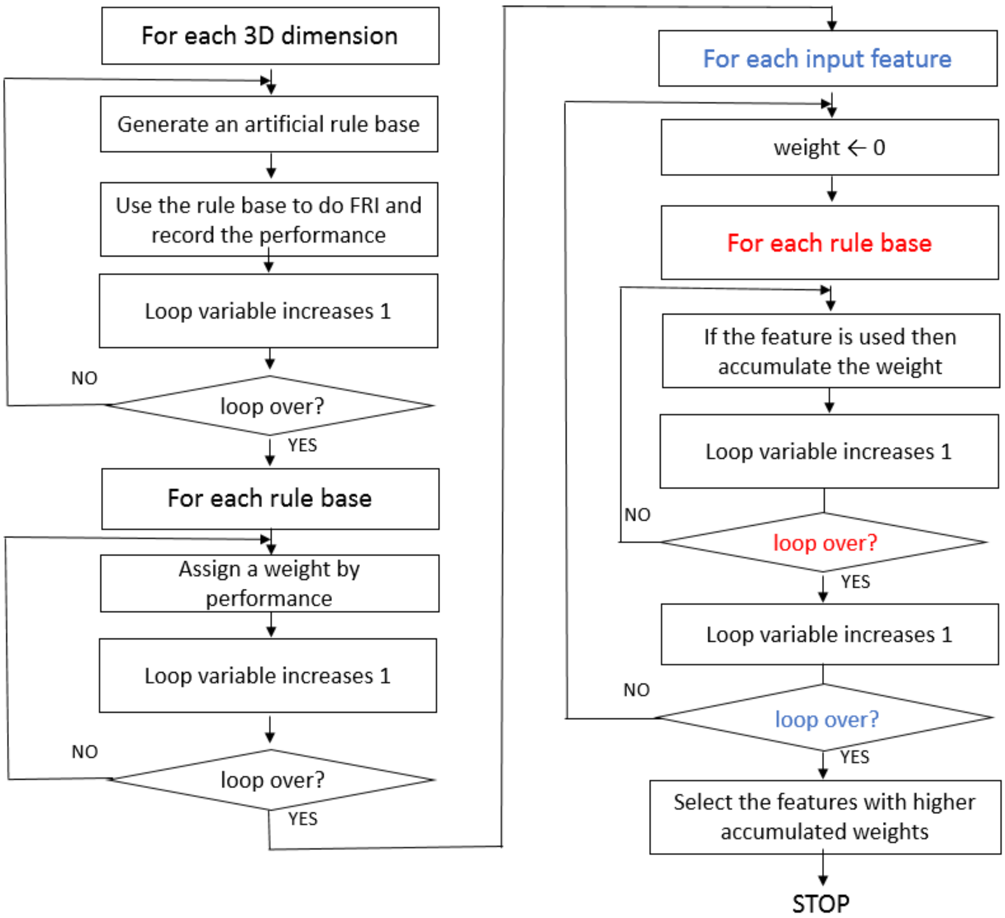

Fig. 3.5 Flowchart for the feature discrimination

pair of input features is then evaluated by its corresponding rule bases by applying the training dataset to perform FRI with the rule bases. From this, all the artificial rule bases, representing every pair of features, can be ranked. A weight function is designed to convert the ranking to a weight for the pair of features. To avoid that the curvature values calculated in different 3D dimensions may has different metrics, here the weight function is using the rank index number of each cube in each basic case. Note that each input feature appears in $|\mathbb{X}-1|$ artificial rule bases. Therefore, the importance of the feature is calculated as the summation of the weights of all the rule bases that consider such feature.

Note that the proposed method provides an important key to solve the issues of 'curse of dimensionality', which means that as many attributes and attribute values need to be considered, the number of rules grows exponentially with the number of input variables and the number of fuzzy terms. The difficulty of feature selection lies in how to find good clues to select feature variables that can most efficiently represent the intrinsic characteristics of the data. Its essence is a complex combinatorial optimization problem. Irrespective of all desirable features of fuzzy rule interpolation techniques the success of a particular method depends on its computational complexity, which should be minimised. For example, if there are n feature variables, each feature variable has two possible states when we are modelling: “selected" and “omitted". Then the number of elements in the sets of feature combination dimensions is 2^n. If n = 3, the traditional exhaustive methods need 8 combinations [242]. However, use the proposed curvature based method the curse of dimensionality is greatly reduced as only need 3 combinations. That means, it reduced the traditional feature combination dimensions from 2^n to $C_n^2 = \frac{n!}{2!(n-2)!}$, when n is large the reduction number is very huge. For example, If n = 20, the proposed method reduces the combinations dimensions from 1048576 to only 190, which is less than the 0.02% of traditional one.

3.3.5 Rule Base Initialisation

Similar to the basic case, each hypercube with selected features is expressed as a fuzzy rule. Denote the fuzzified values of a data instance or a hypercube with b selected input features and the output feature as $(A_{k1}, A_{k2}, ...A_{kb}, B_k)$; the corresponding fuzzy rule can be expressed as:

$$\begin{aligned}&\textbf{If } x_1 = A_{k1}, x_2 = A_{k2}, \cdots, \textbf{ and } x_b = A_{kb},\\&\textbf{then } y = B_k.\end{aligned} \tag{3.7}$$

Note that the above rule base is a Mamdani-style rule base, as the consequence of each rule is a fuzzy set [243]. However, depending on the real-world application, a TSK-style rule base may also readily be generated using the existing TSK rule base generation approaches [244]. Note that the resultant rule base has only b antecedents, which is a subset of the input features. In an extreme situation, the resultant rule base may only have two input features, which backtracks to the basic case as discussed in Section 3.2.

3.4 Rule Base Optimisation

The initialised rule base can be improved by fine-tuning the involved fuzzy sets, given that the initial fuzzy sets are specified based on empirical knowledge. This can be implemented using any general optimisation approaches; the genetic algorithm (GA) is particularly employed in this work due to its effectiveness in rule base optimisation [125, 140, 245]. Some other methods can also be used to optimise the if-then rules such as [211].

Genetic Algorithms (GAs) are a class of stochastic search and optimization procedures that are inspired by the Darwinian principle of survival of the fittest individuals and natural selection [246]. GAs can find closest optimal solutions in complex search spaces depending on their defined parameters. GAs differ from classical optimisation and search methods in the following ways [247]:

1. GAs work with a coding of the parameter set, not the parameters themselves.
2. GAs search from a population of points, not a single point.
3. GAs use fitness function, not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

Genetic algorithms can be used to encode a possible solution to a given problem using a chromosome-like data structure, applying recombination operators to these structures in order to exploit vital information. Generally, a random population of chromosomes is generated at the beginning of the GA implementation. These chromosomal structures are then evaluated by allocating reproductive opportunities to those chromosomal structures which better fit the solution domain, whist discouraging similar opportunities with poorer solutions. The quality of a solution is naturally defined with respect to the present population [248]. Their operation is dependent on two important operators: crossover and mutation. The population (the set of chromosomes) is initially generated randomly and their members are then selected for reproductive process with respect to their fitness values. The chromosomes with higher fitness values have better chances to reproduce. The reproductive process is repeated until desired conditions are met, such as a desired fitness level, or a maximum number of generations.

3.4.1 Population Initialisation

Specifically, a chromosome is designed to represent all the fuzzy sets in the initialised rule base [249]. Given the fixed representative value and isosceles shape of fuzzy set A, A can be readily constructed from the support of A (denoted as $S(A)$). Then, the ith rule in Eq. 3.7 can be represented by $b+1$ parameters: $S(A_{i1})$,$S(A_{i2})$,...,$S(A_{ib})$ and $S(B_i)$. Thus a chromosome representing all the rules in the rule base has $(b+1)*m$ genes, as illustrated in Fig. 3.6, where m represents the number of rules in the rule base. Following this, the initial population can then be generated by creating a number of individuals, i.e., $\mathbb{P} = \{I_1, I_2, ..., I_{|\mathbb{P}|}\}$. Each individual I_i is a chromosome representing a potential solution to the given problem, where $1 \leq i \leq |\mathbb{P}|$.

Typically, the size of population ($|\mathbb{P}|$) is determined based on the given problem, and it may typically contain from several hundreds to several thousands of individuals, each of which is a potential solution to the problem. The first individual I_1 in this case represents exactly the raw rule base generated through the approach discussed in Section 3.2.3 and Section 3.3.5, whilst the other individuals are modified versions of the first one. Intuitively, the evolved individuals should have a larger chance to be similar to the one in the raw rule base than others. Therefore, individuals $\{I_2, I_3 ..., I_{|\mathbb{P}|}\}$ are randomly generated such that the modified support values of a particular rule antecedent or consequence follow a normal distribution. In particular, suppose that the value at position i of individual I_1 is $supp_i$, then the i^{th} position for the rest of the individuals can be generated using a Gaussian distributed random number generation approach such as the classical Box-Muller-Wiener algorithm, with $supp_i$ being the expected value [250].

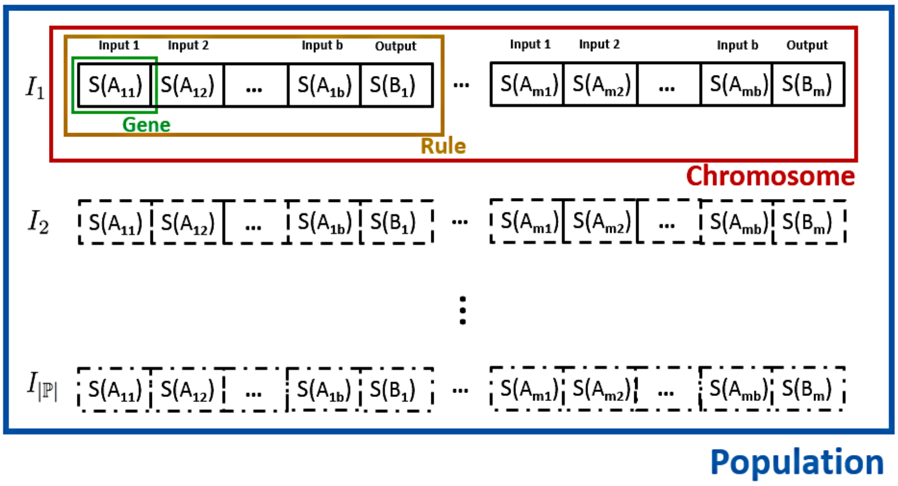

Fig. 3.6 Chromosome representation in GA

3.4.2 Fitness Evaluation

An objective function is used in GA to measure the fitness or quality of individuals. Typically, in this initial work, the objective function is defined as the root mean square of the error (RMSE). Given an individual I_i, $1 \leq i \leq |\mathbb{P}|$, the RMSE value regarding this individual can be calculated as follows:

$$RMSE_i = \sqrt{\frac{\sum_{j=1}^{m}(z_j - \hat{z}_j)^2}{m}}, \tag{3.8}$$

where m is the size of the training data set; z_j is the labeled (defuzzifised) output value of the j^{th} training data instance and $\hat{z}_j$ represents the (defuzzifised) output that is generated by a particular FRI approach.

In this case, the fittest individuals will have the lowest numerical value of the associated objective function. As the 'roulette wheel' selection method is used in the current work to probabilistically select individuals for reproduction, a fitness function is used to transform the objective function value into a measure of relative fitness [251], in an effort to prevent premature convergence by limiting the reproductive range so that no individuals generate an excessive number of offspring. The fitness of an individual I_i in the current work is calculated as follows [251]:

$$f(I_i) = 2 - max + \frac{2(max-1)(r_i-1)}{|\mathbb{P}|}, \tag{3.9}$$

where r_i is the ranking position of individual I_i in the ordered population $\mathbb{P}$, and max is the bias or selective pressure, towards the fittest individuals in the population.

3.4.3 Reproduction and Selection

Two genetic operations (crossover and mutation) are used to produce the next generation of the population. The population $\mathbb{P}$ are ranked based on their fitness values, and then k pairs of individuals or elites are selected using a 'roulette wheel' mechanism to produce the next generation of individuals by crossover and mutation.

Crossover exchanges contiguous sections of the chromosomes, which takes two parent solutions and produces two children solutions from them. Each pair of individuals in the k pairs of selected elites acts as the parents for reproduction. A single crossover point on each pair of selected parents, organism string is selected, and all data after the index point of the two parents are swapped. The resulting individuals (after mutation) will be part of the next generation, denoted as $\mathbb{P}'$.

Mutation is used to maintain genetic diversity from one generation of a population to the next, which simulates biological mutation. Mutation alters one gene value in a chromosome from its initial state, which helps the algorithm to avoid local minima by preventing the population of individuals from becoming too similar to each other. A certain percentage of offspring in $\mathbb{P}'$ are selected to take the mutation operation. In addition, a single or multiple points of a rule can take the mutation procedure. Particularly in the current work, the mutation procedure produces a random support value regarding a particular fuzzy set for the mutation point using another Gaussian random number generation approach.

Suppose that the second generation of population ($\mathbb{P}'$) has been generated. The fitness function, Equation 3.9, will be employed again to determine the quality of each individual in

$\mathbb{P}'$. Finally, the best individuals in $\mathbb{P}'$, which are represented by the smallest fitness values, will be selected and used to replace the worst ranked individuals in $\mathbb{P}$, thus completing one iteration of the GA searching process by generating a new population of solution $\mathbb{P}''$. The entire process of selection and reproduction is shown in Fig. 3.7.

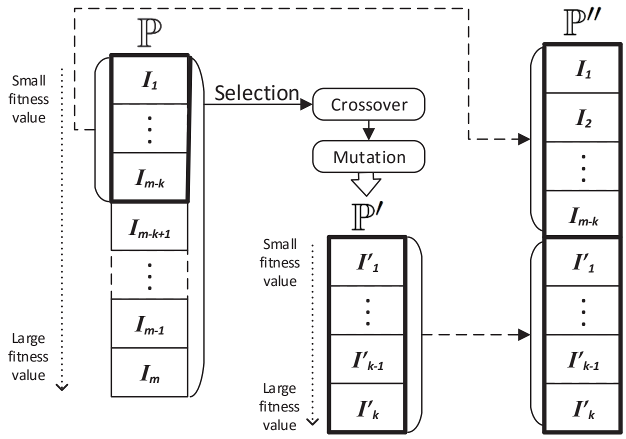

Fig. 3.7 Selection and reproduction process

3.4.4 Termination

The reproduction procedure is repeated until the pre-specified maximum number of iterations is reached or the objective value of an individual is less than a predefined threshold. When the GA terminates, the fittest individual in the current population is the optimal solution. Following this, a number of the best individuals in the next generation are selected and used to replace the worst-ranked individuals in the initial population. In this way, one iteration of the GA process is completed. This process is iterated until the pre-specified maximum number of iterations is reached or the objective value of an individual is less than a predefined threshold. When the GA terminates, the fittest individual in the current population is the optimal solution.

Notice that GA has been used to optimise fuzzy rule bases to support FRI in [252]. Specifically, in the current work, GA was used for clustering, which is the process of grouping similar interpolated rules to form clusters. These clusters were further used to form new, aggregated rules to update the existing rule base. Therefore, GA was used in the rule base updating stage after a number of inference iterations. In contrast, in this proposed approach, GA is used to optimise the raw rule base before it is used for inference. The introduction of GA to rule base generation indeed provokes high computational complexity, but this is acceptable, as the rule base generation is a one-time process that is carried out offline.

3.5 Summary

This work presents a novel data-driven rule base generation approach for FRI, which is able to directly generate a sparse rule base from data. In particular, the approach firstly divides complex problem into small problems, each with the basic case of $2+1$ dimensions. Then in each basic case the sub-problem is fuzzy-partitioned into a number of sub-regions and each sub-region can be represented as a fuzzy rule. Following this, the curvature of each sub-region is calculated. As detailed in Section 3.2.2, curvature values of regions (instances) in the training data to some extent represent their importance for solving the given problem. By artificially viewing the pattern to be modelled hidden in the training data set as a geometry object, the curvature values can be used to represent the linearity of the surface of the hidden pattern, as they reveal the extent to which a geometric object deviates from being 'flat' or 'straight'. Given that most of the existing FRI approaches are essentially fuzzy extensions of crisp linear interpolation, the 'flat' or 'straight' parts of a pattern can be easily approximated by its surroundings, thus they can be omitted. In contrast, the regions (instances) with higher curvature values can make prominent effects to the conclusion and are necessary to be selected to construct the sparse rule base. Those sub-regions which have higher curvature values are then identified, and the corresponding rules will be selected to initialise the rule base. In the complex problem or model, the performances of all rule bases in the basic cases are collected to identify the important features, thus the most important instances (or hypercubes) with their important features will be selected to generate the final rule base. The initial rule base can be optionally optimised by fine-tuning the membership functions of the fuzzy sets involved in the rules by using a genetic algorithm (GA) optimisation method. The proposed method will be evaluated in the next chapter.

Chapter 4

Applications of Curvature-based Sparse Rule Base Generation

The proposed method was evaluated using a simulated experiment and three real-world applications, i.e., an indoor environment localisation problem, a student knowledge level evaluation problem, and the zero-shot learning image classification problem. The experiments demonstrate the working procedure of the proposed approach, while the applications show the power of the proposed approach in solving real-world problems. The proposed method provides a flexible and effective framework to generate a sparse rule base. It efficiently and accurately utilises the available data to generate a sparse rule base for FRI and then inference results which outperforms the state-of-the-art methods. Although there are curvature calculation methods existing in the literature, no work has so far been done to apply the curvature idea into sparse rule base generation for fuzzy rule interpolation, and no work has applied the curvature to solve the problems in the fields of the above three real-world applications. For example, it is the first time to introduce curvature-based sparse rule base generation and fuzzy rule interpolation into Zero-Shot learning image classification (computer vision field). The program was developed using MATLAB (R2017b) and was run on a laptop with the Intel i7-4810MQ CPU 2.8 GHz and 16 GB of RAM, in windows 7.

4.1 Experimentation Design

Firstly, for better understanding, a simulation experimentation as basic case is explained, with only 2 input features and 1 output feature, to help readers easily understood the key idea of the curvature based sparse rule base generation. This can easily explain the mechanism and the power of the curvature utilisation.

Secondly, a complex application for indoor environment localisation is used, with 7 multiple input features and 1 out feature (4 classes), and with totally 2000 instances but evenly distributed with four output classes (each class 500 instances). This experiment shows the proposed method works well in general case with multiple inputs.

Thirdly, a more complex application with student knowledge level evaluation is used, with 5 input features and 1 output feature (4 classes). Note that both the training and testing instances are not evenly distributed with four output classes, and the output are even in linguistic expert knowledge (Very Low: 50, Low: 129, Middle: 122, High 130). This experiment shows the proposed work can also work for linguistic environment and can provide an objective way to distinguish what is important thus somehow ease the subjective expert knowledge.

Finally, some much more complex ZSL (zero-shot learning) applications are used, to validate the proposed method in state-of-the-art hot fields. For example, the AWA (Animal with Attributes) dataset with 4096 input features and 1 output feature (50 classes), and with totally 37322 instances. This experiment shows that the proposed method introduce the curvature based sparse rule base generation into the computer vision field at the first time, and it can infer full associations between seen and unseen classes using only a few similes. The number of required labelling is reduced from 50x85 to only 10x2 that is only 0.47% of the original annotation work.

4.2 Simulated Experiment

4.2.1 Problem Partition

The problem presented in [4, 58, 244] is reconsidered here for a comparative study, which models the non-linear function as defined as:

$$f(x,y) = \sin\left(\frac{x}{\pi}\right)\sin\left(\frac{y}{\pi}\right),$$

where the domains of the two inputs are $x \in [-10, 10]$ and $y \in [-10, 10]$, and the domain of the output is $z \in [-1, 1]$.

In order to reveal such a mathematical model, the approach first uniformly partitions the problem space into 20×20 grid areas, resulting in a total of 400 sub-regions, as illustrated in Fig. 4.1. The input domain of the variable x has been divided into 20 equal intervals, with each represented as a fuzzy set, as shown in Fig. 4.2. This is also the case for the variable y.

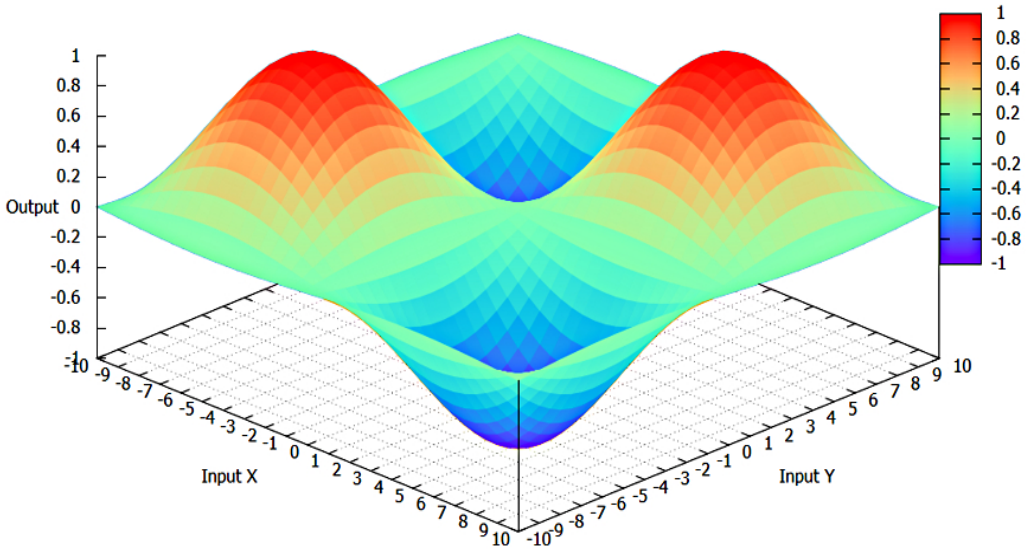

Fig. 4.1 Problem space partition for the illustrative example

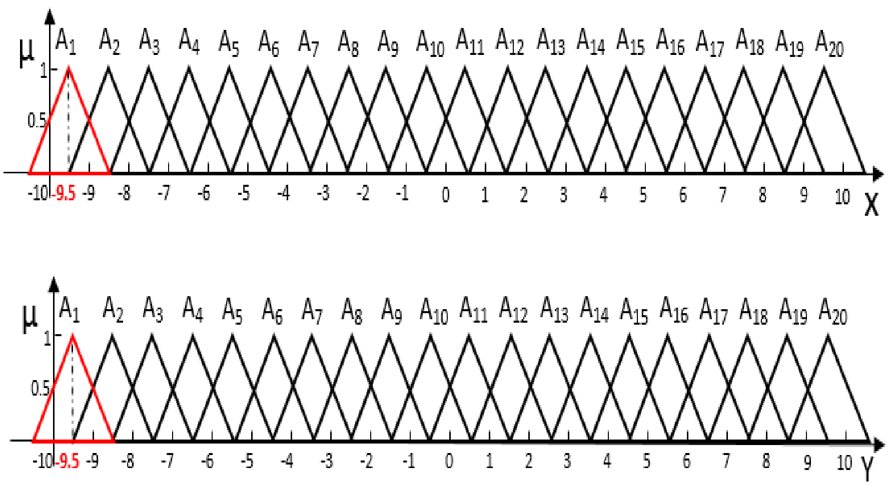

Fig. 4.2 Fuzzy partition of the input domain

4.2.2 Curvature Calculation and Rule Selection

The degree of flatness or sharpness of each such sub-region can be represented by its curvature value. The curvature values of all sub-regions are calculated using the profile curvature from the directional derivative method, with the results listed in Table 4.1. For better revealing the hidden model, the sub-regions with different curvature values are also marked with different colours as summarised in Fig. 4.5. For example, in the first sub-region (where x is around -9.5 and y is around -9.5), the curvature value is 0.099. This figure is relatively high, which means that the sub-region is relatively sharp and cannot be easily approximated by neighbouring sub-regions. This selected sub-region is represented as a fuzzy rule for initialising the rule base, which is optimised using the GA.

The sub-regions with higher curvature values are more important in summarising and generalising the pattern entailed by the dataset. Therefore, they can be used to construct a initial sparse rule base. The important regions can be selected according to the curvature values. Each selected region is expressed as a fuzzy rule.

4.2.3 Rule Base Optimisation

The initialised rule base includes all the most important rules that cannot be accurately represented by their neighbouring ones, but these rules are not optimal in terms of their membership functions. After generating the raw rule base, the optimisation algorithm GA is employed to fine-tune the membership functions involved in the rules. In this experiment, the population size was set as 100, with the first individual in the population configured to represent exactly the generated raw rule base. All other individuals were randomly generated using the approach introduced in Section 3.4. The *max* value parameter of the fitness function (Equation 3.9) was set to 2, the maximum number of generations was set to 1000, and the probabilities of crossover and mutation were set to 0.8 and 0.01, respectively. The optimised rules of a rule base with 23 rules are summarised in Table 4.3.

The optimisation generally leads to a decrease of the average error by 5% to 20% compared to the error resulting from the employment of the raw rule base. When the rule base contains a relatively large number of rules, a significant improvement in accuracy was recorded, as shown in Fig. 4.4, where the rule base with 12 rules has the smallest error; conversely, when the rule base consists of fewer rules, only a very small improvement was recorded, as demonstrated in Fig. 4.3, where rule bases with 8 and 4 rules have smaller errors.

Table 4.1 Curvature values of the sub-regions

x/y Interval	1	2	3	4	5	6	7	8	9	10
...	11	12	13	14	15	16	17	18	19	20
1/1-10	**0.0985**	0.0385	0.0100	-0.0043	-0.0109	-0.0113	-0.0055	0.0076	0.0333	**0.0930**
1/11-20	**-0.0930**	-0.0333	-0.0076	0.0055	0.0113	0.0109	0.0043	-0.0100	-0.0385	**-0.0985**
2/1-10	0.0385	0.0652	0.0240	-0.0167	-0.0393	-0.0407	-0.0208	0.0180	0.0615	0.0491
2/11-20	-0.0491	-0.0615	-0.0180	0.0208	0.0407	0.0393	0.0167	-0.0240	-0.0652	-0.0385
3/1-10	0.0100	0.0240	0.0063	-0.0323	-0.0639	-0.0661	-0.0375	0.0019	0.0235	0.0132
3/11-20	-0.0132	-0.0235	-0.0019	0.0375	0.0661	0.0639	0.0323	-0.0063	-0.0240	-0.0100
4/1-10	-0.0043	-0.0167	-0.0323	-0.0550	-0.0824	-0.0849	-0.0586	-0.0348	-0.0185	-0.0058
4/11-20	0.0058	0.0185	0.0348	0.0586	0.0849	0.0824	0.0550	0.0323	0.0167	0.0043
5/1-10	-0.0109	-0.0393	-0.0639	-0.0824	**-0.0948**	**-0.0963**	-0.0843	-0.0667	-0.0428	-0.0148
5/11-20	0.0148	0.0428	0.0667	0.0843	**0.0963**	**0.0948**	0.0824	0.0639	0.0393	0.0109
6/1-10	-0.0113	-0.0407	-0.0661	-0.0849	**-0.0963**	**-0.0975**	-0.0868	-0.0690	-0.0443	-0.0153
6/11-20	0.0153	0.0443	0.0690	0.0868	**0.0975**	**0.0963**	0.0849	0.0661	0.0407	0.0113
7/1-10	-0.0055	-0.0208	-0.0375	-0.0586	-0.0843	-0.0868	-0.0619	-0.0399	-0.0228	-0.0075
7/11-20	0.0075	0.0228	0.0399	0.0619	0.0868	0.0843	0.0586	0.0375	0.0208	0.0055
8/1-10	0.0076	0.0180	0.0019	-0.0348	-0.0667	-0.0690	-0.0399	-0.0021	0.0175	0.0100
8/11-20	-0.0100	-0.0175	0.0021	0.0399	0.0690	0.0667	0.0348	-0.0019	-0.0180	-0.0076
9/1-10	0.0333	0.0615	0.0235	-0.0185	-0.0428	-0.0443	-0.0228	0.0175	0.0585	0.0429
9/11-20	-0.0429	-0.0585	-0.0175	0.0228	0.0443	0.0428	0.0185	-0.0235	-0.0615	-0.0333
10/1-10	**0.0930**	0.0491	0.0132	-0.0058	-0.0148	-0.0153	-0.0075	0.0100	0.0429	**0.0962**
10/11-20	**-0.0962**	-0.0429	-0.0100	0.0075	0.0153	0.0148	0.0058	-0.0132	-0.0491	**-0.0930**
11/1-10	**-0.0930**	-0.0491	-0.0132	0.0058	0.0148	0.0153	0.0075	-0.0100	-0.0429	**-0.0962**
11/11-20	**0.0962**	0.0429	0.0100	-0.0075	-0.0153	-0.0148	-0.0058	0.0132	0.0491	**0.0930**
12/1-10	-0.0333	-0.0615	-0.0235	0.0185	0.0428	0.0443	0.0228	-0.0175	-0.0585	-0.0429
12/11-20	0.0429	0.0585	0.0175	-0.0228	-0.0443	-0.0428	-0.0185	0.0235	0.0615	0.0333
13/1-10	-0.0076	-0.0180	-0.0019	0.0348	0.0667	0.0690	0.0399	0.0021	-0.0175	-0.0100
13/11-20	0.0100	0.0175	-0.0021	-0.0399	-0.0690	-0.0667	-0.0348	0.0019	0.0180	0.0076
14/1-10	0.0055	0.0208	0.0375	0.0586	0.0843	0.0868	0.0619	0.0399	0.0228	0.0075
14/11-20	-0.0075	-0.0228	-0.0399	-0.0619	-0.0868	-0.0843	-0.0586	-0.0375	-0.0208	-0.0055
15/1-10	0.0113	0.0407	0.0661	0.0849	**0.0963**	**0.0975**	0.0868	0.0690	0.0443	0.0153
15/11-20	-0.0153	-0.0443	-0.0690	-0.0868	**-0.0975**	**-0.0963**	-0.0849	-0.0661	-0.0407	-0.0113
16/1-10	0.0109	0.0393	0.0639	0.0824	**0.0948**	**0.0963**	0.0843	0.0667	0.0428	0.0148
16/11-20	-0.0148	-0.0428	-0.0667	-0.0843	**-0.0963**	**-0.0948**	-0.0824	-0.0639	-0.0393	-0.0109
17/1-10	0.0043	0.0167	0.0323	0.0550	0.0824	0.0849	0.0586	0.0348	0.0185	0.0058
17/11-20	-0.0058	-0.0185	-0.0348	-0.0586	-0.0849	-0.0824	-0.0550	-0.0323	-0.0167	-0.0043
18/1-10	-0.0100	-0.0240	-0.0063	0.0323	0.0639	0.0661	0.0375	-0.0019	-0.0235	-0.0132
18/11-20	0.0132	0.0235	0.0019	-0.0375	-0.0661	-0.0639	-0.0323	0.0063	0.0240	0.0100
19/1-10	-0.0385	-0.0652	-0.0240	0.0167	0.0393	0.0407	0.0208	-0.0180	-0.0615	-0.0491
19/11-20	0.0491	0.0615	0.0180	-0.0208	-0.0407	-0.0393	-0.0167	0.0240	0.0652	0.0385
20/1-10	**-0.0985**	-0.0385	-0.0100	0.0043	0.0109	0.0113	0.0055	-0.0076	-0.0333	**-0.0930**
20/11-20	**0.0930**	0.0333	0.0076	-0.0055	-0.0113	-0.0109	-0.0043	0.0100	0.0385	**0.0985**

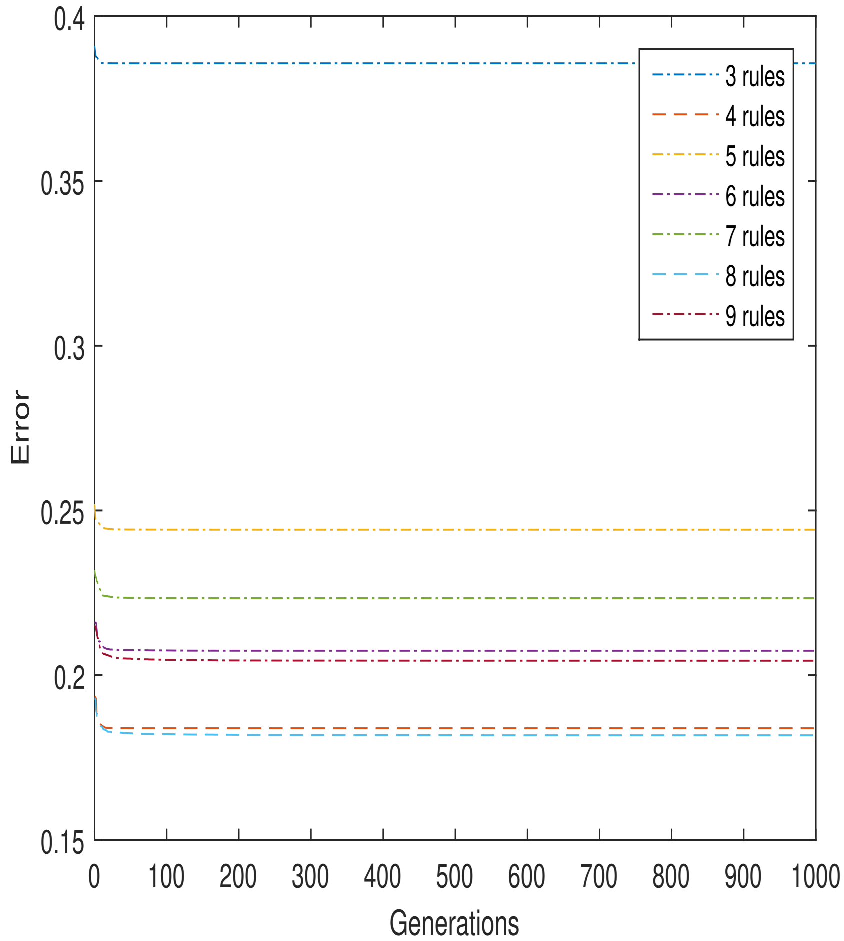

Fig. 4.3 The average error values decrease over time during membership function optimisation for rule bases with 3 to 9 rules

4.2.4 Inference Results and Analysis

By employing the traditional similarity-based method in [58], if the number of rules is smaller than 23, then the sum error of testing results is too high to be discussed. However, the proposed curvature-based method can still generate acceptable results, ie., it can achieve the same or even better performance with the sparse rule base using fewer rules than 23 rules. The important regions can be selected according to the curvature values. The first 23 important rules are listed in Table 4.2. After selecting the most significant m rules, the optimisation algorithm GA is employed to fine-tune the membership functions involved in the rules. In this experiment, the population size was set to 100, the maximum number of generations was set to 1,000, and the probabilities of crossover and mutation were set to 0.8

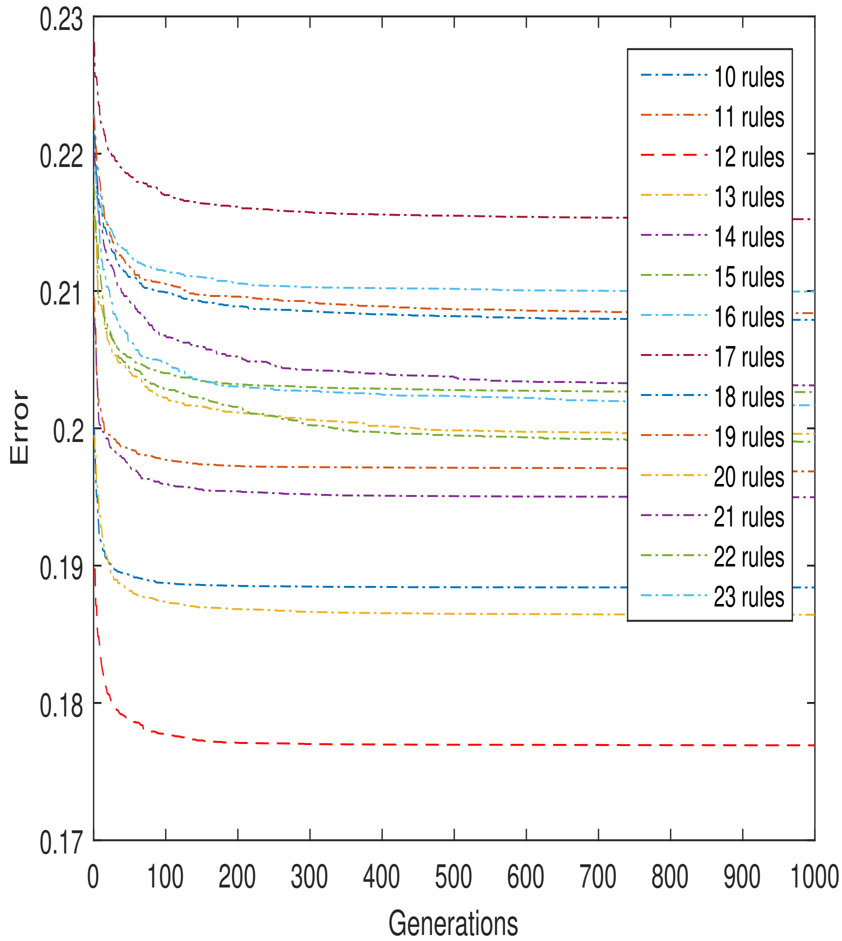

Fig. 4.4 The average error values decrease over time during membership function optimisation for rule bases with 10 to 23 rules

and 0.01, respectively. The optimised rules of a rule base with 23 rules are summarised in Table 4.3.

To enable a comparative study, the sum error from 36 random testing points produced by the proposed method based on different sizes of rule bases is given in Table 4.4. Compared to Fig. 15 of [58], error measure using reduction and different interpolation methods were shown. From this table, it is clear that rule bases with fewer rules generally lead to larger sum error and poorer system performance, while rule bases with more rules generally result in smaller sum error and better performance. However, it should be noted that this is not always the case. For instance, the sum error produced by the rule base with 12 rules is smaller than that produced by 17 rules. In fact, as shown in the table, the rule bases with 12, 10, 8, and 4 rules in this experiment have demonstrated better performance. This is partly because these numbers of rules more efficiently represent the intrinsic characteristics of the data.

Table 4.2 The initialised rule base

	IF		THEN		IF		THEN
i	**x**	**y**	**z**	**i**	**x**	**y**	**z**
1	$(-5,-4.5,-4)$	$(-5,-4.5,-4)$	$(0.881,0.981,1.081)$	13	$(0,0.5,1)$	$(-1,-0.5,0)$	$(-0.125,-0.025,0.075)$
2	$(-5,-4.5,-4)$	$(4,4.5,5)$	$(-1.081,-0.981,-0.881)$	14	$(-1,-0.5,0)$	$(0,0.5,1)$	$(-0.125,-0.025,0.075)$
3	$(4,4.5,5)$	$(-5,-4.5,-4)$	$(-1.081,-0.981,-0.881)$	15	$(0,0.5,1)$	$(0,0.5,1)$	$(-0.075,0.025,0.125)$
4	$(4,4.5,5)$	$(4,4.5,5)$	$(0.881,0.981,1.081)$	16	$(-1,-0.5,0)$	$(-1,-0.5,0)$	$(-0.075,0.025,0.125)$
5	$(4,4.5,5)$	$(5,5.5,6)$	$(0.874,0.974,1.074)$	17	$(5,5.5,6)$	$(-6,-5.5,-5)$	$(-1.068,-0.968,-0.868)$
6	$(4,4.5,5)$	$(-6,-5.5,-5)$	$(-1.074,-0.974,-0.874)$	18	$(-6,-5.5,-5)$	$(-6,-5.5,-5)$	$(0.868,0.968,1.068)$
7	$(-6,-5.5,-5)$	$(4,4.5,5)$	$(-1.074,-0.974,-0.874)$	19	$(-6,-5.5,-5)$	$(5,5.5,6)$	$(-1.068,-0.968,-0.868)$
8	$(-5,-4.5,-4)$	$(4,4.5,5)$	$(0.874,0.974,1.074)$	20	$(5,5.5,6)$	$(5,5.5,6)$	$(0.868,0.968,1.068)$
9	$(5,5.5,6)$	$(4,4.5,5)$	$(0.874,0.974,1.074)$	21	$(-5,-4.5,-4)$	$(-4,-3.5,-3)$	$(0.789,0.889,0.989)$
10	$(-5,-4.5,-4)$	$(5,5.5,6)$	$(-1.074,-0.974,-0.874)$	22	$(4,4.5,5)$	$(-4,-3.5,-3)$	$(-0.989,-0.889,-0.789)$
11	$(5,5.5,6)$	$(-5,-4.5,-4)$	$(-1.074,-0.974,-0.874)$	23	$(-4,-3.5,-3)$	$(4,4.5,5)$	$(-0.989,-0.889,-0.789)$
12	$(-6,-5.5,-5)$	$(-5,-4.5,-4)$	$(0.874,0.974,1.074)$				

Table 4.3 The optimised rule base

	IF		THEN		IF		THEN
i	**x**	**y**	**z**	**i**	**x**	**y**	**z**
1	$(-4.999,-4.5,-4.001)$	$(-4.999,-4.5,-4.001)$	$(0.962,0.981,1)$	13	$(-2.192,0.5,3.192)$	$(-2.456,-0.5,1.456)$	$(-0.303,-0.025,0.253)$
2	$(-4.999,-4.5,-4.001)$	$(3.998,4.5,5.002)$	$(-1,-0.981,-0.962)$	14	$(-2.149,-0.5,1.149)$	$(-2.275,0.5,3.275)$	$(-0.273,0.025,0.323)$
3	$(4.483,4.5,4.517)$	$(-4.504,-4.5,-4.496)$	$(-1,-0.981,-0.962)$	15	$(-1.22,0.5,2.22)$	$(-2.352,0.5,3.352)$	$(0.003,0.025,0.047)$
4	$(3.992,4.5,5.008)$	$(4.002,4.5,4.998)$	$(0.962,0.981,1)$	16	$(-2.475,-0.5,1.475)$	$(-2.191,-0.5,1.191)$	$(-0.075,0.025,0.125)$
5	$(2.006,4.5,6.994)$	$(2.986,5.5,8.014)$	$(0.948,0.974,1)$	17	$(5.433,5.5,5.567)$	$(-5.628,-5.5,-5.372)$	$(-0.969,-0.968,-0.967)$
6	$(4.005,4.5,4.995)$	$(-6.999,-5.5,-4.001)$	$(-1,-0.974,-0.948)$	18	$(-5.998,-5.5,-5.002)$	$(-6.002,-5.5,-4.998)$	$(0.936,0.968,1)$
7	$(-6.008,-5.5,-4.992)$	$(4.003,4.5,4.997)$	$(-0.983,-0.974,-0.965)$	19	$(-5.528,-5.5,-5.472)$	$(5.475,5.5,5.525)$	$(-1,-0.968,-0.936)$
8	$(-4.964,-4.5,-4.036)$	$(-5.998,-5.5,-5.002)$	$(0.971,0.974,0.977)$	20	$(4.989,5.5,6.011)$	$(4.991,5.5,6.009)$	$(0.936,0.968,1)$
9	$(3.004,5.5,7.996)$	$(4.001,4.5,4.999)$	$(0.948,0.974,1)$	21	$(-7.002,-4.5,-1.998)$	$(-3.997,-3.5,-3.003)$	$(0.886,0.889,0.892)$
10	$(-4.997,-4.5,-4.003)$	$(4.997,5.5,6.003)$	$(-1,-0.974,-0.948)$	22	$(3.001,4.5,5.999)$	$(-4.001,-3.5,-2.999)$	$(-1,-0.889,-0.778)$
11	$(5.004,5.5,5.996)$	$(-4.995,-4.5,-4.005)$	$(0.948,0.974,1)$	23	$(-3.999,-3.5,-3.001)$	$(2.997,4.5,6.003)$	$(-1,-0.889,-0.778)$
12	$(-5.514,-5.5,-5.486)$	$(-4.593,-4.5,-4.407)$	$(-0.243,-0.025,0.193)$				

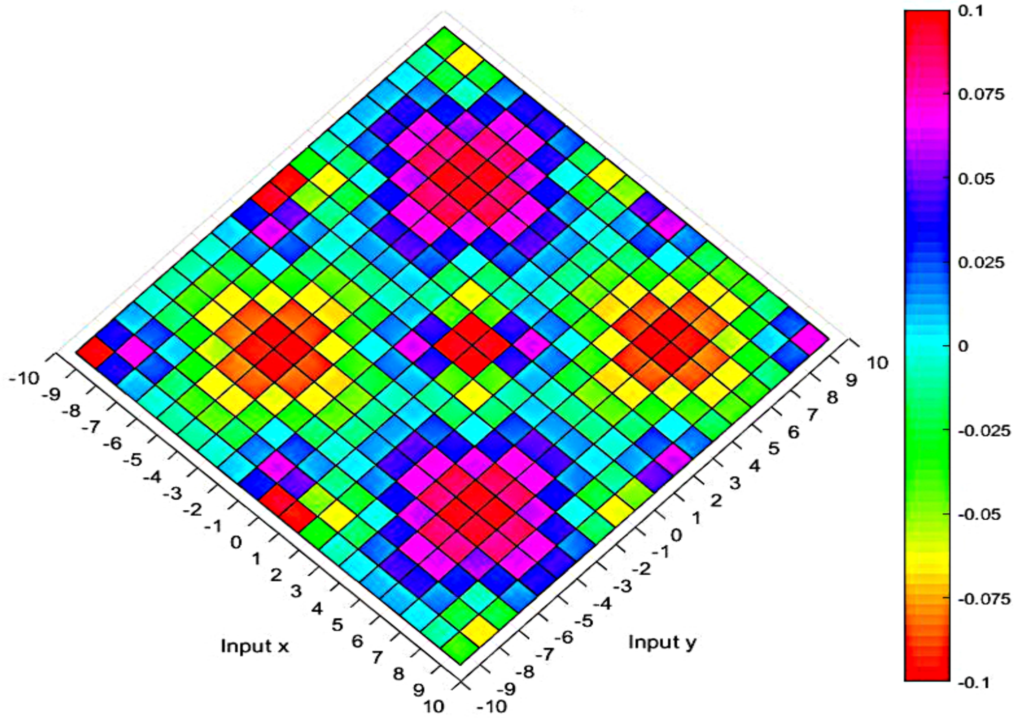

Fig. 4.5 Curvature values of the sub-regions

Table 4.4 The sum error from 36 random testing points based on different sizes of rule bases

Number of rules	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
Sum error	6.1080	6.1275	6.2183	5.9880	6.4906	6.4186	6.8214	6.3904	6.2331	6.0030	5.9036	5.6788	6.4905	5.7084	6.2782	5.8516	7.5389	7.0080	7.9842	5.8159	13.3417

4.3 Indoor Environment Localisation

4.3.1 Application Description

Detecting users in an indoor environment based on the strength of Wi-Fi signals has a wide application domain. Deployable models have been developed in monitoring and tracking users based on the Wi-Fi signal strength of their personal devices. The applications of such models include locating users in smart-home systems, locating criminals in bounded regions, and obtaining the count of users on an access point etc. An indoor environment localisation dataset was employed in this application to validate and evaluate the proposed approach [253]. The dataset was collected in an indoor space by observing the signal strengths of seven Wi-Fi signals visible on a smart-phone. The dataset includes 2,000 instances, each with seven inputs and one output. Each input attribute is a Wi-Fi signal strength observed on the smart-phone, while the output decision class is one of the four locations. The sample data for user localisation using wireless are listed in Table 4.5.

Table 4.5 Sample Data for user localisation using wireless signal strength

input 1	input 2	input 3	input 4	input 5	input 6	input 7	output
64	56	61	66	71	82	81	1
68	57	61	65	71	85	85	1
17	66	61	37	68	75	77	2
16	70	58	14	73	71	80	2
52	48	56	53	62	78	81	3
49	55	51	49	63	81	73	3
58	52	58	60	51	87	88	4
62	59	52	69	46	91	98	4

4.3.2 Curvature Calculation

In comparison to the synthetic dataset (where the required data can be obtained from anywhere in the space, thus resulting in a very dense dataset), the collected small dataset is irregular and contains discrete data. In this case, due to the sparseness of the dataset, each data instance in the dataset is regarded as a high-dimensional hypercube and is represented as a fuzzy rule if it is selected. In order to distinguish the data instances and to select the important instances, all high-dimensional hypercubes are broken down into $C_7^2 = \frac{7!}{2!(7-2)!} = 21$ cubes. Following this, the curvature values of all these 21 cubes from each hypercube are calculated, using the mean curvature from the moving least-squares (MLS) surface method as introduced in Section 3.1. The Gaussian scale parameter h in Eq. 2.38 (which determines the width of the Gaussian kernel) is set to 0.35, and the number-of-neighbours parameter is set to 38. For simplicity, only the curvature values of the decomposed 21 cubes from the first instance are listed in Table 4.6, while all the other 1999 instances are omitted here due to the space limit. Following this, the accumulated virtual ‘curvature value’ of each hypercube was calculated, being as the summation of all its corresponding decomposed 21 cubes.

4.3.3 Rule Base Generation

Note that in the rule base generation process for classification problems, each output class should be covered in order to avoid misclassification. Therefore, when selecting instances or hypercubes, the instances should be considered in tandem with the local higher curvature values, instead of the global higher ones. Otherwise, the instances with higher curvature values may just belong to one or two output classes in this particular application. This dataset has 500 instances in each output class, with a total of 2,000 instances in all four output classes. Thus, in each class, m important instances are chosen to guarantee that all the output classes are covered. In other situations that involve an uneven distribution, the number of selected instances in each class can be adjusted accordingly. Based on their accumulated curvature values, for each output class, the most significant m instances or hypercubes within the output class were selected, simply by taking the first m hypercubes with the highest accumulated curvature values. Finally, the most important $4*m$ hypercubes or instances were selected to form the final raw rule base. The raw rule base is optimised by GA to fine-tune the membership functions involved in the rules.

4.3.4 Inference Results and Analysis

Empirical study shows that, in this example, $m = 7$ produces the best performance. The result shows that although there are 2,000 instances, only $4*7 = 28$ important instances were needed to construct a sparse rule base, as summarised in Table 4.7. The initial rule base was optimised by applying the GA over the training dataset in fine-tuning the membership functions of the fuzzy sets which are involved in the 28 rules. The results produced by the proposed method using two to seven features and other approaches using all seven features were compared, as shown in Table 4.8. The classification rate of the proposed method is 99.25%, which outperforms all the existing methods. Furthermore, using the FRI performance of the constructed intermediate rule bases, the important features can also be identified; these appear in descending order as [5, 1, 4, 7, 6, 3, 2]. If all the seven input features are used, the accuracy is 99.25%. Instead, if only the most important two to six input features are used, the accuracy still remains 96.75%, 98.15%, 98.6%, 98.8%, and 99.15%, respectively. This clearly demonstrates the superior advantage of the proposed approach.

Table 4.6 Curvature values of the decomposed cubes from the first instance

Cube Index	1	2	3	4	5	6	7	8	9	10	11
Curvature value	0.0494	0.0149	0.1472	0.0069	0.1189	0.0204	0.0496	0.0568	0.2812	0.2004	0.0558
Cube Index	12	13	14	15	16	17	18	19	20	21	
Curvature value	0.0145	0.0305	0.0247	0.0732	0.0734	0.1314	0.0046	0.049	0.0264	0.0211	

Table 4.7 Data of the selected 28 instances

input 1	input 2	input 3	input 4	input 5	input 6	input 7	output	index of instance
63	59	60	65	69	81	84	1	384
59	57	59	64	73	79	84	1	261
60	53	60	62	73	81	82	1	162
63	59	57	65	69	80	86	1	361
60	59	61	62	68	81	85	1	300
60	56	56	63	65	80	83	1	334
63	57	63	64	67	81	83	1	377
48	58	58	44	71	77	79	2	644
43	56	59	37	64	74	77	2	885
44	57	53	46	67	78	79	2	678
41	55	53	37	64	79	76	2	827
50	55	58	43	73	75	80	2	645
42	54	56	39	63	76	78	2	904
38	55	54	42	63	78	71	2	547
45	56	55	46	68	79	78	3	446
48	54	54	49	67	77	89	3	1187
48	57	49	53	62	79	87	3	1203
45	54	54	48	63	78	82	3	1028
51	57	54	55	62	87	81	3	1462
51	52	49	50	63	79	79	3	1090
51	52	52	56	68	79	87	3	1175
58	57	55	66	51	88	87	4	1734
56	53	51	59	50	84	84	4	1964
57	56	49	58	50	87	85	4	1810
64	54	52	58	52	89	88	4	1694
59	51	57	59	52	87	86	4	1545
58	51	56	58	50	88	88	4	1542
66	56	56	66	49	89	87	4	1635

Table 4.8 Results for comparison in application 1

Previous Methods	PSO-NN	GSA-NN	PSOGSA-NN	FPSOGSA-NN	SVM	NAIVE BAYES	**Proposed Approach**
Accuracy (%)	64.66	77.53	83.28	95.16	92.68	90.47	**99.2**

Table 4.9 Results for comparison in application 1

Proposed Approach	7 features	6 features	5 features	4 features	3 features	2 features
Accuracy (%)	99.2	99.15	98.8	98.6	98.15	96.75

4.4 Student Knowledge Level Evaluation

4.4.1 Application Description

This application is illustrated by adopting the problem considered in [254]. The behaviours or dynamic data of students are obtained from their interactions with web-environment by the user modeling system. Such obtained data are used as input data to create and to update user knowledge model, thus to customise subjects according to their knowledge. One key difficulty in user modeling system design is to efficiently and accurately utilise the stored data then classify the students' knowledge level. A random subset with 258 observations of the available data set was selected to generate the training set. A random subset with 145 observations of the available data set was selected to generate the validation set, each observation or instance has the 5 inputs (crisp values) and 1 output (linguistic term) below.
Inputs:
STG (The degree of study time for goal object materials)
SCG (The degree of repetition number of user for goal object materials)
STR (The degree of study time of user for related objects with goal object)
LPR (The exam performance of user for related objects with goal object)
PEG (The exam performance of user for goal objects)
Output:
UNS (The knowledge level of user)

The proposed method is applied to test the testing data then classify the students' knowledge level. As the output results are four linguistic terms of the knowledge level classes, which are 'very_low', 'low', 'middle', 'high', firstly some data pre-processing need to be done to translate the linguistic terms into some crisp values, such as 1,2,3,4 respectively. The first 10 instances of the 258 training observations are listed in Table 4.10 and 4.11. Then the proposed methods can be used to calculate the curvature values and help select features and instances.

Table 4.10 Linguistic Sample data for student knowledge level evaluation

input 1 (STG)	input 2 (SCG)	input 3 (STR)	input 4 (LPR)	input 5 (PEG)	output (UNS)	index of instance
0	0	0	0	0	VeryLow	1
0.08	0.08	0.1	0.24	0.9	High	2
0.06	0.06	0.05	0.25	0.33	Low	3
0.1	0.1	0.15	0.65	0.3	Middle	4
0.08	0.08	0.08	0.98	0.24	Low	5
0.09	0.15	0.4	0.1	0.66	Middle	6
0.1	0.1	0.43	0.29	0.56	Middle	7
0.15	0.02	0.34	0.72	0.25	VeryLow	8
0.2	0.14	0.35	0.72	0.25	Low	9
0	0	0.5	0.2	0.85	High	10

Table 4.11 Digital Sample data for student knowledge level evaluation

input 1 (STG)	input 2 (SCG)	input 3 (STR)	input 4 (LPR)	input 5 (PEG)	output (UNS)	index of instance
0	0	0	0	0	1	1
0.08	0.08	0.1	0.24	0.9	4	2
0.06	0.06	0.05	0.25	0.33	2	3
0.1	0.1	0.15	0.65	0.3	3	4
0.08	0.08	0.08	0.98	0.24	2	5
0.09	0.15	0.4	0.1	0.66	3	6
0.1	0.1	0.43	0.29	0.56	3	7
0.15	0.02	0.34	0.72	0.25	1	8
0.2	0.14	0.35	0.72	0.25	2	9
0	0	0.5	0.2	0.85	4	10

4.4.2 Curvature Calculation

As elaborated in the 3.3.2, the whole domain space now is easily broken down into $C_5^2 = \frac{5!}{2!(5-2)!} = 10$ Basic-3D-Spaces, with the total number as the combination of any 2 input variables out from 5 input variables. The moving least-squares (MLS) surface method is used to calculate the curvature values. The Gaussian scale parameter h which determines the width of the Gaussian kernel in the Eq. 2.38 is set to 0.382, and the number of neighbours parameter is set to 28, which is about 1/10 of the total number of training samples. The mean curvature in the Eq. 2.42 is calculated, and the curvatures values for all instances in all the

Table 4.12 The performance in each basic case

Index of basic case	1	2	3	4	5	6	7	8	9	10
Misclassified in total 258	156	157	132	88	160	151	63	135	85	23

Basic-3D-Spaces form the below matrix with 258 rows and 10 columns:

$$C = \begin{pmatrix} 0.173 & 0.304 & 0.188 & 0.136 & 0.208 & 0.171 & 0.087 & 0.266 & 0.143 & 0.092 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 0.078 & 0.147 & 0.114 & 0.073 & 0.149 & 0.112 & 0.073 & 0.198 & 0.138 & 0.096 \\ 0.069 & 0.130 & 0.078 & 0.073 & 0.088 & 0.124 & 0.074 & 0.139 & 0.156 & 0.095 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \end{pmatrix}, \tag{4.1}$$

where 258 is the total number of training instances and 10 is the total number of divided Basic-3D-Spaces. Each row of the matrix represents the curvatures values of the indexed training instance in all the divided Basic-3D-Space, and reveals the importance of that instance (with its two corresponding input features of the indexed basic case) for the conclusion.

4.4.3 Rule Base Generation

According to the curvature values above, in each basic case, the instances with high curvature values are selected and a sparse rule base for FRI can be generated, using the two corresponding input features and one output conclusion of the initial instances to construct the fuzzy rules. Then each generated sparse rule base in the divided basic case is applied to calculate results of total 258 instances in the training data, using their corresponding two input features only. By comparing all the produced results with the original output conclusion, the best performance can be found. As shown in the Table 4.12, the misclassified numbers of total 258 instances in all divided basic cases are list below, and the performance in the 10th basic case is the best one. With the index of basic case which has the best performance, which is 10, its corresponding two features can be easily found as the most important pairwise features, which are 4th input and 5th input (LPR and PEG). Then with these two important pairwise features and the initial instances, a sparse rule base for FRI is generated. The empirical study shows that although there are 258 instances, only $4*6=24$ important instances were needed to construct a sparse rule base, as summarised in Table 4.13. It is applied to calculate results of instances in the testing data, using their corresponding 4th and 5th input features only. Finally using this sparse rule base for FRI, there are 6 misclassified results by Stabilised-KH method in total 145 testing observations, that is, 95.9% accuracy.

Table 4.13 Data of the selected 24 instances

input 1 (STG)	input 2 (SCG)	input 3 (STR)	input 4 (LPR)	input 5 (PEG)	output (UNS)	index of instance
0	0	0	0	0	1	1
0.05	0.07	0.7	0.01	0.05	1	17
0.265	0.6	0.28	0.66	0.07	1	117
0.25	0.1	0.03	0.09	0.15	1	66
0.6	0.19	0.55	0.08	0.1	1	203
0.32	0.2	0.06	0.26	0.24	1	67
0.06	0.06	0.05	0.25	0.33	2	3
0.08	0.08	0.08	0.98	0.24	2	5
0.06	0.06	0.51	0.41	0.3	2	12
0.2	0.14	0.35	0.72	0.25	2	9
0.15	0.32	0.05	0.27	0.29	2	19
0.12	0.28	0.2	0.78	0.2	2	21
0.1	0.1	0.43	0.29	0.56	3	7
0.1	0.1	0.15	0.65	0.3	3	4
0.09	0.15	0.4	0.1	0.66	3	6
0.2	0.29	0.25	0.49	0.56	3	20
0.2	0.2	0.7	0.3	0.6	3	15
0.1	0.27	0.31	0.29	0.65	3	23
0.08	0.08	0.1	0.24	0.9	4	2
0.18	0.18	0.55	0.3	0.81	4	11
0.12	0.12	0.75	0.35	0.8	4	16
0.1	0.1	0.7	0.15	0.9	4	14
0	0	0.5	0.2	0.85	4	10
0.15	0.275	0.8	0.21	0.81	4	30

4.4.4 Inference Results and Analysis

The results led by proposed method and by other approaches in the reference list are as shown in the Table 4.14. The abbreviations of the algorithms are Decision Tree (J48), Random Forest (RF), Support Vector Machine (SVM), Simple Logistic (SL), Multi Layer Perception Neural Network (MLP), K-Nearest-Neighbour (KNN). The EU, MA and MI represent the Euclidian Distance, Manhattan Distance, Minkowski Distance respectively. The results shows the proposed method can only use several important instances with their important few features to achieve competitive performance (accuracy 95.9%). Here both the training

Table 4.14 Results for Comparison in Application 2

Items	**Proposed Approach**	**Bayes**	**J48**	**RF**	**SVM, SL, MLP**	**KNN**			**Approach in [254]**		
						EU	MA	MI	EU	MA	MI
Misclassified	**6**	38	12.2	5.66	2	26.2	21.7	30.5	3	3	5
Error Rates	**4.1%**	26.2%	8.4%	3.9%	1.4%	18.1%	15%	21%	2.1%	2.1%	3.5%
Accuracy	**95.9%**	73.8%	91.6%	96.1%	98.6%	81.9%	85%	79%	97.9%	97.9%	96.5%

and testing instances are not evenly distributed with four output classes, and the output are even in linguistic expert knowledge (Very Low: 50, Low: 129, Middle: 122, High 130). This experiment shows the proposed work can also work for linguistic environment and can provide an objective way to distinguish what is important.

4.5 Zero-Shot Learning Image Classification

4.5.1 Application Description

Existing image classification techniques highly rely on supervised models that are trained on large-scale datasets. Despite improved ontology engineering, such as ImageNet that includes 20k+ daily categories, the scale is far behind the requirement of generic image recognition. First of all, semantic concepts are complex and structured whereas the label space for most of current supervised learning consists of discrete and disjoint one-hot category vectors. The associations between classes are imposed to be neglected. Secondly, the dimension of label space is ever-growing. For example, on average, 1,000 new entries are added to Oxford Dictionaries Online every year. Consequently, for the scalability of conventional supervised learning is limited due to expensive acquisition of high-quality training images with annotations.

In the past decade, Zero-shot learning (ZSL) was proposed as a potential solution which aims to transfer a learnt supervised model to unseen classes without acquiring new training data at the test time. The essential problem is how to teach the machine what visual features will present in the test class using prior human knowledge. Therefore, the representation of human knowledge is required to maximally bridge the visual-semantic gap. Most of existing approaches adopt visual attributes [255] so that a discrete class label can be embedded by a boolean representation, each dimension of which denotes whether an attribute present or absent. In this way, visual-attribute model from seen classes can be shared to unseen ones with using pre-defined attribute embeddings.

Although the generalisation to new classes can circumvent training image collection, constructing an attribute-based ontology is even more costly. As shown in Fig.4.7 (B), both seen and unseen classes need to be annotated by tens or hundreds of attributes. For example, the most popular benchmark, AwA, requires the annotator to give 85 attributes for each of 50 classes, let alone instance-level datasets, such as aPY and SUN which contain hundreds of thousands of manual annotations. Such restrictions severely prevent ZSL from being widely applied to many non-attribute scenarios. Furthermore, designing attributes is an ambiguous work since most of visual features are intangible. Constructing a large-scale ontology with attributes is thus time-consuming and error-prone.

An efficient simile-based ZSL (zero-shot learning) framework was proposed and it can recognise unseen classes with light-weight simile annotations. It has been elaborated in [31], with the key idea of curvature based FRI as below. Using the simile vectors from light-weight annotations, ZSL problems can be described as:

If a Leopard is *'to* A_1 *extent like a Bobcat'* and *'to* A_2 *extent like a Tiger'*, then it is *'to B extent like a Lion'*,

the task is to infer *'to what B extent a Leopard is like a Lion'*, as shown as in Fig. 4.6. Note that A_1, A_2, B are known as the representative values of the fuzzy sets. The above form is just as same as the Eq. 3.1, therefore it can be easily solved by the proposed curvature-based sparse rule base generation method. By regarding both input and output as fuzzy variables, the proposed method significantly boosted the ZSL performance by accurately predicting the similarity value of each seen class in the simile vector using only discrete simile annotations.

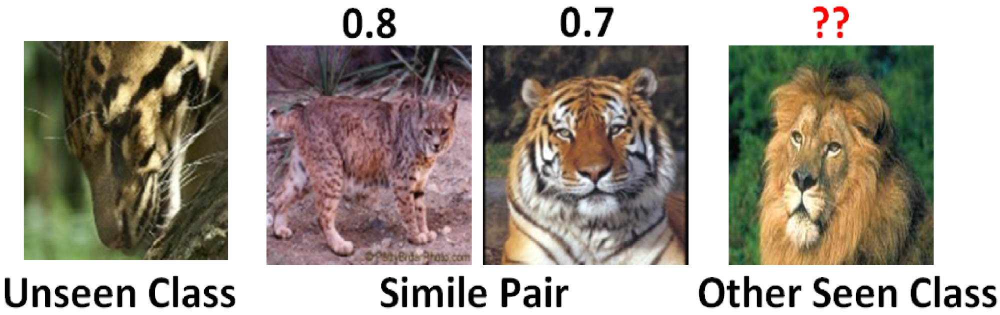

Fig. 4.6 The key idea of curvature-based FRI in application 4.

In this work, investigation on how to spend the minimal annotation cost while still retain the high performance of that using attributes are implemented. The key idea is inspired by an intuitive fact. To describe an unseen instance, the most straightforward way is to relate it to previously seen classes. Such expressions are called *Similes* e.g. facial similarity[256] or *Classemes* [257] which explicitly compare two things by connecting words, *like, as, as,*. Simile refers to a part of speech which is proposed to describe complex visual features, and Classemes can describe either objects similar to, or objects seen in conjunction with an unseen class, i.e. class-to-class-similarities. However, existing methods often involve expensive class-to-class annotations [258], which is no difference to that of using attributes in terms of annotation cost. And the annotation cost, as shown in Fig.4.7 (C), is not less than that using attributes. Fuzzy Interpolative Reasoning leverages a few similes to infer the full

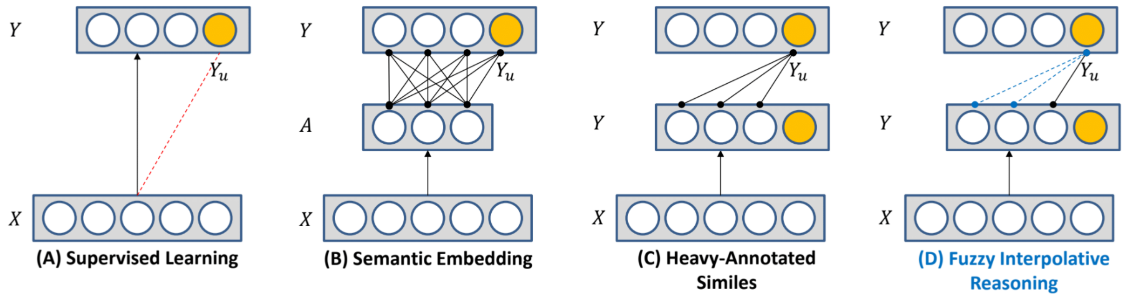

Fig. 4.7 Zero-shot Learning framework comparison.

associations of an unseen class to all of seen classes. Solid lines and arrows denote required annotations or associations. Furthermore, most of existing approaches fail to quantify the simile between each pair of seen and unseen classes.

Therefore, despite a large-scale simile-based ontology [259], such a stream of approaches have not gain much attention until some recent work [260]. Contrary to these methods, the proposed work utilise word embeddings as clues to find some initial similes to further minimise the human intervention. Due to the non-visual similarities of word embeddings, the human intervention is required to adjust the rank based on visual similarities and select a number of top similes. Despite light-weight annotations, the inferred representation through FIR can significantly boost existing ZSL methods by substituting their used attributes.

4.5.2 Simile Construction

Kernelised Image Representation Kernel techniques are widely adopted to increase the generalisation ability of a classifier so as to cope with various conditional changing. Kernel-based learning algorithms, such as SVM (Support Vector Machine), Kernel PCA (Principal Component Analysis), or Gaussian Processes, have been successfully employed on many fundamental problems, such as classification, regression, density estimation and clustering [261]. The methods can be roughly divided into using Mercer and none-Mercer kernels (whether positive-definite or not), which mainly concerns the convergence when being applied to classifiers, such as SVM. The spirit of image-to-class kernel [262] is shared in this work. However, the aim of this work is based on an embedding aspect so as to bridge the visual-simile gap for ZSL rather than purely dealing with outliers or noise from the same classes of training samples. Our representation is also similar to VLAD related approach [263]. But, each dimension of the proposed SMS has an explicit class name instead of implicit anchor points.

Knowledge Representation

Similes provides an effective way to qualitatively describe visual similarities between images. Compared to attributes or texts, similes are more visual-related and do not involve extra concepts, thus lead to less information loss. To represent an image x using similarities, the most straightforward approach is to estimate the likelihood $p(x_c|x_n)$ between this image x and each image in a seen class in the visual space, $x_c \in \mathscr{X}_c$. In this work, the Parzen likelihood estimation is simply adopted to compare a pair of images:

$$p(x_c|x) = K(x_c - x) = \exp(-\frac{1}{2\sigma^2}||x_c - x||_2^2), \tag{4.2}$$

where $K(.)$ is the Parzen match kernel function under a typical Gaussian distribution leading to a non-negative integrated value; $||.||_2$ represents the ℓ_2-norm distance between two vectors. Using the above equation, a similarity match score (SMS) $p(\mathscr{X}_c|x)$ can be concluded by averaging the likelihoods to all images in class c. However, most of the likelihood values are negligible as the visual space is high-dimensional and the likelihood values are K exponentially decreasing with the distance. Moreover, the training set might be noisy, and thus a conclusion based on all of the samples may not lead to the best result. Therefore, the top k nearest points of x_n, ie. $\{x^1_{NN_c}, ..., x^k_{NN_c}\} \in \mathscr{X}_c$, can be used to make an improved estimation:

$$p(\mathscr{X}_c|x_n) = \frac{1}{k}\sum_{i=1}^{k} \exp(-\frac{1}{2\sigma^2}||x^i_{NN_c} - x_n||_2^2). \tag{4.3}$$

From this, each image can be represented as a vector of SMS (vSMS) in reference to the known classes, or referred to as a SV (Simile Vector). This simple yet effective vSMS generation approach is referred to as match kernel embedding (MKE). Each dimension of the generated vSMS stands for the likelihood value or similarity to the corresponding class. Since the sum of vSMS is normalised to one, different values of σ do not make significant difference. For simplicity, we set $\sigma = 1$. Formally, the MKE approach is defined as follows:

$$f_1(x_n) = [p(\mathscr{X}_1|x_n), ..., p(\mathscr{X}_C|x_n)] = v_n \in \mathscr{V}. \tag{4.4}$$

The vSMS not only quantitatively expresses similes representing the visual-semantic relationship, but also effectively discriminates the embedding space to better support ZSL, as illustrated in Fig. 4.8. FIR is performed to infer the centroid of each class as the prototype.

Simile Quantification For each unseen class, we have $[c_1, ..., c_k]$ similes. As argued earlier that it is difficult to give a specific value of the similarity and simile annotations which suffer from subjective variances. In this work, we propose a novel empirical alternative that use the values in the rule base for initial quantification of discrete similes. Specifically, an unseen

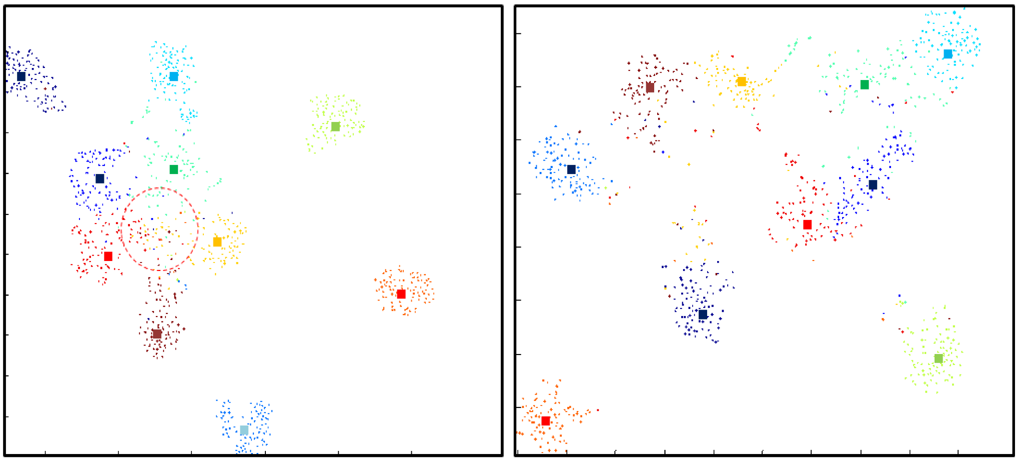

Fig. 4.8 (A) Raw visual feature distribution of the 10 unseen classes in AwA. (B) After MKE, non-discriminative points (red circle in (A)) are separated.

class has a simile of seen class $c_i \in [c_1,...,c_k]$, which can use the averaged self similarities of instances in c_i to make an approximation:

$$v'_{c_i} = \frac{1}{|c_i|} \sum_{v_i \in \mathscr{V}_{c_i}} v_i, \tag{4.5}$$

where $|.|$ is the cardinality of a class c_i; $\mathscr{V}_{c_i}$ denotes self-similarity values of all instances in class c_i: $\phi_{c_i}(\mathbf{x})$. In this way, we can in turn calculate the initial similarity values of the give similes $[c_1,...,c_k] \rightarrow [v'_{c_1},...,v'_{c_k}]$. Next, we elaborate how to select proper observations in the rule base to complete the FIR algorithm: $f_{2_c}(v'_{c_1},...,v'_{c_k}) = v_c$ for C times to achieve a full simile vector $\mathbf{v} = [v_1,...,v_c,...v_C]$. Note that the initialised v's are also updated so as to further mitigate the annotation bias.

As shown in Fig. 4.9, we initialise v'_1 and v'_2 by averaging the SMS in the simile classes $s^{(1)}$ and $s^{(2)}$:

$$v'_1 = \frac{1}{|c_1|} \sum_{y_n = c_1} v_{nc_1},\; v'_2 = \frac{1}{|c_2|} \sum_{y_n = c_2} v_{nc_2}, \tag{4.6}$$

where $c_1 = s^{(1)}$ and $c_2 = s^{(2)}$; $|.|$ is the cardinality of a class. The remaining task is to infer a prototype v^* of the unseen class: $f_2(s^{(1)},s^{(2)}) : [v^*_1, v^*_2,...,v^*_C] = v^* \in \mathscr{V}$. Note that the initialised v'_1 and v'_2 are also updated. The f_2 is achieved by sequentially inferring the SMS to each seen class using the following procedures. The similarity-based histogram is used as a new image representation. The SMS on the training set is computed solely based on visual similarities. Such representations are intuitive and discriminative so that we do not rely on attributes anymore for ZSL. Fuzzy inferences are repeated on every seen class to obtain the

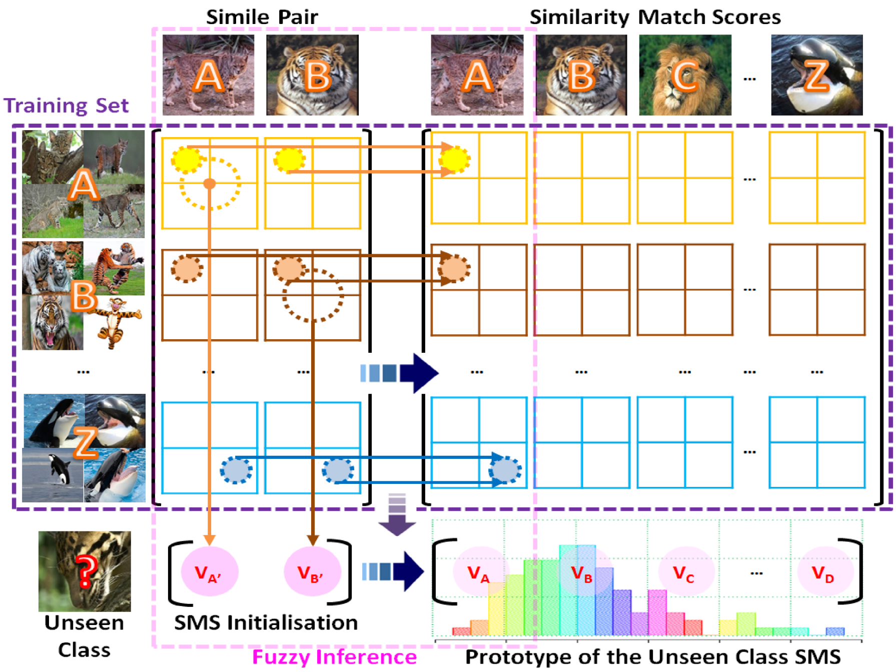

Fig. 4.9 The framework of fuzzy inference.

overall vSMS representation as a prototype.

Similarity Measure

A fuzzy membership function is defined as a measurable map:

$$f_2 := M(v) \rightarrow [0,1] \tag{4.7}$$

For example, given *leopard* is 0.9 similar to *bobcat*, the function can infer how much it is similar to *tiger* based on a rule base of observed membership, *i.e.* shared similarity between *bobcat* and *tiger*. Different from the probability theory that model predictions exclusively at a time, fuzzy process focus more on how much information in common at the same time. In this way, sparse values can likely get shared members to smooth the variance. In this case, we hope to maximise the tolerance to annotation errors, such as bad ranks or missing important similes due to the visual-semantic discrepancy.

In this work, the simplest Gaussian kernel is used as the membership function. Given a $\mathbf{x}$, K nearest neighbours $[\mathbf{x}_c^1, ..., \mathbf{x}_c^K]$ in class c are selected to estimated the similarity between $\mathbf{x}$ and class c:

$$\phi_c(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K} \exp(-\frac{1}{2\sigma^2}\|\mathbf{x}_c^i - \mathbf{x}\|_2^2). \tag{4.8}$$

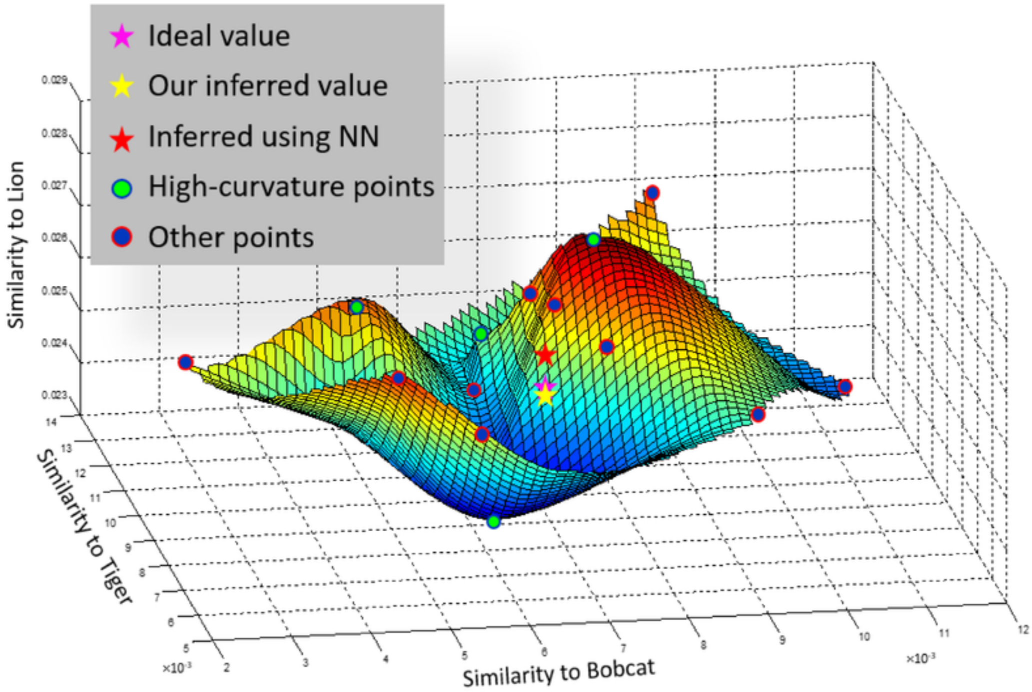

Fig. 4.10 An demonstration of sparse fuzzy rule selection.

where $\sigma = 1$ without loss of generality. The similarity value is normalised to [0,1] using a sigmoid function. In this way, we can convert the whole training set $\mathscr{X}$ into the simile space $\mathscr{V}$ as the fuzzy rule base using $\mathscr{V} = [\phi_1(\mathscr{X}), \ldots, \phi_C(\mathscr{X})]$.

4.5.3 Rule Base Generation

Fuzzy Rule Selection

Like the membership estimation, it is not necessary to use all of the training instances as fuzzy rules. Only rules that can make prominent effects to the conclusion are required. In this work, we adopt a sparse manner to refine the fuzzy rule base. Firstly, the local regions surrounding the observation instead of the global ones are used. The process is implemented by searching Q nearest neighbours of $[v'_{c_1}, \ldots, v'_{c_k}]$ in the c_k dimensional rule base. Afterwards, the *profile curvature* of the local region is constructed to represent the extent to which the local region deviates from being ‘flat’ or ‘straight’. By viewing the pattern to be modelled as a geometry object, as shown in Fig.4.10. The ideal value is computed by averaging the Leopard-Lion SMSs using real visual data. Using high-curvature points, the inferred value is more accurate than that using nearest neighbours.

The proposed method which firstly discussed in [264] is employed to select the points with the steepest downward gradient for a given direction. Let the surface of the local region be denoted as $f(v_1,v_2)$, the gradient can be expressed as a 2-*D* vector field $\nabla f = (f_{v_1}, f_{v_2}, 0) = f_{v_1}^{(i)} + f_{v_2}^{(j)}$, where i and j are steps. The *slope* is defined as a scalar field just like Eq. 2.35:

$$S(v_1,v_2) = |\nabla f| = \sqrt{f_{v_1}^2 + f_{v_2}^2} \tag{4.9}$$

Using the S, the corresponding unit vector u is $u = (-\nabla f/S)$. For a given scalar field $F(v_1,v_2)$ the *directional derivative* D_u on the direction u and the overall profile curvature value K_v can be calculated:

$$D_u(F) = \nabla F \cdot u \tag{4.10}$$

$$K_v = -S^{-2}(f_{v_1}^2 f_{v_1v_2} + 2f_{v_1}f_{v_2}f_{v_1v_2} + f_{v_2}^2 f_{v_2v_2}), \tag{4.11}$$

where D_u compute the changing rate at F given a movement u; and K_v can be either positive or negative which corresponds to the convexity and the concavity respectively. For simplicity, we use *longitudinal profile curvature* that is a streamline passing through F(x,y). Firstly, we calculate eight directional derivatives for each point (clockwise from North to Northwest) which corresponds to the cardinal and inter-cardinal directions: $\{D_{u_1},...,D_{u_8}\}$. The point on each direction is interpolated using the *v4* function of Matlab toolbox *griddata* with parameter u as the density unit. The longitudinal profile curvature K_u can be calculated by comparing the pair of directional derivatives D_u and D'_u on the opposite directions ($D_u > D'_u$):

$$K_u = \frac{D_u - D'_u}{S^2} \tag{4.12}$$

Now the overall rule base $\mathscr{V}$ is refined into only top R points $[\mathbf{v}_1,...,\mathbf{v}_R]$ with the highest K_u values. In this work, we propose a novel multi-dimensional Gaussian membership function that can simultaneously accounts the R points, as shown in Fig.4.11 (R=5 here). For each of the r selected high-curvature points $\mathbf{v}_r$, the fuzzy set is constructed using its T nearest neighbours $[\mathbf{v}_1,...,\mathbf{v}_T]$ from the overall rule base. The membership function for each r point is:

$$M(\mathbf{v}_r) = \exp\left(-\frac{(\mathbf{v}_r - mu_r)^2}{2\Sigma_r^2}\right), \tag{4.13}$$

where mu_r and Σ_r are the mean and covariance of $[\mathbf{v}_1,...,\mathbf{v}_T]$. Hereby, the representative value is updated from v_r to $Rep(\mathbf{v}_r) = \mathbf{mu}_r$ for further smoothing the data. Note that the membership value denotes the degree that a point belonging to the fuzzy set, where $M(\mathbf{mu}_r) = 1$.

Fuzzy Rule Interpolation Let fuzzy variables sim_i, sim_j and sim_k ($k \in \{1,\ldots,c\}$) represent the similarity values between a labelled image in the training set and classes c_i, c_j and c_k, respectively, where c_i and c_j are the two given similes in respect to an unseen class. Each rule of the rule base is in the following form:

$$\textbf{If } sim_i = v_{c_i} \text{ and } sim_j = v_{c_j}, \textbf{ then } sim_k = v_{c_k}. \tag{4.14}$$

For example, given a pair of symbolic similes of an unseen class Leopard between class i Bobcat and class j Tiger, the similarity value of Leopard and class k lion is as shown in Fig. 4.6 with the corresponding rule as:

$$\begin{aligned}&\textbf{If } it\ is\ `to\ V_{c_i}\ extent\ like\ a\ Bobcat'\ and\ `to\ V_{c_j}\ extent\ like\ a\ Tiger',\\&\textbf{then } it\ is\ `to\ V_{c_k}\ extent\ like\ a\ Lion'.\end{aligned} \tag{4.15}$$

Then with the help of the proposed curvature-based method to selected important features and instances to construct rules, given two similes in the seen classes for an unseen class, the similarities of the unseen class regarding all seen classes can be inferred using FRI, which can then be utilised for unseen class classification.

As shown in Fig.4.11, the refined rule base is composed by R fuzzy rules. Given the initialised observation $\mathbf{v}' = [v'_{c_1}, ..., v'_{c_k}]$, we can construct its corresponding fuzzy set using eq.4.13. The new representative value is $Rep(\mathbf{v}) = mean(\mathbf{v}')$ instead of $\mathbf{v}'$. Our final step is to interpolate the real conclusion v_c using $\mathbf{v}$:

$$v_c = \frac{1}{R}\sum_{r=1}^{R} \lambda_r \mathbf{v}_r, \tag{4.16}$$

where λ_j is the interpolative ratio of the j^{th} fuzzy rule, which can be estimated by:

$$\lambda_r = \alpha_r \exp\left(-\frac{1}{2}(Rep(\mathbf{v}') - Rep(\mathbf{v}_r))^\top \Sigma_r^{-1}(Rep(\mathbf{v}') - Rep(\mathbf{v}_r))\right), \tag{4.17}$$

where $\alpha_j = \frac{1}{\sqrt{(2\pi)^2\|\Sigma_j\|}}, j \in \{1, ..., r\}$. Finally, we can repeat the process from Eq.4.5 to Eq.4.17 for each seen class i to infer an SV class-level prototype: $v^* = [Rep(V_1^*), ..., Rep(V_C^*)]$.

Using FIR, the light-weight sparse simile annotations are converted into a full simile vector for each unseen class u $\mathbf{v}_u = [v_1, ..., v_c, ...v_C]$. During the test, an unseen image $\hat{x}$ is also converted into a simile vector $f_1(\hat{\mathbf{x}}) = \hat{v}$ using Eq.4.9. Without loss of generality, the

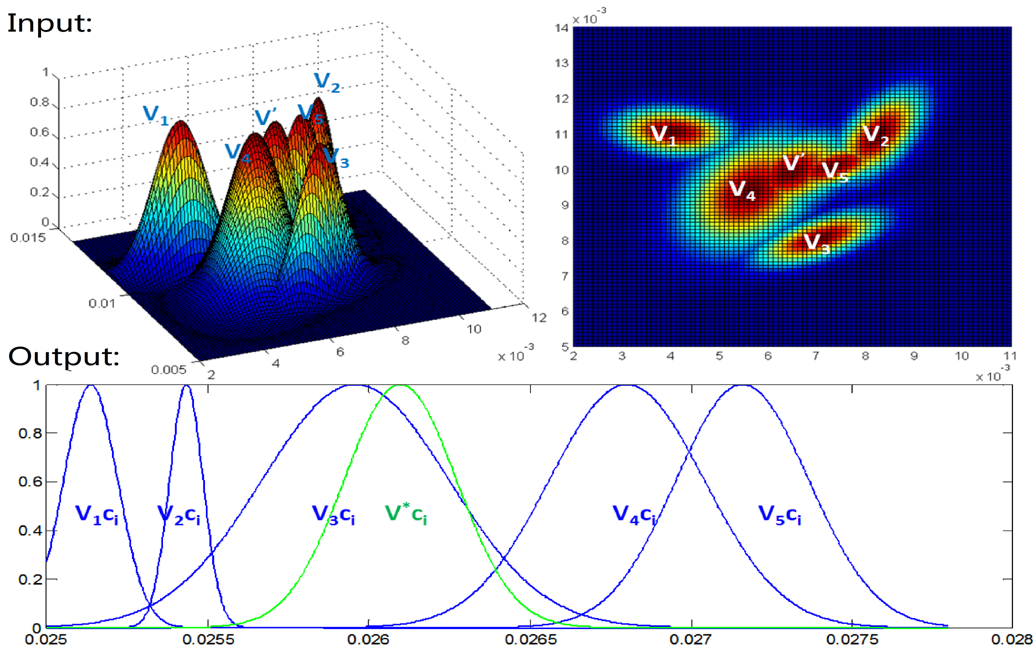

Fig. 4.11 Fuzzy Rule Interpolation. $V_1 : V_5$ are refined high-curvature rules. $V_{1c_i} : V_{5c_i}$ are corresponding output fuzzy sets. $V^*_{c_i}$ is the final interpolation result.

simplest nearest neighbour classifier is simply adopted to predict the label:

$$f_3 := \arg\min_u \|\hat{\mathbf{v}} - \mathbf{v}_u\|_2^2 \tag{4.18}$$

4.5.4 Inference Results and Analysis

4.4.4.1 Study with AwA and aPY

The proposed approach is firstly compared to state-of-the-art results. Since simile-based ZSL has only a few previous work, the comparison involves published results under various settings, frameworks, and visual/semantic data. The characteristics of the proposed method is detailed and try to illustrate how does each component contributes to the overall performance.

The proposed method was evaluated on AwA [265], and aPY [266] benchmarks. The standard 40/10 and 20/12 seen/unseen splits as in [267] are followed for the sake of fair comparison. The VGG-19 deep visual features released by [268] are adopted. Although the whole approach is non-parametric, there are four NN parameters: $[K, Q, R, T]$. The seen classes are divided into four folds. The leave-one-fold-out cross-validation is used to choose the best parameters and fix them for all of the test.

Table 4.15 Comparison to state-of-the-art methods.

Annotation Type	Method	AwA		aPY	
		WE	SV	WE	SV
Unsupervised	DeViSE [269]	44.5	47.5	25.5	27.4
	ConSE [270]	46.1	48.2	22.0	27.8
	Text2Visual [271]	55.3	-	30.2	-
	SynC [272]	57.5	58.9	-	-
	ALE [273]	58.8	60.1	33.3	36.2
	LatEm [274]	62.9	63.2	-	-
	CAAP [275]	67.5	-	37.0	-
	Attri2Classname [267]	69.9	-	38.2	-
	The proposed method	-	**78.5**	-	**48.8**
		Attri	SV	Attri	SV
Supervised	DAP[265]	54.0	58.5	28.5	36.6
	ENS [276]	57.4		31.7	
	HAT [277]	63.1		38.3	
	ALE-attr [273]	66.7	70.1	-	-
	SSE-INT [268]	71.5	-	44.2	-
	SSE-ReLU [268]	76.3	-	46.2	48.9
	SynC-attr [272]	76.3	78.5	-	-
	SDL [278]	79.1	82.2	50.4	52.5
	The proposed method	80.5	**83.2**	51.2	**56.7**

WE: Word Embedding; SV: Simile Vector; Attri: Attribute Embedding.

One of the main concerns is how to achieve similes of unseen classes. In this work, three protocols are defined :

1) Unsupervised Similes: class similarities are purely estimated by their word embeddings and top c_k similes are fed to FIR;

2) Attribute-based Similes: using conventional attributes as class embeddings, top c_k similes are computed for FIR inputs;

3) Supervised Similes: human intervention to the rank from protocol 1) to correct unsatisfied similes. The annotations use the judgements of a colleague who was unfamiliar with the details of this work.

Baselines

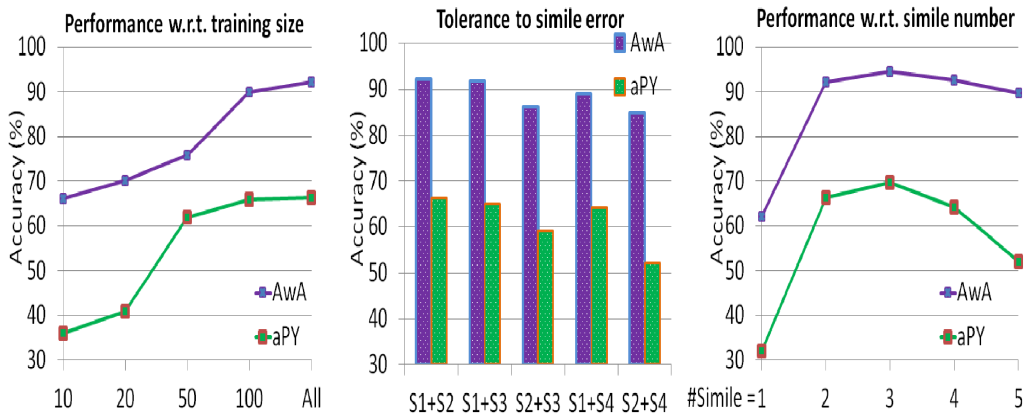

Fig. 4.12 Investigation of the characteristic of using similes.

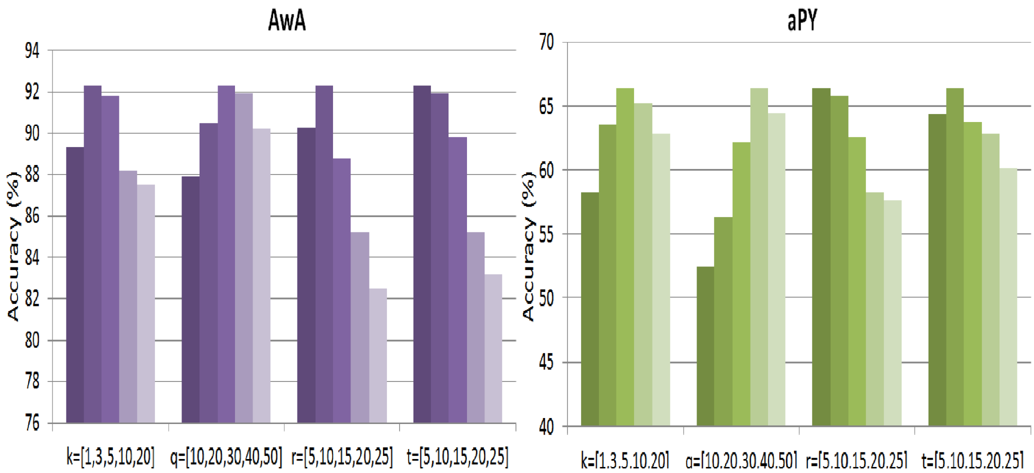

Fig. 4.13 Performance nearest neighbour max-pooling parameters.

Table 4.16 Upper bound increase using SV as representation.

	AwA		aPY	
Feature	Deep	Low-level	Deep	Low-level
Raw	92.33	80.64	94.82	84.73
SV	96.83	91.88	97.42	95.62

The main comparison is summarised in Table 4.15. It can be seen that most of the results of attribute-based approaches are better. The results can be categories by two dimensions. Firstly, the proposed method is compared to pure unsupervised approaches using word embedding-led similes. Then it is compared to conventional ZSL approaches using attributes to compute the similes. Furthermore, it provides light-weight human intervention to correct some simile errors due to semantic dominance from the above two embeddings. It focus on how much improvements can gain from the light-weight annotations. The averaged performance improvement is over 3.5%. Such a promising result indicate that the proposed method can provide an interesting interface for human-computer interaction to actively learn the parameters.

The other dimension is the comparison between conventional semantic models, i.e. WE and Attri, and the proposed SV. Also also implement some existing approaches with their released codes. Again, significant performance gains can be observed if substituting WE and Attri by the proposed inferred SV representation.

The proposed method steadily outperforms all of the above baselines. The success is ascribed to that encoding semantic similes by visual similarities between classes leads to little information loss comparing to attribute or word representations. Also, in contrast, the transductive setting is purely based on visual data distribution which may not be consistent with semantic distributions, whereas the proposed similes are directly related to class labels.

To understand the promising results, the proposed approach is discussed from following aspects that are supported by extensive studies.

Deep feature effect

To understand the contribution of the SV representation, it is separately studied using a supervised setting on seen classes. It can be seen that the supervised classification rates are remarkably increased, indicating the SV not only bridges the visual-semantic gap, but is a better visual representation as well. The proposed method is verified on conventional low-level features, e.g. a concatenation of PCA, PHOG, etc. to show its independence to deep features.

NN parameters

There are totally four NN processes in the proposed approach. NN is a simple way for max-pooling on the feature level that can suppress noise and reduce redundancy. In Fig. 4.13, the effects of each parameter on the overall performance by fixing the other three are shown. Generally, for MKE and high-curvature points, smaller k and r is better so as to pick out high-quality observations. In contrast, higher q and t are better so that the rule base can contain sufficient fuzzy rules.

4.4.4.2 Study with Caltech 101 Dataset

The first reason of revisiting Caltech 101 is to show the flexibility of the proposed method that can make ZSL readily applied on conventional datasets without providing attributes. Secondly, many related kernelised methods have published results on Caltech 101. The proposed ZSL method is compared with supervised results to show the improvement.

Setup

The work follows the conventional settings that use 15 or 30 images in each class for training. However, a supervised scenario is to use all classes for both training and testing, whereas the results are achieved under ZSL scenario where the seen/unseen split is 50/51, i.e. the first 50 classes are used for training and the other 51 for testing. We then swap the seen/unseen classes and use the averaged accuracy to evaluate the overall performance. Similes are used in the same way as for AwA and aPY in Section 4.5.4. The work experiment extracts SIFT, PHOG, LBP, and colour histogram and aggregates each type of local features using 500-D VLAD [263] and concatenates all of them into a rich representation.

Comparison with supervised results

Our comparison is summarised in Table 4.17. The compared previous methods use various kernel techniques. The proposed method under ZSL scenario exceeds the performances of most of the previous kernel methods under supervised scenarios. Although these results may not be the state-of-the-art now, it is still a fair comparison, in terms of the low-level features, match kernel techniques, which can verify the effectiveness of the proposed ZSL approach. Moreover, such results indicate ZSL can be applied to many other problems as Caltech 101 without providing extra attributes or other side information.

Generalised zero-shot learning

Generalised ZSL (GZSL) [279] is recently proposed to investigate how to break the restriction of ZSL by testing images from both seen and unseen classes. For seen classes, 15

Table 4.17 Revisit Caltech 101: comparison with supervised results.

Method	15 images/ seen Class	30 images/ seen class
Grauman& Darrell [280]	49.5	58.2
Tuytelaars [262]	61.3	69.6
Boiman [281]	65.0	70.4
Vedaldi [282]	**66.3**	-
The proposed method	65.8	**70.8**

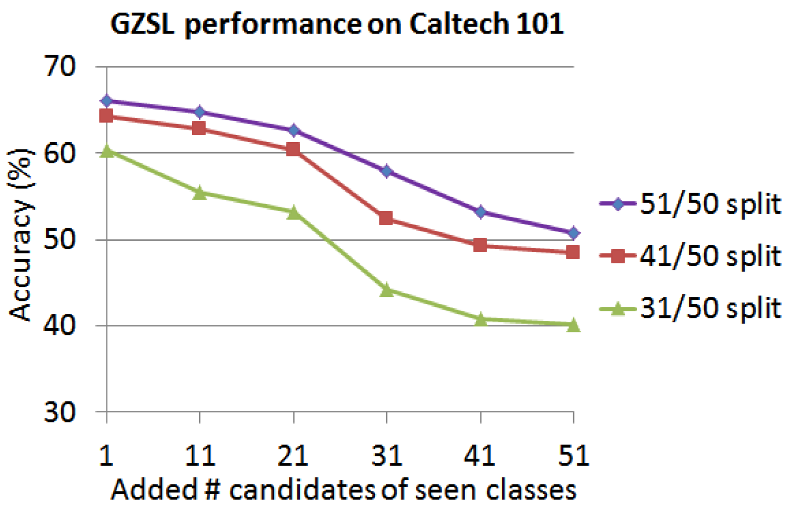

Fig. 4.14 Performance for Generalise ZSL evaluation.

images are used for training and the remaining for testing. 50 smaller-sized classes are used as unseen. GZSL is implemented by using the mean vSMS of each seen class as a prototype and gradually add seen classes as candidates into the unseen candidates. Besides, we also test GZSL using smaller numbers of training classes. The test classes are still 51+50. But we use only 41 and 31 seen classes for FIR, for which we have to change the similes of unseen class from the narrower range of training classes. It is shown in Fig. 4.14 that the proposed method can effectively differentiate unseen from seen classes. This is because the proposed vSMS is a representation of similarities. Unseen classes often have several similes with high SMS, whereas seen classes often have very high self-SMS. Using 41 seen classes for training can achieve comparable results as using 51. But for 31 training classes, we have to change many original similes that were from the other 20 classes, which degrades the performance to some extent.

Table 4.18 Results for comparison in application 3

Unsupervised	**Proposed Approach**	DeViSE [85]	ConSE [167]	SynC [37]	ALE [5]	LatEm [218]
Accuracy (%)	**78.5**	44.5 / 47.5	46.1 / 48.2	57.5 / 58.9	58.8 / 60.1	62.9 / 63.2

Table 4.19 Results 2 for comparison in application 3

Supervised	**Proposed Approach**	DAP[134]	ALE-attr [5]	SynC-attr [37]	SDL [255]
Accuracy (%)	**83.2**	54.0 / 58.5	66.7 / 70.1	76.3 / 78.5	79.1 / 82.2

4.4.4.3 Analysis

The proposed method can infer full associations between seen and unseen classes using only a few similes. It is the first time to introduce curvature-based sparse rule base generation and fuzzy rule interpolation into Zero-Shot learning image classification (computer vision field). The proposed method makes the image representation more interpretable and discriminative. Also, such linguistic representations could benefit the performance for GZSL. By regarding both input and output as fuzzy terms, the proposed method significantly boosted the ZSL performance by accurately predicting the SMS value of each seen class using only qualitative simile clues. The proposed method achieved state-of-the-art results on both AwA and aPY. Also, the application demonstrated how to simply apply the proposed approach to non-attribute problems, such as the Caltech 101 dataset.

The proposed method significantly reduces the required annotation cost compared to conventional supervised learning, with performance exceeds the performances of previous supervised approaches. The empirical study shows only two similes are enough to outperform existing ZSL approaches using attributes. According to the common 40/10 seen/unseen split of AwA (which has more than 20000 instances and each instance has more than 4000 features), the number of required labelling is reduced from 50×85 to only 10×2 that is only 0.47% of the original annotation work. The proposed method can effectively quantify similes and infer a reliable simile vector that can be used as improved semantic auxiliary information over conventional visual attributes. Despite low annotation cost, the proposed approach outperforms state-of-art approaches including those using heavy-annotated attributes.

4.6 Summary

This chapter has further illustrated the operational curvature-based sparse rule base generation method by applying the current research work into several applications. Firstly a simulated experiment (which considering a basic model with two inputs and one output) demonstrates the basic working procedure of the proposed approach, then the proposed method was evaluated using three real-world applications. The three applications (which considering more general situations with multiple inputs and one output) are: an indoor environment localisation problem, a student knowledge level evaluation problem, and the zero-shot learning image classification problem. In particular, it is the first time to introduce curvature-based sparse rule base generation and fuzzy rule interpolation into Zero-Shot learning image classification (computer vision field). The three benchmark datasets (including AwA, aPY and Caltech 101) have been used to do evaluation. The results have shown that the proposed method can infer full associations between seen and unseen classes using only a few similes. It makes the image representation more interpretable and discriminative (averaged performance improvement is over 3.5%). The promising results of all these applications have shown the power of the proposed approach in solving real-world problems.

However, the current applications are the first and initial attempts towards the use of curvature based sparse rule base generation. It is therefore necessary to carry out more extensive and complex applications to make the proposed approach improved to be a more generalised sparse rule base generation method for fuzzy rule interpolation.

Chapter 5

Conclusion

This chapter concludes the thesis and gives some suggestions for possible future developments. It presents a summary of the research as detailed in the previous chapters and discusses on some findings and contributions. The thesis has proposed a novel curvature-based sparse rule base generation method to support fuzzy rule interpolation, in an effort to make fuzzy reasoning systems more efficient and accurate. The proposed method works in this way: for problems with dense datasets, the approach first fuzzy partitions the problem domain into a number of hypercubes; or in the circumstance when only a small dataset is available, the approach represents each data instance as a hypercube. Then the hypercubes and their features are distinguished by effectively using their curvature values; thus each important hypercube with its important features is represented as a fuzzy rule and the most important hypercubes with their important features are selected to generate the final rule base. The capabilities and potential of the proposed method have been validated with experiment and real-world applications, and compared with conventional work it leads to very comparative results. However, further research is needed to enhance the proposed system. Some preliminary suggestions and future works have also been discussed in this chapter.

5.1 Discussion of the Proposed Work

There are generally two methods of fuzzy rule base generation, which are the data-driven method and the knowledge-driven method. The former one is implemented based on numerical data, which may suffer from the 'curse of dimensionality' as redundancy often exists in such rule bases. In order to address this, various rule base simplification approaches have been developed, and most of the existing data-driven rule base generation and reduction approaches are based on certain similarity measures; therefore, they are likely to cause performance deterioration along with size reduction. For the latter rule base generation approach, expert knowledge may be subjective, as different experts may have different opinions. These indicate that existing methods lack a more objective and efficient way to generate an effective sparse rule base, which distinguishes important rules by identifying important features and instances from the training data sets. In order to address this research gap, this project focused on how to keep a good balance between model accuracy, efficiency and transparency by proposing a subjective sparse rule base generation approach, i.e., how to generate a sparse rule base which is very accurate, computationally efficient, and linguistically tractable.

Compared to traditional fuzzy rule base generation methods in the literature, this work proposes a novel curvature-based sparse fuzzy rule base generation approach to support FRI. Firstly it presents the proposed approach in the basic case with two inputs and the general case with multiple inputs. As the curvature values are only workable with three-dimensional spaces, the basic inference (with two inputs and one output) is considered then followed by the general case with multiple inputs. Because a complex higher-dimensional problem can be regarded as a collection of smaller scale problems with two inputs and one output (i.e., multiple basic cases), any high-dimensional problems can be addressed by applying the basic case solutions multiple times. Therefore, the proposed approach first represents each data instance as a hypercube when handling a small dataset, or fuzzy partitions the problem domain into a number of hypercubes for problems with dense datasets. From this, the instances or hypercubes are discriminated by effectively using the curvature values and each important hypercube is represented as a fuzzy rule. In the complex problem, the performances of all rule bases in the basic cases are collected to identify the important features, thus the most important features of each instance (or hypercube) will also be discriminated. Then the most important instances (or hypercubes) with their important features will be selected to initialise the rule base. The initial rule base can be optionally optimised by fine-tuning the membership functions of the fuzzy sets involved in the rules by using an optimisation method, such as genetic algorithm (GA).

The experiments for system demonstration and validation show promising results. After a synthetic dataset to illustrate the working procedure, this work uses three real-world applications to demonstrate the power of the proposed approach in solving real-world problems. Firstly, in the indoor environment localisation problem, the accuracy of the classification result by the proposed method is 99.25%, which outperforms all the existing methods. Furthermore, if only the most important 2 to 6 input features are used (parts of the totally 7 features), the accuracy still remains 96.75%, 98.15%, 98.6%, 98.8%, and 99.15%, respectively. This clearly demonstrates the superior advantage of the proposed approach. Secondly, in the student knowledge level evaluation problem, the results also prove the proposed method can achieve competitive performance (accuracy 95.9%) by only using several important instances with several important features. Finally, in the zero-shot learning image classification problem, the proposed method makes the image representation more interpretable and discriminative. The performance significantly exceeds those of previous supervised approaches (with an average performance improvement of around 3.5%).

There are two breakthroughs or improvements during this piece of PhD work. One breakthrough is to apply curvature values into fuzzy rule base generation and to extend the three-dimensional curvature idea into high-dimensional problems. However, given that traditional curvature values only work with three-dimensional data, the most challenging part is to develop an approach to calculate the artificial 'curvature' values in a high-dimensional space. In this work, by regarding a higher-dimensional complex problem as a collection of three dimensional problems with basic case solution, any high-dimensional problems can thus be addressed by applying the proposed basic case solutions multiple times. The second breakthrough is the design of an efficient framework to innovatively apply curvature based sparse rule base generation method into real-world problems, including real-world image classification problem. The key idea is inspired by an intuitive fact that to describe an unseen instance the most straightforward way is to relate it to previously seen classes. Therefore, an efficient framework was proposed and it can recognise unseen classes with light-weight simile annotations, as explained as in Fig. 4.6. Semantic concepts are complex and structured whereas the label space for most of current supervised learning consists of discrete and disjoint one-hot category vectors. Therefore, in traditional machine-learning methods the associations between classes are imposed to be neglected and the labelling work is always inefficient. Compared to attributes or texts, similes are more visual-related and do not involve extra concepts, thus lead to less information loss. Assigning only two similes for each seen class and using curvature values to select important rules, the proposed method makes the image representation more interpretable and discriminative, with results significantly boost the ZSL performance.

In summary, significant contributions have been made during the development of this PhD research project. A novel rule base generation approach has been proposed which applies curvature values to select important instances and features to construct rules. The proposed approach provides an objective tool to distinguish important instances and features, and thus to ease the subjectiveness of expert knowledge. The proposed approach also alleviates the 'curse of dimensionality' by efficiently generating sparse rule bases with a smaller number of important rules. This project has also extended the basic curvature-based feature and instance selection idea into high-dimensional situations. The extended approach has been innovatively applied to several real-world problems, including an efficient framework for light-weight simile annotations, with competitive results generated.

5.2 Future Works

Although the experimental results demonstrated the promise of the proposed method, much can be done to further improve the work presented in this thesis. The following points address a number of interesting issues and the successful solutions for them will help to establish the current research on a more robust foundation. Also, it is important to apply this proposed approach to more real-life applications in order to test the practical usability of the approach.

One possible future direction is to extend the traditional curvature value calculation in a more effective way (given that the current approach is combinational and thus requires high computational power), to support curvature-based rule base generation. In this thesis, a higher-dimensional complex problem is regarded as a collection of simpler problems with two inputs and one output (i.e., multiple basic cases). With the curvature-based approach discussed in Section 3.2, any high-dimensional problems are addressed by applying the basic case solutions multiple times. If a mathematical equation can be directly designed to calculate a high-dimensional 'curvature' value then it is much more efficient to evaluate the importance of a high-dimensional instance.

The proposed approach creates new rules by applying many operations such as partitioning and aggregation, to the interpolated results. This is implemented with the use of the given weighted aggregation method. It would be useful to test the application of other aggregation operators such as ordered weighted average (OWA), generalised ordered weighted average (GOWA), and induced ordered weighted average (IOWA) as discussed in Section 3.2.3. If better rules can be obtained using such alternative aggregation operators then this will help improve the accuracy of the created rules and therefore improve the accuracy of the proposed method.

In this research work, for the purpose of preliminary investigation and experimentation, only the KH and HS approaches are employed. It is interesting to investigate the application of other FRI techniques. A comparative study may be helpful to investigate the most effective and accurate curvature-based sparse rule base using a certain FRI method. This work could be extended using the approaches reviewed in Section 1.2. This is certainly a systematic work that requires a lot of additional experiments.

The proposed work divides high dimensional problem into sub-problem in three dimensions and uses a one-time process to carry out curvature calculation off-line, thus helps to reduce the computational cost for sparse rule base generation. For example, in benchmark dataset AwA, the number of required labelling is reduced from 50×85 to only 10×2 that is only 0.47% of the original annotation work. However, more investigation on computational cost should be considered in future work.

Theoretically, the curvature-based approaches can be readily generalised as a generic feature selection approach to work collectively with other decision making algorithms, such as classifiers. This will greatly widen the applicability of the proposed approach. It will be very interesting to develop a general curvature-based framework to include all the existing curvature approaches, and their extensions for multiple dimensional real-world problems, given that many curvature calculations approaches have been proposed in the literature. From this, it also worthwhile to conduct a comparative study to further investigate the pros and cons of each approach working along with different classifiers and datasets.

Although three real-world applications have been included in this project, the proposed approach can be readily applied to more real-world applications for the development of concise rule bases and thus more efficient fuzzy inference systems. In addition, fuzzy inference has been used along with other machine learning approaches for better performance, such as [283, 284], it is therefore worthwhile to investigate the potential of the proposed approach in such situations.

References

[1] Zhiheng Huang. Rule model simplification. 2006.

[2] J. Li, H. P. H. Shum, X. Fu, G. Sexton, and L. Yang. Experience-based rule base generation and adaptation for fuzzy interpolationn. In *IEEE World Congress on Computation Intelligence Internation Conference*, 2016.

[3] Nitin Naik, Ren Diao, Chai Quek, and Qiang Shen. Towards dynamic fuzzy rule interpolation. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pages 1–7. IEEE, 2013.

[4] Jie Li, Yanpeng Qu, Hubert PH Shum, and Longzhi Yang. TSK inference with sparse rule bases. In *Advances in Computational Intelligence Systems*, pages 107–123. Springer, 2017.

[5] Chun-Tian Cheng, Jian-Yi Lin, Ying-Guang Sun, and Kwokwing Chau. Long-term prediction of discharges in manwan hydropower using adaptive-network-based fuzzy inference systems models. In *International Conference on Natural Computation*, pages 1152–1161. Springer, 2005.

[6] Timothy J Ross. *Fuzzy logic with engineering applications*. John Wiley & Sons, 2005.

[7] André Jones, Arnold Kaufmann, and Hans-Jürgen Zimmermann. *Fuzzy sets theory and applications*, volume 177. Springer Science & Business Media, 2012.

[8] Kaveh Khalili-Damghani, Soheil Sadi-Nezhad, Farhad Hosseinzadeh Lotfi, and Madjid Tavana. A hybrid fuzzy rule-based multi-criteria framework for sustainable project portfolio selection. *Information Sciences*, 220:442–462, 2013.

[9] Pradeep Singh, Nikhil R Pal, Shrish Verma, and Om Prakash Vyas. Fuzzy rule-based approach for software fault prediction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(5):826–837, 2016.

[10] Susmita Kar. A comprehensive protection scheme for micro-grid using fuzzy rule base approach. *Energy Systems*, 8(3):449–464, 2017.

[11] YI Kudinov, VA Kolesnikov, FF Pashchenko, AF Pashchenko, and L Papic. Optimization of fuzzy pid controller's parameters. *Procedia Computer Science*, 103:618–622, 2017.

[12] Alberto Fernández, Sara del Río, Abdullah Bawakid, and Francisco Herrera. Fuzzy rule based classification systems for big data with mapreduce: granularity analysis. *Advances in Data Analysis and Classification*, 11(4):711–730, 2017.

[13] TV Avdeenko and ES Makarova. Integration of case-based and rule-based reasoning through fuzzy inference in decision support systems. *Procedia Computer Science*, 103:447–453, 2017.

[14] Giancarlo Lucca, José Antonio Sanz, Graçaliz Pereira Dimuro, Benjamin Bedregal, Maria José Asiain, Mikel Elkano, and Humberto Bustince. Cc-integrals: Choquet-like copula-based aggregation functions and its application in fuzzy rule-based classification systems. *Knowledge-Based Systems*, 119:32–43, 2017.

[15] Mehrbakhsh Nilashi, Othman Ibrahim, Hossein Ahmadi, and Leila Shahmoradi. A knowledge-based system for breast cancer classification using fuzzy logic method. *Telematics and Informatics*, 34(4):133–144, 2017.

[16] Clarence W De Silva. *Intelligent control: fuzzy logic applications*. CRC press, 2018.

[17] Liviu-Cristian Duţu, Gilles Mauris, and Philippe Bolon. A fast and accurate rule-base generation method for mamdani fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 26(2):715–733, 2018.

[18] Ehsan Pourjavad and Rene V Mayorga. A comparative study and measuring performance of manufacturing systems with mamdani fuzzy inference system. *Journal of Intelligent Manufacturing*, 30(3):1085–1097, 2019.

[19] K Bipirayeh, O Abedinia, and HA Shayanfar. Optimal multi-stage fuzzy pid bundled psotvac in multimachine environment. *International Journal on Technical and Physical Problems of Engineering (IJTPE)*, 14:37–43, 2019.

[20] Russ Eberhart, Pat Simpson, and Roy Dobbins. *Computational intelligence PC tools*. Academic Press Professional, Inc., 1996.

[21] Masaharu Mizumoto and Hans-Jürgen Zimmermann. Comparison of fuzzy reasoning methods. *Fuzzy sets and systems*, 8(3):253–283, 1982.

[22] Shyi-Ming Chen and Yu-Chuan Chang. Weighted fuzzy interpolative reasoning for sparse fuzzy rule-based systems. *Expert Systems with Applications*, 38(8):9564–9572, 2011.

[23] Zhiheng Huang and Qiang Shen. Fuzzy interpolative reasoning via scale and move transformations. *Fuzzy Systems, IEEE Transactions on*, 14(2):340–359, 2006.

[24] Longzhi Yang and Qiang Shen. Adaptive fuzzy interpolation and extrapolation with multiple-antecedent rules. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8. IEEE, 2010.

[25] Longzhi Yang and Qiang Shen. Adaptive fuzzy interpolation. *Fuzzy Systems, IEEE Transactions on*, 19(6):1107–1126, 2011.

[26] Longzhi Yang and Qiang Shen. Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1–22, 2013.

[27] Gergely I Molnárka, Szilveszter Kovács, and László T Kóczy. Fuzzy rule interpolation based fuzzy signature structure in building condition evaluation. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2214–2221. IEEE, 2014.

[28] Jie Li, Longzhi Yang, Hubert P. H. Shum, and Yao Tan. Intelligent home heating controller using fuzzy rule interpolation. In *The 15th UK Workshop on Computational Intelligence (UKCI'2015)*, 2015.

[29] Longzhi Yang, Jie Li, Gerhard Fehringer, Phoebe Barraclough, Graham Sexton, and Yi Cao. Intrusion detection system by fuzzy interpolation. In *Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[30] Jie Li, Longzhi Yang, Xin Fu, Fei Chao, and Yanpeng Qu. Dynamic QoS solution for enterprise networks using TSK fuzzy interpolation. In *Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[31] Yang Long, Yao Tan, Daniel Organisciak, Longzhi Yang, and Ling Shao. Towards light-weight annotations: Fuzzy interpolative reasoning for Zero-shot image classification. In *BMVC 2018, The 29th British Machine Vision Conference on*. IEEE, 2018.

[32] Longzhi Yang, Jie Li, Fei Chao, Phil Hackney, and Mark Flanagan. Job shop planning and scheduling for manufacturers with manual operations. *Expert Systems*, page e12315, 2018.

[33] Noe Elisa, Jie Li, Zheming Zuo, and Longzhi Yang. Dendritic cell algorithm with fuzzy inference system for input signal generation. In *UK Workshop on Computational Intelligence*, pages 203–214. Springer, 2018.

[34] Zsolt Csaba Johanyák and Szilveszter Kovács. Fuzzy rule interpolation based on polar cuts. In *Computational Intelligence, Theory and Applications*, pages 499–511. Springer, 2006.

[35] Yu-Chuan Chang, Shyi-Ming Chen, Churn-Jung Liau, et al. Fuzzy interpolative reasoning for sparse fuzzy-rule-based systems based on the areas of fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 16(5):1285, 2008.

[36] Kok Wai Wong, Domonkos Tikk, Tamás D Gedeon, and László T Kóczy. Fuzzy rule interpolation for multidimensional input spaces with applications: A case study. *IEEE transactions on fuzzy systems*, 13(6):809–819, 2005.

[37] Z. Csaba Johanyák and S. Kovács. A brief survey and comparison on various interpolation based fuzzy reasoning methods. *Acta Polytechnica Hungarica*, 3(1):91–105, 2006.

[38] ZC Johanyák. Clonal selection based parameter optimization for sparse fuzzy systems. In *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)*, pages 369–373. IEEE, 2012.

[39] I. Perfilieva, D. Dubois, H. Prade, F. Esteva, L. Godo, and P. Hoďáková. Interpolation of fuzzy data: Analytical approach and overview. *Fuzzy Sets and Systems*, 192(0):134 – 158, 2012. Fuzzy Set Theory — Where Do We Stand and Where Do We Go?

[40] Min-You Chen and Derek A Linkens. Rule-base self-generation and simplification for data-driven fuzzy models. In *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, volume 1, pages 424–427. IEEE, 2001.

[41] Li-Xin Wang and Jerry M Mendel. Generating fuzzy rules by learning from examples. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(6):1414–1427, 1992.

[42] Raouf Ketata, Hatem Bellaaj, Mohamed Chtourou, and Mohamed Ben Amer. Adjustment of membership functions, generation and reduction of fuzzy rule base from numerical data. *Malaysian Journal of Computer Science*, 20(2):147–169, 2017.

[43] Zsolt Csaba Johanyák and Szilveszter Kovács. Sparse fuzzy system generation by rule base extension. In *Proceedings of 11th IEEE International Conference of Intelligent Engineering Systems (INES 2007), Budapest, Hungary*, pages 99–104. Citeseer, 2007.

[44] Xiaowei Xu, Martin Ester, H-P Kriegel, and Jörg Sander. A distribution-based clustering algorithm for mining in large spatial databases. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 324–331. IEEE, 1998.

[45] Plamen P Angelov. An evolutionary approach to fuzzy rule-based model synthesis using indices for rules. *Fuzzy sets and systems*, 137(3):325–338, 2003.

[46] Plamen P Angelov and Richard A Buswell. Automatic generation of fuzzy rule-based models from data by genetic algorithms. *Information Sciences*, 150(1-2):17–31, 2003.

[47] Shin-Jye Lee, Xiao-Jun Zeng, and Hui-Shin Wang. Generating automatic fuzzy system from relational database system for estimating null values. *Cybernetics and Systems: An International Journal*, 40(6):528–548, 2009.

[48] Di Wang, Xiao-Jun Zeng, and John A Keane. An evolving-construction scheme for fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 18(4):755–770, 2010.

[49] Yeung Yam and Laszlo T Koczy. Cartesian representation for fuzzy interpolation. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 3, pages 2936–2937. IEEE, 1998.

[50] Laszlo T Koczy and Kaoru Hirota. Size reduction by interpolation in fuzzy rule bases. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(1):14–25, 1997.

[51] Chin-Wang Tao. A reduction approach for fuzzy rule bases of fuzzy controllers. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 32(5):668–675, 2002.

[52] A Krone, P Krause, and T Slawinski. A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces. In *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, volume 2, pages 694–699. IEEE, 2000.

[53] Min-You Chen and Derek A Linkens. Rule-base self-generation and simplification for data-driven fuzzy models. In *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, volume 1, pages 424–427. IEEE, 2001.

[54] Min-You Chen and Derek A Linkens. Rule-base self-generation and simplification for data-driven fuzzy models. *Fuzzy sets and systems*, 142(2):243–265, 2004.

[55] Magne Setnes and Robert Babuska. Rule base reduction: Some comments on the use of orthogonal transforms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(2):199–206, 2001.

[56] N Xiong and Lothar Litz. Reduction of fuzzy control rules by means of premise learning–method and case study. *Fuzzy Sets and Systems*, 132(2):217–231, 2002.

[57] Feng Liu, Chai Quek, and Geok See Ng. A novel generic hebbian ordering-based fuzzy rule base reduction approach to mamdani neuro-fuzzy system. *Neural Computation*, 19(6):1656–1680, 2007.

[58] Hatem Bellaaj, Rouf Ketata, and Mohamed Chtourou. A new method for fuzzy rule base reduction. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 25(3):605–613, 2013.

[59] Pintu Chandra Shill, MAH Akhand, MD Asaduzzaman, and Kazuyuki Murase. Optimization of fuzzy logic controllers with rule base size reduction using genetic algorithms. *International Journal of Information Technology & Decision Making*, 14(05):1063–1092, 2015.

[60] Arturo Garcia-Garcia, Marek Z Reformat, and Andres Mendez-Vazquez. Similarity-based method for reduction of fuzzy rules. In *Fuzzy Information Processing Society (NAFIPS), 2016 Annual Conference of the North American*, pages 1–6. IEEE, 2016.

[61] Qian Ning Zhang, Ze Chun Huang, Zhu Xu, and Hai Bin Shang. Study on sampling rule and simplification of LiDAR point cloud based on terrain complexity. *Journal of the Indian Society of Remote Sensing*, 46(11):1773–1784, 2018.

[62] Sylvie Galichet and Laurent Foulloy. Size reduction in fuzzy rulebases. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 3, pages 2107–2112. IEEE, 1998.

[63] LászlóT Kóczy and Kaoru Hirota. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197–225, 1993.

[64] Veronique Adriaenssens, Bernard De Baets, Peter LM Goethals, and Niels De Pauw. Fuzzy rule-based models for decision support in ecosystem management. *Science of the Total Environment*, 319(1-3):1–12, 2004.

[65] SN Sivanandam, Sai Sumathi, SN Deepa, et al. *Introduction to fuzzy logic using MATLAB*, volume 1. Springer, 2007.

[66] Valerie Cross and Thomas Sudkamp. Fuzzy implication and compatibility modification. In *Fuzzy Systems, 1993., Second IEEE International Conference on*, pages 219–224. IEEE, 1993.

[67] Lotfi A Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on systems, Man, and Cybernetics*, (1):28–44, 1973.

[68] Lotfi A Zadeh, King-Sun Fu, and Kokichi Tanaka. *Fuzzy sets and their applications to cognitive and decision processes: Proceedings of the us–japan seminar on fuzzy sets and their applications, held at the university of california, berkeley, california, july 1-4, 1974*. Academic press, 2014.

[69] Ebrahim H Mamdani and Sedrak Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1):1–13, 1975.

[70] Ebrahim H Mamdani. Advances in the linguistic synthesis of fuzzy controllers. *International Journal of Man-Machine Studies*, 8(6):669–678, 1976.

[71] Li-Wei Lee and Shyi-Ming Chen. A new fuzzy interpolative reasoning method for sparse fuzzy rule-based systems. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 745–755. Springer, 2007.

[72] Michio Sugeno and GT Kang. Structure identification of fuzzy model. *Fuzzy sets and systems*, 28(1):15–33, 1988.

[73] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *Systems, Man and Cybernetics, IEEE Transactions on*, (1):116–132, 1985.

[74] Yahachiro Tsukamoto. An approach to fuzzy reasoning method. *Advances in fuzzy set theory and applications*, 1979.

[75] Yahachiro Tsukamoto. Fuzzy logic based on lukasiewicz logic and its applications to diagnosis and control. *Doctral Dissertation, Tokyo Inst. of Tehcnol.*, 1979.

[76] Ebrahim H Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *Computers, IEEE Transactions on*, 100(12):1182–1191, 1977.

[77] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - i. *Information Sciences*, 8(3):199 – 249, 1975.

[78] Chuen Chien Lee. Fuzzy logic in control systems: fuzzy logic controller. ii. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):419–435, 1990.

[79] Maria Jose Gacto, Rafael Alcalá, and Francisco Herrera. Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, 181(20):4340–4360, 2011.

[80] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[81] William Siler and James J Buckley. *Fuzzy expert systems and fuzzy reasoning*. John Wiley & Sons, 2005.

[82] Rafael Alcalá, Jesús Alcalá-Fdez, Jorge Casillas, Oscar Cordón, and Francisco Herrera. Hybrid learning models to get the interpretability–accuracy trade-off in fuzzy modeling. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 10(9):717–734, 2006.

[83] Xin Fu, Xizo jun Zeng, Di Wang, Di Xu, and Longzhi Yang. Fuzzy system approaches to negotiation pricing decision support. *Journal of Intelligent & Fuzzy Systems*, 29(2):685–699, 2015.

[84] Longzhi Yang, Daniel Neagu, Mark T. D. Cronin, Mark Hewitt, Steven J. Enoch, Judith C. Madden, and Katarzyna Przybylak. Towards a fuzzy expert system on toxicological data quality assessment. *Molecular Informatics*, 32(1):65–78, 2013.

[85] Alberto Fernández and Francisco Herrera. Linguistic fuzzy rules in data mining: follow-up mamdani fuzzy modeling principle. In *Combining Experimentation and Theory*, pages 103–122. Springer, 2012.

[86] Shi Yan, Masaharu Mizumoto, and Wu Zhi Qiao. Reasoning conditions on koczy's interpolative reasoning method in sparse fuzzy rule bases. *Fuzzy Sets and Systems*, 75(1):63–71, 1995.

[87] Bernadette Bouchon-Meunier, Didier Dubois, Christophe Marsala, Henri Prade, and Laurent Ughetto. A comparative view of interpolation methods between sparse fuzzy rules. In *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, pages 2499–2504. IEEE, 2001.

[88] Sándor Jenei. Interpolation and extrapolation of fuzzy quantities revisited–an axiomatic approach. *Soft Computing*, 5(3):179–193, 2001.

[89] Z. Csaba Johanyák and S. Kovács. A brief survey and comparison on various interpolation based fuzzy reasoning methods. *Acta Polytechnica Hungarica*, 3(1):91–105, 2006.

[90] Z. Csaba Johanyák and S. Kovács. Survey on various interpolation based fuzzy reasoning methods. *Prod. Syst. Inf. Eng*, 3:39–56, 2006.

[91] S. Chen and Y. Chang. A new method for fuzzy rule interpolation based on the ratio of fuzziness of interval type-2 fuzzy sets. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–6, 2010.

[92] L. Yang, C. Chen, N. Jin, X. Fu, and Q. Shen. Closed form fuzzy interpolation with interval type-2 fuzzy sets. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2184–2191. IEEE, 2014.

[93] LászlóT. Kóczy and Kaoru Hirota. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197 – 225, 1993.

[94] LT Kóczy and K Hirota. Rule interpolation by α-level sets in fuzzy approximate reasoning. *J. BUSEFAL, Automne, URA-CNRS*, 46:115–123, 1991.

[95] L. T. Kóczy and K. Hirota. Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Sciences*, 71(1):169–201, 1993.

[96] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.

[97] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.

[98] Lotfi A Zadeh. Quantitative fuzzy semantics. *Information Sciences*, 3(2):159–176, 1971.

[99] Yu-Chuan Chang, Shyi-Ming Chen, and Churn-Jung Liau. Fuzzy interpolative reasoning for sparse fuzzy-rule-based systems based on the areas of fuzzy sets. *Fuzzy Systems, IEEE Transactions on*, 16(5):1285–1301, 2008.

[100] Y. Chang, S. Chen, and C. Liau. Fuzzy interpolative reasoning for sparse fuzzy-rule-based systems based on the areas of fuzzy sets. *Fuzzy Systems, IEEE Transactions on*, 16(5):1285–1301, 2008.

[101] W. Hsiao, S. Chen, and C. Lee. A new interpolative reasoning method in sparse rule-based systems. *Fuzzy Sets and Systems*, 93(1):17 – 22, 1998.

[102] Kiyohiko Uehara and Kaoru Hirota. Multi-level interpolation for inference with sparse fuzzy rules: An extended way of generating multi-level points. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 17(2):127–148, 2013.

[103] Zhi Qiao Wu, Mizumoto Masaharu, and Yan Shi. An improvement to kóczy and hirota's interpolative reasoning in sparse fuzzy rule bases. *International Journal of Approximate Reasoning*, 15(3):185 – 201, 1996.

[104] Domonkos Tikk, István Joó, László Kóczy, Péter Várlaki, Bernhard Moser, and Tamás D Gedeon. Stability of interpolative fuzzy kh controllers. *Fuzzy Sets and Systems*, 125(1):105–119, 2002.

[105] B. Bouchon-Meunier and L. Valverde. A fuzzy approach to analogical reasoning. *Soft Computing*, 3(3):141–147, 1999.

[106] Zhiheng Huang and Qiang Shen. Fuzzy interpolation and extrapolation: A practical approach. *Fuzzy Systems, IEEE Transactions on*, 16(1):13–28, 2008.

[107] Qiang Shen and Longzhi Yang. Generalisation of scale and move transformation-based fuzzy interpolation. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 15(3):288–298, 2011.

[108] Didier Dubois and Henri Prade. Gradual inference rules in approximate reasoning. *Information Sciences*, 61(1-2):103–122, 1992.

[109] L Ding, Z Shen, and M Mukaidono. A new method for approximate reasoning. In *Proceedings The Nineteenth International Symposium on Multiple-Valued Logic*, pages 179–185. IEEE, 1989.

[110] 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.

[111] L. Yang and Q. Shen. Towards adaptive interpolative reasoning. In *2009 IEEE International Conference on Fuzzy Systems*, pages 542–549, Aug 2009.

[112] L. Yang and Q. Shen. Extending adaptive interpolation: From triangular to trapezoidal. In *2009 UK Workshop on Computational Intelligence*, pages 25–30, Sep 2009.

[113] Longzhi Yang, Fei Chao, and Qiang Shen. Generalised adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 25(4):839–853, 2017.

[114] K. Yin, K. Xiang, M. Pang, J. Chen, P. Anderson, and L. Yang. Personalised control of robotic ankle exoskeleton through experience-based adaptive fuzzy inference. *IEEE Access*, 7:72221–72233, 2019.

[115] Johan De Kleer and Brian C Williams. Diagnosing multiple faults. *Artificial intelligence*, 32(1):97–130, 1987.

[116] Johan De Kleer. An assumption-based tms. *Artificial intelligence*, 28(2):127–162, 1986.

[117] Carl De Boor, Carl De Boor, Etats-Unis Mathématicien, Carl De Boor, and Carl De Boor. *A practical guide to splines*, volume 27. Springer-Verlag New York, 1978.

[118] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. Numerical recipes: The art of scientific computing (new york. *Cambridge University Press*, pages 636–9, 1992.

[119] Shyi-Ming Chen and Dipto Barman. Adaptive fuzzy interpolative reasoning based on similarity measures of polygonal fuzzy sets and novel move and transformation techniques. *Information Sciences*, 489:303–315, 2019.

[120] Jing Yang, Changjing Shang, Ying Li, Fangyi Li, and Qiang Shen. Anfis construction with sparse data via group rule interpolation. *IEEE Transactions on Cybernetics*, 2019.

[121] N. Naik, R. Diao, and Q. Shen. Genetic algorithm-aided dynamic fuzzy rule interpolation. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2198–2205, 2014.

[122] Nitin Naik, Ren Diao, and Qiang Shen. Dynamic fuzzy rule interpolation and its application to intrusion detection. *IEEE Transactions on Fuzzy Systems*, 26(4):1878–1892, 2018.

[123] Nitin Naik, Changjing Shang, Qiang Shen, and Paul Jenkins. D-fri-ciscofirewall: Dynamic fuzzy rule interpolation for cisco asa firewall. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6. IEEE, 2019.

[124] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[125] BB Chaudhuri and Gautam Garai. Grid clustering with genetic algorithm and tabu search process. *Journal of Pattern Recognition Research*, 4(1):152–168, 2009.

[126] Bogdan Gabrys and Andrzej Bargiela. General fuzzy min-max neural network for clustering and classification. *IEEE transactions on neural networks*, 11(3):769–783, 2000.

[127] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. 1973.

[128] L-X Wang and Jerry M Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on systems, man, and cybernetics*, 22(6):1414–1427, 1992.

[129] Bin-Da Liu, Chuen-Yau Chen, and Ju-Ying Tsao. Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(1):32–53, 2001.

[130] Jorge Casillas, Oscar Cordón, and Francisco Herrera. Cor: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 32(4):526–537, 2002.

[131] Hisao Ishibuchi and Tomoharu Nakashima. Effect of rule weights in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 9(4):506–515, 2001.

[132] Oscar Cordón, Francisco Herrera, and Pedro Villar. Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. *IEEE Transactions on fuzzy systems*, 9(4):667–674, 2001.

[133] Michela Antonelli, Pietro Ducange, Beatrice Lazzerini, and Francesco Marcelloni. Learning concurrently partition granularities and rule bases of mamdani fuzzy systems in a multi-objective evolutionary framework. *International Journal of Approximate Reasoning*, 50(7):1066–1080, 2009.

[134] Pei-Chann Chang, Chen-Hao Liu, and Robert K Lai. A fuzzy case-based reasoning model for sales forecasting in print circuit board industries. *Expert Systems with Applications*, 34(3):2049–2058, 2008.

[135] Mladenka Blagojević, Milica Šelmić, Dragana Macura, and Dragana Šarac. Determining the number of postal units in the network–fuzzy approach, serbia case study. *Expert Systems with Applications*, 40(10):4090–4095, 2013.

[136] Liviu-Cristian Duţu, Gilles Mauris, and Philippe Bolon. A linear-complexity rule base generation method for fuzzy systems. In *IFSA-EUSFLAT 2015-16th World Congress of the International Fuzzy Systems Association (IFSA)-9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)*, pages pp–520. Atlantis Press, 2015.

[137] Oscar Cordón, Francisco Herrera, and Igor Zwir. Linguistic modeling by hierarchical systems of linguistic rules. *IEEE Transactions on fuzzy systems*, 10(1):2–20, 2002.

[138] Mohammad Shokouhifar and Ali Jalali. Optimized sugeno fuzzy clustering algorithm for wireless sensor networks. *Engineering applications of artificial intelligence*, 60:16–25, 2017.

[139] Stelios Krinidis and Vassilios Chatzis. A robust fuzzy local information c-means clustering algorithm. *IEEE Transactions on Image Processing*, 19(5):1328–1337, 2010.

[140] Francisco José Berlanga, AJ Rivera, María José del Jesús, and Francisco Herrera. Gp-coach: Genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems. *Information Sciences*, 180(8):1183–1200, 2010.

[141] ZS Zu. Generalized chi square method for the estimation of weights. *Journal of Optimization Theory and Applications*, 107(1):183–192, 2000.

[142] John C Harsanyi. Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility. *Journal of political economy*, 63(4):309–321, 1955.

[143] Ronald R Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1):183–190, 1988.

[144] Ronald R Yager and Dimitar P Filev. Induced ordered weighted averaging operators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(2):141–150, 1999.

[145] Vicenç Torra. The weighted owa operator. *International Journal of Intelligent Systems*, 12(2):153–166, 1997.

[146] Ronald R Yager. Generalized owa aggregation operators. *Fuzzy Optimization and Decision Making*, 3(1):93–107, 2004.

[147] WM Dong and FS Wong. Fuzzy weighted averages and implementation of the extension principle. *Fuzzy sets and systems*, 21(2):183–199, 1987.

[148] Ram Narasimhan. A geometric averaging procedure for constructing supertransitive approximation to binary comparison matrices. *Fuzzy Sets and Systems*, 8(1):53–61, 1982.

[149] Janos Aczél and Thomas L Saaty. Procedures for synthesizing ratio judgements. *Journal of mathematical Psychology*, 27(1):93–102, 1983.

[150] Zeshui Xu. On consistency of the weighted geometric mean complex judgement matrix in ahp. *European journal of operational research*, 126(3):683–687, 2000.

[151] Sergei Ovchinnikov. An analytic characterization of some aggregation operators. *International Journal of Intelligent Systems*, 13(1):59–68, 1998.

[152] Ronald R Yager. A note on weighted queries in information retrieval systems. *Journal of the American Society for Information Science*, 38(1):23–24, 1987.

[153] Gloria Bordogna and Gabriella Pasi. Linguistic aggregation operators of selection criteria in fuzzy information retrieval. *International journal of intelligent systems*, 10(2):233–248, 1995.

[154] Ronald R Yager, Dimitar P Filev, and Tom Sadeghi. Analysis of flexible structured fuzzy logic controllers. *IEEE transactions on systems, man, and cybernetics*, 24(7):1035–1043, 1994.

[155] Ronald R Yager and Dimitar P Filev. Generalizing the modeling of fuzzy logic controllers by parameterized aggregation operators. *Fuzzy Sets and Systems*, 70(2-3):303–313, 1995.

[156] Ronald R Yager. A general approach to criteria aggregation using fuzzy measures. *International Journal of Man-Machine Studies*, 39(2):187–213, 1993.

[157] Vincenzo Cutello and Javier Montero. Hierarchies of aggregation operators. *International Journal of Intelligent Systems*, 9(11):1025–1045, 1994.

[158] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[159] Richard Jensen and Qiang Shen. Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. *IEEE Transactions on knowledge and data engineering*, 16(12):1457–1471, 2004.

[160] Srilatha Chebrolu, Ajith Abraham, and Johnson P Thomas. Feature deduction and ensemble design of intrusion detection systems. *Computers & Security*, 24(4):295–307, 2005.

[161] S Sakinah S Ahmad, S Ahmad, and Witold Pedrycz. Data and feature reduction in fuzzy modeling through particle swarm optimization. *Applied Computational Intelligence and Soft Computing*, 2012:10, 2012.

[162] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[163] Sheng Chen, Colin FN Cowan, and Peter M Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on neural networks*, 2(2):302–309, 1991.

[164] L-X Wang and Jerry M Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE transactions on Neural Networks*, 3(5):807–814, 1992.

[165] John Yen and Liang Wang. Simplifying fuzzy rule-based models using orthogonal transformation methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(1):13–24, 1999.

[166] Yeung Yam, Péter Baranyi, and Chi-Tin Yang. Reduction of fuzzy rule base via singular value decomposition. *IEEE Transactions on fuzzy Systems*, 7(2):120–132, 1999.

[167] Feng Wan, Huilan Shang, Li-Xin Wang, and You-Xian Sun. How to determine the minimum number of fuzzy rules to achieve given accuracy: a computational geometric approach to siso case. *Fuzzy Sets and Systems*, 150(2):199–209, 2005.

[168] Haiping Du and Nong Zhang. Application of evolving takagi–sugeno fuzzy model to nonlinear system identification. *Applied soft computing*, 8(1):676–686, 2008.

[169] Yong Zhang, Xiao-bei Wu, Zong-yi Xing, and Wei-Li Hu. On generating interpretable and precise fuzzy systems based on pareto multi-objective cooperative co-evolutionary algorithm. *Applied Soft Computing*, 11(1):1284–1294, 2011.

[170] Sean N Ghazavi and Thunshun W Liao. Medical data mining by fuzzy modeling with selected features. *Artificial Intelligence in Medicine*, 43(3):195–206, 2008.

[171] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3):483–519, 2013.

[172] Li-Xin Wang. The wm method completed: a flexible fuzzy system approach to data mining. *IEEE Transactions on fuzzy systems*, 11(6):768–782, 2003.

[173] J Arturo Olvera-López, J Ariel Carrasco-Ochoa, J Francisco Martínez-Trinidad, and Josef Kittler. A review of instance selection methods. *Artificial Intelligence Review*, 34(2):133–143, 2010.

[174] Elena Marchiori. Class conditional nearest neighbor for large margin instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):364–370, 2010.

[175] Salvador Garcia, Joaquin Derrac, Jose Cano, and Francisco Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 2012.

[176] Michela Antonelli, Pietro Ducange, and Francesco Marcelloni. Genetic training instance selection in multiobjective evolutionary fuzzy systems: A coevolutionary approach. *IEEE Transactions on fuzzy systems*, 20(2):276–290, 2012.

[177] Salvador García, José Ramón Cano, and Francisco Herrera. A memetic algorithm for evolutionary prototype selection: A scaling up approach. *Pattern Recognition*, 41(8):2693–2709, 2008.

[178] Ismael Rodríguez-Fdez, Manuel Mucientes, and Alberto Bugarín. An instance selection algorithm for regression and its application in variance reduction. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.

[179] Yoel Caises, Antonio González, Enrique Leyva, and Raúl Pérez. Scis: combining instance selection methods to increase their effectiveness over a wide range of domains. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 17–24. Springer, 2009.

[180] Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118:62–69, 2012.

[181] NicoláS GarcíA-Pedrajas, Aida De Haro-GarcíA, and Javier PéRez-RodríGuez. A scalable approach to simultaneous evolutionary instance and feature selection. *Information Sciences*, 228:150–174, 2013.

[182] Pei-Chann Chang, Chen-Hao Liu, and Yen-Wen Wang. A hybrid model by clustering and evolving fuzzy rules for sales decision supports in printed circuit board industry. *Decision Support Systems*, 42(3):1254–1269, 2006.

[183] Joaquín Derrac, Salvador García, and Francisco Herrera. Ifs-coco: Instance and feature selection based on cooperative coevolution with nearest neighbor rule. *Pattern Recognition*, 43(6):2082–2105, 2010.

[184] Joaquín Derrac, Isaac Triguero, Salvador García, and Francisco Herrera. Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(5):1383–1397, 2012.

[185] Yaochu Jin. Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement. *IEEE Transactions on Fuzzy Systems*, 8(2):212–221, 2000.

[186] Xin Fu and Qiang Shen. Fuzzy complex numbers and their application for classifiers performance evaluation. *Pattern Recognition*, 44(7):1403–1417, 2011.

[187] Shyi-Ming Chen, Yu-Chuan Chang, and Jeng-Shyang Pan. Fuzzy rules interpolation for sparse fuzzy rule-based systems based on interval type-2 gaussian fuzzy sets and genetic algorithms. *IEEE transactions on fuzzy systems*, 21(3):412–425, 2013.

[188] A Chong, TD Gedeon, and LT Kóczy. Projection based method for sparse fuzzy system generation. In *Proceedings of the 2nd WSEAS International Conference on Scientific Computation and Soft Computing, Crete, Greece*, pages 321–325, 2002.

[189] Z. Csaba Johanyák and S. Kovács. Sparse fuzzy system generation by rule base extension. In *Proceedings of 11th IEEE International Conference of Intelligent Engineering Systems (INES 2007), Budapest, Hungary*, pages 99–104. Citeseer, 2007.

[190] Dávid Vincze and Szilveszter Kovács. Reduced rule base in fuzzy rule interpolation-based q-learning. In *Proceedings of 10th International Symposium of Hungarian*

Researchers on Computational Intelligence and Informatics, November, pages 12–14, 2009.

[191] Dávid Vincze and Szilveszter Kovács. Incremental rule base creation with fuzzy rule interpolation-based q-learning. In *Computational Intelligence in Engineering*, pages 191–203. Springer, 2010.

[192] Ronald R Yager and Dimitar P Filev. Generation of fuzzy rules by mountain clustering. *Journal of Intelligent & Fuzzy Systems*, 2(3):209–219, 1994.

[193] U Kaymak and Robert Babuska. Compatible cluster merging for fuzzy modelling. In *Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE Int*, volume 2, pages 897–904. IEEE, 1995.

[194] Chun-Tang Chao, Young-Jeng Chen, and Ching-Cheng Teng. Simplification of fuzzy-neural systems using similarity analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(2):344–354, 1996.

[195] Magne Setnes, Robert Babuska, Uzay Kaymak, and Hans R van Nauta Lemke. Similarity measures in fuzzy rule base simplification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(3):376–386, 1998.

[196] Raghu Krishnapuram and Chih-Pin Freg. Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern recognition*, 25(4):385–400, 1992.

[197] Yaochu Jin. Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement. *IEEE Transactions on Fuzzy Systems*, 8(2):212–221, 2000.

[198] S. M. Chen. New methods for subjective mental workload assessment and fuzzy risk analysis. *Cybernetics and Systems*, 27(5):449–472, 1996.

[199] B Sridevi and R Nadarajan. Fuzzy similarity measure for generalized fuzzy numbers. *Int. J. Open Problems Compt. Math*, 2(2):242–253, 2009.

[200] S. J. Chen and S. M. Chen. Fuzzy risk analysis based on similarity measures of generalized fuzzy numbers. *IEEE Transactions on Fuzzy Systems*, 11(1):45–56, 2003.

[201] Shan Huo Chen and Chih Hsun Hsieh. Ranking generalized fuzzy number with graded mean integration representation. In *Proceedings of the Eighth International Conference of Fuzzy Sets and Systems Association World Congress*, volume 2, pages 551–555, 1999.

[202] Zhiheng Huang and Qiang Shen. A new fuzzy interpolative reasoning method based on center of gravity. In *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, volume 1, pages 25–30. IEEE, 2003.

[203] W.A. Farag, V.H. Quintana, and G. Lambert-Torres. A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems. *Neural Networks, IEEE Transactions on*, 9(5):756–767, 1998.

[204] C. C. Chen and C. C. Wong. Self-generating rule-mapping fuzzy controller design using a genetic algorithm. *Control Theory and Applications, IEE Proceedings -*, 149(2):143–148, 2002.

[205] Zsolt Csaba Johanyák and Olga Papp. A hybrid algorithm for parameter tuning in fuzzy model identification. *Acta Polytechnica Hungarica*, 9(6):153–165, 2012.

[206] Jinjun Tang, Fang Liu, Wenhui Zhang, Ruimin Ke, and Yajie Zou. Lane-changes prediction based on adaptive fuzzy neural network. *Expert Systems with Applications*, 91:452–463, 2018.

[207] Jorge Casillas, Oscar Cordón, Maria José Del Jesus, and Francisco Herrera. Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE Transactions on Fuzzy Systems*, 13(1):13–29, 2005.

[208] Sin-Jun Kang, Chun-Hee Woo, Hee-Soo Hwang, and Kwang B Woo. Evolutionary design of fuzzy rule base for nonlinear system modeling and control. *IEEE Transactions on Fuzzy Systems*, 8(1):37–45, 2000.

[209] Oscar Cord et al. *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*, volume 19. World Scientific, 2001.

[210] Plamen Angelov. An approach for fuzzy rule-base adaptation using on-line clustering. *International Journal of Approximate Reasoning*, 35(3):275–289, 2004.

[211] J-SR Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3):665–685, 1993.

[212] R Jayadevan, Satish R Kolhe, Pradeep M Patil, and Umapada Pal. Offline recognition of devanagari script: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):782–796, 2011.

[213] Debashis Mukherjee, Debotosh Bhattacharjee, and Mita Nasipuri. Curvature based localization of nose tip point for processing 3d-face from range images. In *Communications and Signal Processing (ICCSP), 2014 International Conference on*, pages 046–050. IEEE, 2014.

[214] Chin-Seng Chua, Feng Han, and Yeong-Khing Ho. 3d human face recognition using point signature. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 233–238. IEEE, 2000.

[215] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T Silva. Computing and rendering point set surfaces. volume 9, pages 3–15. IEEE, 2003.

[216] Xiaoguang Lu, Dirk Colbry, and Anil K Jain. Three-dimensional model based face recognition. In *null*, pages 362–366. IEEE, 2004.

[217] Kevin W Bowyer, Kyong Chang, and Patrick Flynn. A survey of approaches and challenges in 3d and multi-modal 3d+ 2d face recognition. *Computer vision and image understanding*, 101(1):1–15, 2006.

[218] Scott D Peckham. Profile, plan and streamline curvature: A simple derivation and applications. *Proceedings of Geomorphometry*, pages 27–30, 2011.

[219] Pinghai Yang and Xiaoping Qian. Direct computing of surface curvatures for point-set surfaces. *SPBG*, 7:29–36, 2007.

[220] David Levin. The approximation power of moving least-squares. *Mathematics of Computation of the American Mathematical Society*, 67(224):1517–1531, 1998.

[221] David Levin. Mesh-independent surface interpolation. In *Geometric modeling for scientific visualization*, pages 37–49. Springer, 2004.

[222] Nina Amenta and Yong Joo Kil. Defining point-set surfaces. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 264–270. ACM, 2004.

[223] Nina Amenta and Yong Joo Kil. The domain of a point set surface. In *SPBG*, pages 139–147, 2004.

[224] Tamal K Dey, Samrat Goswami, and Jian Sun. Extremal surface based projections converge and reconstruct with isotopy. *Tech. Rep. OSU-CISRC-05-TR25*, 2005.

[225] Tamal K Dey and Jian Sun. An adaptive mls surface for reconstruction with guarantees. In *Symposium on Geometry processing*, pages 43–52, 2005.

[226] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.

[227] Gady Agam and Xiaojing Tang. A sampling framework for accurate curvature estimation in discrete surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 11(5):573–583, 2005.

[228] Gabriel Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *iccv*, page 902. IEEE, 1995.

[229] Tatiana Surazhsky, Evgeni Magid, Octavian Soldea, Gershon Elber, and Ehud Rivlin. A comparison of gaussian and mean curvatures estimation methods on triangular meshes. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 1021–1026. IEEE, 2003.

[230] Ioannis Douros and Bernard F Buxton. Three-dimensional surface curvature estimation using quadric surface patches. *Scanning*, 44, 2002.

[231] Min Yang and Eungki Lee. Segmentation of measured point data using a parametric quadric surface approximation. *Computer-aided design*, 31(7):449–457, 1999.

[232] Ron Goldman. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design*, 22(7):632–658, 2005.

[233] Yao Tan, Hubert P. H. Shum, Fei Chao, V. Vijayakumar, and Longzhi Yang. Curvature-based sparse rule base generation for fuzzy rule interpolation. *Journal of Intelligent & Fuzzy Systems*, 36(5):4201–4212, May 2019.

[234] Serge Guillaume. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on fuzzy systems*, 9(3):426–443, 2001.

[235] Zong-Mu Yeh. A systematic method for design of multivariable fuzzy logic control systems. *IEEE Transactions on Fuzzy Systems*, 7(6):741–752, 1999.

[236] Hisao Ishibuchi and Takashi Yamamoto. Rule weight specification in fuzzy rule-based classification systems. *IEEE transactions on fuzzy systems*, 13(4):428–435, 2005.

[237] Q. Guo, Y. Qu, A. Deng, and L. Yang. A new fuzzy-rough feature selection algorithm for mammographic risk analysis. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 934–939, Aug 2016.

[238] Y. Qu, Y. Rong, A. Deng, and L. Yang. Associated multi-label fuzzy-rough feature selection. In *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, pages 1–6, June 2017.

[239] Q. Zhang, Y. Qu, A. Deng, and L. Yang. Hierarchical quotient spaces-based feature selection. In *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, pages 770–775, March 2018.

[240] Q. Sun, Y. Qu, A. Deng, and L. Yang. Fuzzy-rough feature selection based on lambda-partition differentiation entropy. In *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 1222–1227, July 2017.

[241] Yanpeng Qu, Guanli Yue, Changjing Shang, Longzhi Yang, Reyer Zwiggelaar, and Qiang Shen. Multi-criterion mammographic risk analysis supported with multi-label fuzzy-rough feature selection. *Artificial Intelligence in Medicine*, 100:101722, 2019.

[242] Yuhu Cheng, Xue Qiao, Xuesong Wang, and Qiang Yu. Random forest classifier for zero-shot learning based on relative attribute. *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

[243] Longzhi Yang, Zheming Zuo, Fei Chao, and Yanpeng Qu. Fuzzy interpolation systems and applications. In *Modern Fuzzy Control Systems and Its Applications*. InTech, 2017.

[244] Jie Li, Longzhi Yang, Yanpeng Qu, and Graham Sexton. An extended takagi–sugeno–kang inference system TSK+ with fuzzy interpolation and its rule base generation. *Soft Computing*, 22(10):3155–3170, 2018.

[245] Rahila H Sheikh, Mukesh M Raghuwanshi, and Anil N Jaiswal. Genetic algorithm based clustering: a survey. In *First International Conference on Emerging Trends in Engineering and Technology*, pages 314–319. IEEE, 2008.

[246] John Henry Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[247] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[248] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[249] Yao Tan, Jie Li, Martin Wonders, Fei Chao, Hubert PH Shum, and Longzhi Yang. Towards sparse rule base generation for fuzzy rule interpolation. In *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on*, pages 110–117. IEEE, 2016.

[250] David B Thomas, Wayne Luk, Philip HW Leong, and John D Villasenor. Gaussian random number generators. *ACM Computing Surveys (CSUR)*, 39(4):11, 2007.

[251] James Edward Baker. Adaptive selection methods for genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pages 101–111, 1985.

[252] Naren Naik, Ren Diao, and Qiang Shen. Genetic algorithm-aided dynamic fuzzy rule interpolation. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2198–2205. IEEE, 2014.

[253] Jayant G Rohra, Boominathan Perumal, Swathi Jamjala Narayanan, Priya Thakur, and Rajen B Bhatt. User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks. In *Proceedings of Sixth International Conference on Soft Computing for Problem Solving*, pages 286–295. Springer, 2017.

[254] H Tolga Kahraman, Seref Sagiroglu, and Ilhami Colak. The development of intuitive knowledge classifier and the modeling of domain dependent data. *Knowledge-Based Systems*, 37:283–295, 2013.

[255] Yang Long, Li Liu, Fumin Shen, Ling Shao, and Xuelong Li. Zero-shot learning using synthesised unseen visual data with diffusion regularisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[256] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.

[257] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010.

[258] Felix Yu, Liangliang Cao, Rogerio Feris, John Smith, and Shih-Fu Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013.

[259] Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010.

[260] Soravit Changpinyo, Wei-Lun Chao, and Fei Sha. Predicting visual exemplars of unseen classes for zero-shot learning. *ICCV*, 2017.

[261] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[262] Tinne Tuytelaars, Mario Fritz, Kate Saenko, and Trevor Darrell. The nbnn kernel. In *ICCV*, 2011.

[263] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *CVPR*, 2013.

[264] Yao Tan, Jie Li, Martin Wonders, Fei Chao, Hubert P. H. Shum, and Longzhi Yang. Towards sparse rule base generation for fuzzy rule interpolation. In *FUZZ*, 2016.

[265] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[266] Alireza Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[267] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning. *ICCV*, 2017.

[268] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015.

[269] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[270] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014.

[271] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *CVPR*, 2013.

[272] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016.

[273] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013.

[274] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016.

[275] Ziad Al-Halah, Makarand Tapaswi, and Rainer Stiefelhagen. Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In *CVPR*, 2016.

[276] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, 2011.

[277] Ziad Al-Halah and Rainer Stiefelhagen. How to transfer zero-shot object recognition via hierarchical transfer of semantic attributes. In *WACV*, 2015.

[278] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, 2016.

[279] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. *arXiv preprint arXiv:1605.04253*, 2016.

[280] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.

[281] Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

[282] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[283] Dajun Zhou, Minghui Shi, Fei Chao, Chih-Min Lin, Longzhi Yang, Changjing Shang, and Changle Zhou. Use of human gestures for controlling a mobile robot via adaptive CMAC network and fuzzy logic controller. *Neurocomputing*, 282:218 – 231, 2018.

[284] F. Chao, D. Zhou, C. Lin, L. Yang, C. Zhou, and C. Shang. Type-2 fuzzy hybrid controller network for robotic systems. *IEEE Transactions on Cybernetics*, pages 1–15, 2019.

Appendix A

Publications

Publications arising from this work

A few publications have been generated from the research carried out within the PhD project. Below lists the resultant publications that are in close relevance to the thesis, including all the papers already published.

1. Li, Jie, Yang, Longzhi, Shum, Hubert P. H., Sexton, Graham and Tan, Yao (2015) Intelligent Home Heating Controller Using Fuzzy Rule Interpolation. In: UKCI 2015 - UK Workshop on Computational Intelligence, 7th - 9th September 2015, Exeter, UK.
http://nrl.northumbria.ac.uk/28258/

2. Tan, Yao, Li, Jie, Wonders, Martin, Chao, Fei, Shum, Hubert P. H. and Yang, Longzhi (2016) Towards Sparse Rule Base Generation for Fuzzy Rule Interpolation. In: WCCI 2016 - IEEE World Congress on Computational Intelligence, 24th - 29th July 2016, Vancouver, Canada.
http://nrl.northumbria.ac.uk/27181/

3. Long, Yang, Tan, Yao, Organisciak, Daniel, Yang, Longzhi and Shao, Ling (2018) Towards Light-weight Annotations: Fuzzy Interpolative Reasoning for Zero-shot Image Classification. In: BMVC 2018 - British Machine Vision Conference, 3rd - 6th September 2018, Newcastle upon Tyne, UK.
http://nrl.northumbria.ac.uk/35747/

4. Yao Tan, Hubert Shum, Fei Chao, V. Vijayakumar and Longzhi Yang (2018) Curvature-based sparse rule base generation for fuzzy rule interpolation, in Journal of Intelligent & Fuzzy Systems.

Appendix B

Acronyms

AA	Arithmetic Averaging
CNF	Convex and Normal Fuzzy (Sets)
COR	Cooperative Rules
CRI	Compositional Rule of Inference
CSA	Clonal Selection Algorithm
DFRI	Dynamic Fuzzy Rule Interpolation
FCM	Fuzzy C-means
FL	Fuzzy Logic
FIS	Fuzzy Inference System
FRBS	Fuzzy Rule-Based System
FRI	Fuzzy Rule Interpolation
GA	Genetic Algorithm
GOWA	Generalised Ordered Weighted Average
FWA	Fuzzy Weighted Averaging
GPS	Global Positioning System
GZSL	Generalised Zero-Shot Learning
H	High
IOWA	Induced Ordered Weighted Averaging
KH	Koczy-Hirota
KNN	K-Nearest-Neighbour
L	Large
L	Low
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LFM	Linguistic Fuzzy Modeling

M Medium
MISO Multiple Input Single Output
MIMO Multiple Input Multiple Output
MLP Multiple Layer Perception (Neural Network)
MLS Moving Least Square
MKE Match Kernel Embedding
OAA Ordered Weighted Averaging
PCA Principal Component Analysis
PFM Precise Fuzzy Modelling
PHOG Pyramid Histogram of Oriented Gradients
PSR Prototype Selection by Relevance
RF Random Forest
RMSE Root Mean Squire Error
SIFT Scale Invariant Feature Transform
SISO Single Input Single Output
SL Simple Logistic
SMS Similarity Match Scores
SR Selection Reduction
SV Simile Vector
SVM Support Vector Machine
TS Tabu Search
TSK Takagi-Sugeno-Kang
VH Very High
VL Very Low
VLAD Vector of Locally Aggregated Descriptors
VSMS Vector of Similarity Match Scores
WAA Weighted Arithmetic Averaging
WE Word Embedding
WGA Weighted Geometric Averaging
WOSA Weighted Ordered Statistic Averaging
WOWA Weighted Ordered weighted Averaging
ZSL Zero-Shot Learning

Appendix C

Algorithms

Algorithm 1 Hierarchical partition & region selection

Inputs: T, the training dataset
n_1, the partition number for input variable x_1
n_2, the partition number for input variable x_2
θ, the curvature threshold
p, a threshold increasing ratio
$RR = \emptyset$, a set hosts the selected region but initialised as empty
Outputs: RR, the selected regions

procedure *Selection*(T, n_1, n_2, θ, p)
    RR'=GridPartition (T, n_1, n_2)
    for each R' *in* RR' **do**
        if $\theta \leq c_{R'} \leq \theta * (1+p)$ **then**
            $RR = RR \cup R'$
        end if
        if $c_{R'} \geq \theta * (1+p)$ **then**
            Selection$(R', n_1, n_2, \theta * (1+p), p)$
        end if
    end for
    return RR
end procedure

Algorithm 2 Hypercube Selection

Inputs: $\mathbb{H}$: the given set of hypercubes.
m: the required number of rules in the to-be-generated rule base, ie., the number of selected data instances.
Outputs: $\mathbb{H}'$: the selected important hypercubes.

procedure *HypercubeSelection*($\mathbb{H}$, m)
  for each $H_i \in \mathbb{H}$ **do**
    $H_i.weight = 0$
    for each $C_{ij} \in H_i$ **do**
      $H_i.weight \leftarrow H_i.weight + v_{ij}$
    end for
  end for
  $\mathbb{H}'' = Sort_{descending}(\mathbb{H})$
  $\mathbb{H}'$ = first m instances in $\mathbb{H}''$
end procedure

Algorithm 3 Feature Discrimination

Inputs: $\mathbb{H}'$: the selected set of hypercubes.
b: the number of selected input features.
T: the training dataset.
Outputs: $\mathbb{X}'$: a set of the significant features.

procedure *FeatureDiscrimination*($\mathbb{H}', b, T$)
  $\mathbb{R} = \emptyset$
  for $j = 1 \rightarrow c$ **do**
    Generate an artificial rule base R_j using all cubes C_{ij}
    $\mathbb{R} = \mathbb{R} \cup R_j$
    $p_j = FRI(T, R_j)$
  end for
  $\mathbb{R}' = Sort_{Ascending}(\mathbb{R}, p_j)$
  for $R_j \in \mathbb{R}'$ **do**
    $R_j.weight \leftarrow CalculateWeight(\mathbb{R}', R_j)$
  end for
  for each input dimension $x_k \in \mathbb{X}$ **do**
    $x_k.weight \leftarrow 0$
    for each $R_j \in \mathbb{R}'$ **do**
      if x_k is used by R_j **then**
      $x_k.weight \leftarrow x_k.weight + R_j.weight$
      end if
    end for
  end for
  $Sort_{Ascending}(\mathbb{X})$
  $\mathbb{X}'$ = first b features in $\mathbb{X}$
end procedure