Induction of Accurate and Interpretable Fuzzy Rules

Tianhua Chen

Supervisors: Prof. Qiang Shen
Dr. Changjing Shang

Ph.D. Thesis

Department of Computer Science
Institute of Mathematics, Physics and Computer Science
Aberystwyth University

November 27, 2017

Declaration and Statement

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ..... (candidate)

Date .....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Where **correction services**[1] have been used, the extent and nature of the correction is clearly marked in a footnote(s).

Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed ..... (candidate)

Date .....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate)

Date .....

[1] This refers to the extent to which the text has been corrected by others.

Abstract

Knowledge discovery from data with fuzzy modelling is currently an active research area in the field of computational intelligence. Fuzzy modelling describes systems by establishing relationships between input and output variables with fuzzy logic and fuzzy set theory. One of the main advantages of fuzzy modelling lies in the interpretability, such that they can formulate the knowledge with linguistic fuzzy rules to gain insights into behaviours of a complex system. However, the interpretability is not automatically given due to only using fuzzy rules. Unlike accuracy that can be used to objectively assess performance of the underlying system, interpretability is a subjective property that may be affected by a range of practical issues, especially regarding the representation of the underlying concepts and domain knowledge. Despite of no commonly accepted mechanism to adjudge interpretability, the incorporation of domain expertise encoded as predefined fuzzy sets is desirable to effectively interpret a fuzzy model. This facilitates enhanced transparency in both learning the models and the inferences performed with the learned models.

In light of this, the thesis is focused on the automatic generation of accurate and interpretable fuzzy models expressed as classification rules, where the use of fixed and predefined quantity spaces is a must for semantic interpretability. In this thesis, several approaches are presented with generated fuzzy rules being interpretable, and achieving competitive performance in comparison to state-of-the-art methods. These include: 1) the approach for the acquisition of fuzzy rules with quantifiers following class-dependent simultaneous rule learning strategy; 2) the approach for the acquisition of weighted fuzzy rules where heuristically generated fuzzy rules are initialised, followed by the global search of optimal rule weights; and 3) the approach that works by utilising existing crisp rules generated by a certain crisp rule-based learning classifier, and then performs rule mapping, followed by global genetic rule and condition selection. Furthermore, to enhance the capability of a fuzzy classifier, the thesis also develops a classifier ensemble approach based on the measure of nearest-neighbour-based reliability. Apart from benchmark data sets that have been utilised for systematic experimental verification, the proposed techniques are applied to a real-world problem of academic journal ranking, demonstrating the efficacy of the present research.

Acknowledgements

I would like to express my uttermost gratitude to my supervisors: Prof. Qiang Shen and Dr. Changjing Shang, for their motivation, guidance and efforts. Without them, this Ph.D. and thesis would never have happened.

I would like to thank all my fellow researchers, especially those from the Advanced Reasoning Group for their useful discussions and helpful advice. I am especially grateful to Dr. Jun He, Dr. Richard Jensen, Dr. Neil S. Mac Parthaláin, Dr. Ren Diao, Dr. Shangzhu Jin, Dr. Chengyuan Chen, Dr. Yongfeng Zhang, Dr. Liping Wang, Dr. Ling Zheng, Zhenpeng Li, Fangyi Li, Jing Yang, Pu Zhang, Tao Xu and Jingyuan Zhang. In particular, I am extremely grateful to Dr. Pan Su for his helpful advice, stimulating discussions and collaborative efforts.

I would like to express the appreciation to my landlord couple Paddy and Sue O'Brien for their tremendous help with my English. I would like to thank all my dear friends both in the United Kingdom and in China.

I am also extremely grateful to all of the academic, administrative, technical, and support staff at the Department of Computer Science, Aberystwyth University, for their kind assistance throughout my entire study.

My sincere gratitude goes to my entire family: my parents Wu Chen and Xiuer Huang, my dear wife Yuting Chen, and her parents Ruiliang Chen and Jiuying Fu. The completion of this Ph.D. would not have been possible without their kind support and encouragement.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

WITH the staggering development of computer technology and the rapid computerisation of business nowadays, huge volume of data is being accumulated and collected at a dramatic pace across a variety of fields. However, raw data is barely of direct interest unless potentially useful information is extracted. Knowledge discovery in databases (KDD) refers to the overall process of extracting useful high-level knowledge from low-level data, where data mining is a particular step among others such as data preparation, data preprocessing, evaluation and interpretation of mined results [110]. Being a computational process involving methods including artificial intelligence, machine learning [109], statistics, and database systems [86], data mining is generally about the application of specific algorithms for the extraction of interesting and useful information by analysing data in large databases [162].

Typical data mining approaches find patterns in relational databases, where rows correspond to objects to be analysed and columns represent values of properties or attributes of underlying objects. Depending on the nature of mining tasks, there are two main categories of data mining approaches. That is supervised learning that induces models from class-labelled data for prediction and classification, and unsupervised learning that induces interesting patterns from unlabelled data for exploratory data analysis. Traditional statistics also provides numerous data analysis, but may be prohibitive on very large data sets for their algorithmic complexity [51]. Instead many of the data mining methods are able to deal with very large data sets in a very efficient way. More importantly, apart from the analytical languages used in statistics, data mining methods also use other forms of formalisms, e.g.,

decision trees and rule sets, to present results of analysis in an appropriate and human understandable way.

Soft computing (often also referred to as Computational Intelligence) aims to provides inexact solutions to complex real-world problems where traditional mathematical modelling may be ineffective, e.g., NP-complete problems or problems being stochastic in nature. Various soft computing techniques have been developed and applied to handle different challenges stemming from data mining. Being a cornerstone of soft computing, fuzzy set theory (FST) [169, 141, 173] enables the tolerance of imprecision, uncertainty and approximation, where many problems in real-life cannot be handled with binary encoding. Fuzzy logic allowing for partial truth is the foundation to perform approximate reasoning, which is closer to human reasoning and aims to generate an inexact conclusion from inexact premises. Recalling that many products in recent years claim to be intelligent, which is more of a property attributed to human reasoning and decision making, demonstrating the efficacy of computational intelligence in general and fuzzy systems in particular for data mining.

The tools and methods that have been developed in the framework of FST have the potential to support all of the steps that comprise the process of KDD [67, 83, 85]. For example, fuzzy set-based techniques have been used in data selection and preparation phase to model vague data in terms of fuzzy sets [155], or to condense several crisp observations into a single fuzzy one [92]. They have also been developed for fuzzy extensions of certain well-known data mining methods without repeating the original methods themselves [67]. For instance, fuzzy cluster analysis [15] extends conventional clustering algorithms such as k-means that produces individual clusters separated by sharp boundaries assigning every object to one cluster in an unequivocal way. To overcome such boolean boundaries that are often not natural or even counterintuitive, fuzzy clustering smoothes the transition between different clusters, allowing an object belonging to differen clusters at the same time to various degrees. With motivations closely resembling clustering analysis, alternative data mining techniques (e.g., association rule mining and decision tree induction) are also softened using fuzzy sets to avoid certain undesirable threshold effects.

Among those, fuzzy rule-based systems (FRBSs) [6, 89, 36] are one of the most important applications of FST in data mining [102, 97]. A fuzzy rule-based model consists of a set of fuzzy rules in the form of if-then statement “IF (antecedent), THEN (consequent)”. The if-then statements specify what actions or behaviour should be

taken under given circumstances, providing a means to incorporate and formulate human knowledge. This model structure is appealing in the sense that it is user-friendly and intuitively reflects natural human thinking. In general, fuzzy systems have been applied to a number of engineering and science areas [146, 145, 140], e.g., in bioinformatics, control engineering, finance, medicine, robotics, and pattern recognition.

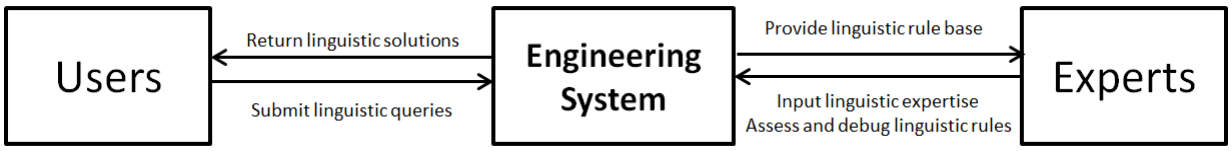

Figure 1.1: Fuzzy rule-based engineering system

In designing and implementing engineering systems, apart from the information obtained from physical sensory measurements and precise mathematical laws, information from experts' descriptions in terms of natural languages is often required to be utilised. Fuzzy systems [117, 133, 138] allow combining both types of information effectively into the system design, in an effort to precisely characterise key features of the engineering systems, and to allow for trackable mathematical and computational analysis.

The roles different entities play in a fuzzy rule-based engineering system may be summarised using Figure 1.1. That is, the fuzzy rule-based engineering system should be able to provide linguistic solutions to domain users, who are not necessarily experts in fuzzy systems. Provided with interpretable and explainable IF-THEN statements in natural languages, domain experts are able to verify the knowledge returned by the rule base, which is constructed on top of their agreed principles and knowledge encoded as domain knowledge. In the meanwhile, the fuzzy rule-based system provides linguistic explanations in natural languages to help users easily understand the domain problem and their solutions.

Being able to deal with vague concepts that are fundamental to natural languages in practical reasoning and decision making, fuzzy rule-based systems facilitate the foundation of knowledge in an intuitive way to both end-users with explainable solutions and experts with transparent insights into the complex systems. The aim of this thesis is therefore focused on the induction of accurate and interpretable fuzzy rules.

1.1 Interpretability Issues of Fuzzy Systems

As indicated previously, one of the most important advantages of fuzzy systems lies in their inherent interpretability as they support the explicit formulation of, and inference with, domain knowledge, gaining insights into the complex systems and facilitating the explanation of their operations. However, unlike criteria such as accuracy that can be used to precisely and objectively measure how good a fuzzy model is with respect to the real system, interpretability is a subjective property, which largely depends on the person who makes the assessment. Due to the subjective nature, interpretability may be affected by a range of practical issues, especially regarding the representation of the underlying concepts and knowledge in the problem domain. Different approaches [8, 108, 172, 84] have been proposed to study interpretability within the general area of fuzzy systems. Although there is still no commonly accepted mechanism to adjudge interpretability, complexity-based and semantics-based methods are typically considered when designing a fuzzy system. Complexity-based interpretability aims to reduce the complexity of a fuzzy model in terms of the number of rules and the number of labels per rule. Semantics-based interpretability aims to preserve the semantics of the membership functions (MFs), such that the fuzzy rules make use of meaningful linguistic labels.

The incorporation of intuitive expert knowledge into linguistic rules through the use of predefined fuzzy sets is desirable to effectively interpret a fuzzy model. This allows for enhanced transparency in both the learned models themselves and the inferences performed by running the learned models [28, 172, 107]. For many real-world applications (e.g., medical diagnosis [142, 114] and intelligence data analysis [101, 144]), the use of a fixed and predefined quantity space per variable is indeed a must. Subsequently, semantic constraints over MFs are often imposed in order to modify the definition of the fuzzy sets [108]. This may help improve the accuracy of the resulting learned model, but such computation may adversely affect the exactly prescribed meaning of the given labels and therefore, the interpretability of the overall rule model that employs such modified linguistic labels. Using domain expertise also makes it easier for experts to verify the obtained knowledge with fuzzy systems, unlike black-box systems such as neural networks [164] that can achieve high performance, but their solutions are difficult to explain.

For example, a masters' student performance p in the UK higher education system can be considered *low* if their score is below 50 (i.e., *fail* if $p < 50$), *medium* if

between 50 and 60 (i.e., *pass* if $50 <= p < 60$), *high* if between 60 and 70 (i.e., *merit* if $60 <= p < 70$), *very high* if the score is greater than or equal to 70 (i.e., *distinction* if $p >= 70$). These definitions have been developed by education experts and accepted by students and parents for a long period of time. The standard of distinction should not be changed simply because no students from a certain module can obtain very high scores in a single exam. Therefore, in situations where the majority has universally agreed on the understanding of certain notations, re-defining these concepts based on limited samples would lead to misleading conclusions. When no students have achieved sufficient scores for distinction, the standard should not be scaled just to fit those that achieve relatively high scores to give them a "distinction" award. The conclusion should be the exam is either too difficult or the batch of the students participating the exams have relatively weak background. Similar situation could also exist when judging whether the blood glucose level of a patient is high or not, where there are agreed principles that are based on long-term medical research, which does not come from the glucose distribution of a certain experiment.

In the above case, it is essential that universally agreed knowledge from a certain problem domain is *a prior* incorporated into system design. The labelled fuzzy terms only make sense if the underlying definitions are consistent with people's commonly held notations. This leads to the induction of fuzzy rule-based systems where domain expertise in terms of predefined fuzzy sets are required to incorporate and remain unchanged. That means an iterative approach that induces a rule base utilising knowledge from the database in the current iteration and then, alters the definitions of the database by sending feedback from the newly built rule base is not feasible if *a prior* incorporated domain expertise is required later. The induction of a rule base in a fuzzy system is independent of the acquisition of the data base that specifies the definitions of fuzzy sets for each variable and that is assumed *a prior* available.

1.2 Research Objectives and Contribution

Given the high desirability to incorporate domain expertise for the interpretation of a fuzzy rule model, the main work of the thesis is therefore, the automatic generation of accurate and interpretable fuzzy classification rules, where the use of fixed and predefined quantity space is a must for semantic interpretability. Owing to the use of fixed quantity space that comes from either domain expertise or static fuzzy set

definitions, the resulting fuzzy rule base is likely to suffer from performance loss, especially when distribution of the underlying training instances does not follow the pre-specified and fixed fuzzy set definitions. The aim of the thesis is thus, focused on exploring alternative means such that the resulting fuzzy rule base is able to achieve satisfactory performance whilst reflecting domain expertise through the use of fixed and predefined fuzzy sets.

In particular, the objective of the thesis is to develop methods that could induce accurate and interpretable fuzzy rules from the following three different perspectives:

1. To address the aforementioned problem of performance loss due to the use of fixed quantity space, work is proposed to utilise certain weighting schemes such that the accuracy of the resulting fuzzy rule base may be improved by adjusting the significance of certain fuzzy system components without disrupting the definitions of the underlying fuzzy sets.

2. When fixed and predefined fuzzy sets are used to partition the input space, the combination of all input and output variable values is likely to lead to the problem of "curse of dimensionality" as the number of input feature increases. An alternative research route in the thesis is to utilise a set of existing crisp rules generated by a certain data-driven crisp rule-based learning mechanisms. This is inspired by the observation that data-driven learning mechanisms are able to omit the empty parts of the input space and focus on places covered by existing training data. Making use of a set of data-driven crisp rules is able to give a head start to a potential fuzzy classifier where the incorporation of fixed and predefined fuzzy sets is a must.

3. Apart from the systematical verification with benchmark data sets, another important goal of the thesis is to apply the proposed techniques into real world problem. In particular, the problem of academic journal ranking is taken a case study here, due to its increasing popularity and significance in the assessment of research output quality.

Following on the initial research objectives as specified above, the thesis has made contribution from the following aspects with all initial research goals achieved:

- A weighting scheme by optimising weights at fuzzy rule level with Particle Swarm Optimisation is proposed, which has achieved statistically significant performance over rule bases without employing such optimised weights. The generated fuzzy rule base is also competitive with popular fuzzy classifiers. The proposed approach has been published in the 14th annual UK Workshop on Computational Intelligence and the journal of Soft Computing.

- An alternative weighting scheme is proposed such that the significance of different individual input features could be revealed by optimising fuzzy quantifiers attached to linguistic variables. The resultant quantified fuzzy rules demonstrate statistically significant performance over rule bases without employing such fuzzy quantifiers. The proposed approach has been published in the 2015 IEEE International Conference on Fuzzy Systems.

- The third approach developed by the thesis utilises a set of existing data-driven crisp rules, which are then transformed into fuzzy rules employing only fixed and predefined fuzzy sets, followed by local rule selection and global genetic optimisation. The resultant fuzzy rules achieve performance superior or competitive to a number of state-of-the-art fuzzy and non-fuzzy classifiers. Furthermore, this approach has been applied to the real world scenario of academic journal ranking problem, demonstrating the transparency of the resulting fuzzy rules and the efficacy of the proposed approach. This work is currently under review for journal publication.

- Apart from fulfilling all pre-specified research objectives, the thesis further develops a classifier ensemble approach to enhance the performance of an existing fuzzy classifier. This is achieve by the incorporation of nearest-neighbour-guided reliability assessment, such that a reduced subset of base classifiers is selected with minimal performance loss, thereby reducing overall running time overheads. The approach has been published in the 2017 IEEE International Conference on Fuzzy Systems.

1.3 Structure of Thesis

This section outlines the structure of the remainder of this thesis. Figure 1.2 illustrates the relationships between the individual chapters (other than the introduction, the

conclusion and the appendices). The direct dependencies between the chapters are denoted using solid arrows. In a nutshell, Chapter 2 provides the background knowledge of fuzzy systems together with the review of relevant literature; Chapter 3 presents the first weighting scheme by optimising weights at the rule level; Chapter 4 presents an alternative weighting scheme to learn quantified fuzzy rules; Chapter 5 then follows the second research route by transforming existing crisp rules into accurate and interpretable fuzzy rules; Chapter 6 applies the technique developed in Chapter 5 into the academic journal ranking problem; Chapter 7 further develops an ensemble approach to enhance performance of an existing fuzzy classifier.

Chapter 2: Background

This chapter provides the background introduction to the thesis. Specifically, a formal introduction of fuzzy sets, fuzzy logic, fuzzy rules and fuzzy modelling will be presented that are the basis of fuzzy rule induction. In addition, the chapter gives a brief introduction to evolutionary algorithms, including genetic algorithms and particle swarm optimisation. Both have been utilised as the optimisation techniques to search for optimal fuzzy rules in the thesis. This chapter also introduces a range of approaches for the induction of an FRBCS in the literature. These include fuzzy decision tree-based approaches, fuzzy association rule-based classifiers, both of which are capable of learning fuzzy rules directly from data with fixed and predefined fuzzy sets. The reviewed literature also covers a number of hybrid methods that combine fuzzy system design with evolutionary algorithms, known as evolutionary fuzzy systems.

Chapter 3: Induction of Weighted Fuzzy Rules with Particle Swarm Optimisation

To deal with situations where behaviour of the engineering system is readily available with experts' linguistic descriptions, this chapter proposes an approach to enhance performance of the existing engineering system depicted with a fuzzy rule base using fixed quantity spaces. This means that an initial rule fuzzy rule-base has been built with predefined fuzzy sets, which are required to be maintained for the purpose of consistent interpretability, both in the learned models and in the inference

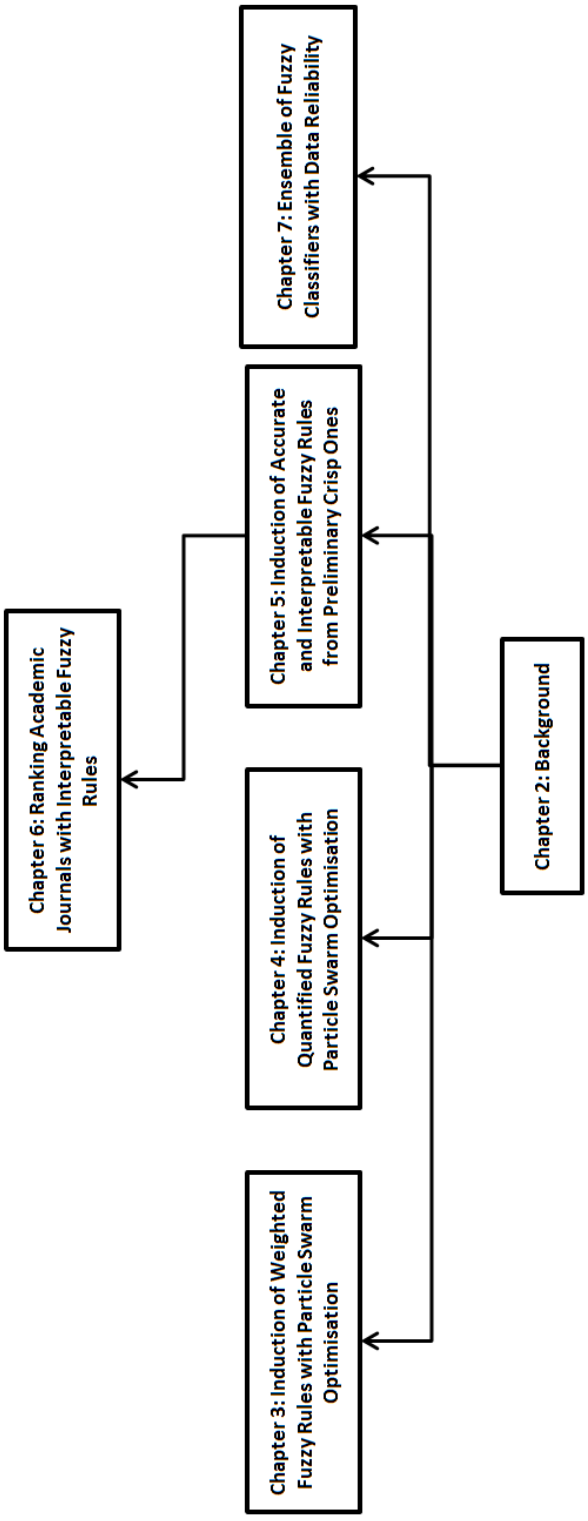

Figure 1.2: Relationships between thesis chapters

results using such models. Compared to those modifying antecedent fuzzy sets in the literature, which not only affects natural meanings of underlying concepts, but involves the learning of a number of parameters for each membership function, rule weight adjustment does not touch the definitions of predefined expertise and is less complex with only one single parameter per rule to learn while gaining the performance improvement for the existing rule base.

Chapter 4: Induction of Quantified Fuzzy Rules with Particle Swarm Optimisation

In the case of utilising experts' domain knowledge in terms of linguistic descriptions towards building an engineering system, the performance of such a system may not be satisfactory. This is because the possibly coarse descriptions may not able to precisely capture the exact characteristics of the underlying problems. This chapter proposes a weighting scheme with fuzzy quantifiers to adapt the interactions and relationships of individual domain features that are not seen equally important in contributing to a certain behaviour. Instead of using crisp weights with fuzzy terms, which may lead to confusion regarding the linguistic interpretation, the use of fuzzy quantifiers to modify the linguistic terms helps build fuzzy systems in a more natural way, ensuring that the inferred results remain consistent in the fuzzy representation adopted.

Chapter 5: Induction of Accurate and Interpretable Fuzzy Rules from Preliminary Crisp Representation

To retain the exactly prescribed meaning of given labels and hence, the interpretability of the overall rule models, the methods of generating fuzzy models utilising fixed and predefined quantity spaces may lead to the problem of curse of dimensionality as the input of input features increases. However, a data-driven rule generation method should omit the empty parts of the input space. Motivated by this observation, this chapter proposes a novel approach to generating interpretable fuzzy classification rules. For a given classification problem, simple crisp rules are utilised for initialisation, with each of them pointing to the model sub-spaces where desirable

fuzzy rules potentially exist. This is followed by a heuristic mapping procedure that converts each preliminary crisp rule into a set of interpretable fuzzy rules involving only the predefined fuzzy sets, thereby maintaining semantic interpretability. A local rule selection method is then performed to obtain a compact subset of initially mapped fuzzy rules that jointly generalise the capability of the underlying crisp rule. A fine grain tuning of all selected subsets of fuzzy rules is finally carried out with a conventional GA, resulting in an accurate and interpretable fuzzy rule-based classifier with a simplified structure.

Chapter 6: Journal Ranking using Induced Interpretable Fuzzy Rules

Given the promising results achieved with fuzzy rules generated from Chapter 6 on benchmark data sets, this chapter applies the proposed approach to the real-world journal ranking problem, which is of practical importance to research quality assessment in general and academic research output evaluation in particular.

Chapter 7: Ensemble of Fuzzy Classifiers with Data Reliability

For a general engineering system consisting of multiple sub-systems, each of them has specific views on the problem domain and may make varied predictions for certain samples. However, the synergetic cooperation of such multiple entities usually outperform than any individual one. Yet, certain sub-systems, which may be outdated or malfunctioned, are considered unreliable and should be discarded, for they are likely to generate false or biased predictions. Making use of the detectable trends that may emerge from local data structures, the method introduced in this chapter measures the reliabilities of individual system components by calculating similarities of each individual with regard to its neighbours. Being able to deliver statistically equivalent performance, the reduced system is with a much smaller cardinality, relieving space requirement and when implemented, making overall engineering process more efficient.

Chapter 8: Conclusion

This chapter summarises the key contributions made by the thesis, together with a discussion of topics which form the basis for future research.

Appendices

Appendix A lists the publications arising from the work presented in this thesis, containing both published papers, and one which is currently under review for journal publication. Appendix B provides information regarding the benchmark data sets employed in the thesis. Appendix C summarises the abbreviations employed throughout the thesis.

Chapter 2

Background

Fuzzy set theory enables the tolerance of imprecision, uncertainty and approximation, where many problems in real-life cannot be handled with binary encoding to model. Fuzzy logic allowing for partial truth is the foundation to perform approximate reasoning, which is closer to human reasoning and aims to generate an inexact conclusion from inexact premises. The induction of fuzzy rule-based systems is appealing in the sense that fuzzy systems support the combination of those obtained from physical sensory measurements and information from experts' descriptions in terms of natural languages, while outputing interpretable knowledge again in natural languages for the transparent insights into the behaviour of a complex system.

In the fuzzy systems literature, there are many approaches that have been proposed for the induction of a fuzzy rule-based classification system. The aim of this chapter is to review those methods that induce interpretable fuzzy rules utilising fixed and predefined fuzzy sets reflecting domain expertise. Before presenting the review, technical details of a typical fuzzy rule base are decomposed and introduced for better understanding of the work. The remainder of this chapter is structured as follows. Section 2.1 introduces a number of important concepts in order to build a fuzzy rule base, together with a literature review on well-known methods that directly learn fuzzy rules with fixed and predefined fuzzy sets. Section 2.2 introduces relevant paradigms of evolutionary algorithms, the combination of which with fuzzy rule induction forms a popular hybrid approach in the recent literature as evolutionary fuzzy systems. The induction of evolutionary fuzzy systems will be further reviewed in Section 2.3 before summarising this chapter in Section 2.4.

2.1 Fuzzy Sets and Systems

This section gives a detailed preliminary knowledge that is fundamental for subsequent fuzzy rule induction, as well as a review of well-known approaches in literature that induces interpretable fuzzy rules.

2.1.1 Prerequisite

This section introduces the basis of a fuzzy rule-base system. In particular, fuzzy set theory and fuzzy logic are first introduced as the fundamental building blocks of approximate reasoning, before the presentation of how observations match against individual fuzzy rules, and how conclusions are derived from the fuzzy rule base.

2.1.1.1 Fuzzy Sets

Fuzzy sets introduced by Zadeh [169] are sets whose elements have degrees of memberships between 0 and 1. Fuzzy sets can be seen as an extension to classical set, whose elements can either belong to (with membership degrees of 1) or not belong to the set (with membership degrees of 0). By contrast, fuzzy sets allow gradual assessment of the membership of elements in the a set. The idea of fuzzy set is based on the premise that change in the real world is not catastrophic but gradual, thus widely used in a wide range of domains where information is uncertain or imprecise, such as medical diagnosis [134].

In particular, a fuzzy set is defined by a membership function. For instance, fuzzy set A can be defined as the function of $\mu_A(x) : R \rightarrow [0,1]$ that maps crisp value $x \in R$ to the value of $[0,1]$. Crisp set can be seen as a special type of fuzzy set valued in $\{0,1\}$. Frequently used membership functions to define the fuzzy set include triangular, trapezoidal and gaussian function. For example, the fuzzy set may be defined as a triangular membership function μ_{tri} according to Figure 2.1 as follows:

$$\mu_{tri}(x) = \begin{cases} 0, & \text{if } x \leq a \\ \dfrac{x-a}{b-a}, & \text{if } a < x \leq b \\ \dfrac{c-x}{c-b}, & \text{if } b < x \leq c \\ 0, & \text{if } x > c \end{cases} \tag{2.1}$$

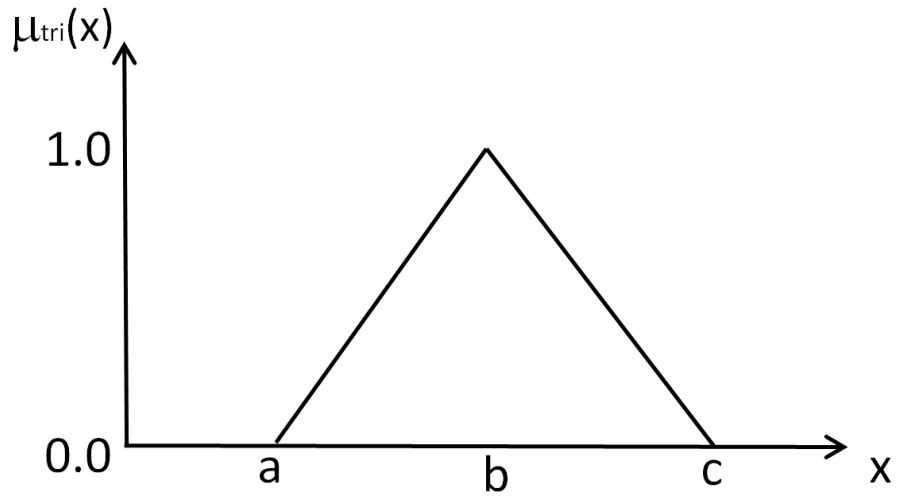

Figure 2.1: Triangle membership function

2.1.1.2 Fuzzy Logic

Logic is the study of methods and principles of reasoning, which is about generating new propositions from existing ones. Fuzzy logic generalises classical two-valued logic to be a real number in the interval of $[0, 1]$. This lays the foundation to perform approximate reasoning. In order to make deductive inferences, inference rules must be used. Inferences rules are various forms of tautologies which are logic formulas that are always true regardless of the truth values of atomic propositions. Three fundamental principles in fuzzy logic have been proposed in literature in order to perform approximate reasoning. These are *generalised modus ponens*, *generalised modus tollens*, and *generalised hypothetical syllogism*.

For instance, *generalised modus ponens* states the rule of getting the new fuzzy value B', given the fuzzy set A' and fuzzy relation R : If x is A, Then y is B shorthanded as $A \rightarrow B$. Formally, this can be defined as follows:

$$B' = A' \circ R_{A \rightarrow B} \tag{2.2}$$

where $\circ$ signifies the composition operation. Or,

$$\mu'_B(y) = sup_{x \in U}[\mu'_A(x) \star \mu_{A \rightarrow B}(x, y)] \tag{2.3}$$

where $\star$ is the t-norm operator and U refers to the universe of discourse.

Generalised modus ponens states the principle of inferring the conclusion based on the observation and a single fuzzy rule (relation). However, a practical fuzzy

rule base would contain more than one rule to make it work. Given a set of K fuzzy rules $R_j, j = 1, 2, \cdots, K$, there are two ways to perform deductive inference, i.e., composition based inference and individual-rule based inference.

Composition based inference combines all fuzzy rules in the rule base into a single fuzzy relation, which can be viewed a single fuzzy production rule. The principle of *generalised modus ponens* can then be performed for the combined fuzzy relation the same way it is used for single fuzzy rule. In particular, the often used operator chosen for combining the rules in literature is the union operator. This is based on the argument that individual rules should be treated as independent conditional statements, therefore the combined fuzzy relation should be defined as:

$$\mu_R(x, y) = \cup_{k=1}^{K} \mu_{R_k}(x, y) \tag{2.4}$$

where $\cup$ is the s-norm operator. The output of the fuzzy inference procedure given fuzzy set A' is then calculated as

$$\mu'_B(y) = sup_{x \in U}[\mu'_A(x) \star \mu_R(x, y)] \tag{2.5}$$

On the other hand, individual-rule based inference determines an output for each fuzzy rule in the rule base and then combine those outputs together. Similarly, the often adopted combination operator is the union operator. Therefore the output of the fuzzy inference procedure given fuzzy set A' is defined as

$$\mu'_B(y) = \cup_{k=1}^{K} \mu_{B'_k}(y) \tag{2.6}$$

where $\mu_{B'_k}(y), k = 1, 2, \cdots, K$ is as follows:

$$\mu_{B'_k}(y) = sup_{x \in U}[\mu_{A'_k}(x) \star \mu_{R_k}(x, y)] \tag{2.7}$$

2.1.1.3 Fuzzy Production Pules

Approximate reasoning is a process where a possibly inexact conclusion is inferred from a collection of inexact premises. Central to approximate reasoning is fuzzy relation and fuzzy relation composition. A fuzzy relation R is defined in the cartesian product of crisp sets $X_1, X_2, \cdots, X_n$ as a fuzzy set

$$\begin{aligned} R = \{((x_1, x_2, \cdots, x_n), \mu_R(x_1, x_2, \cdots, x_n))| \\ (x_1, x_2, \cdots, x_n) \in X_1 \times X_2 \times \cdots \times X_n\} \end{aligned} \tag{2.8}$$

where $\mu_R(x_1, x_2, \cdots, x_n)) \in [0, 1]$. Approximate reasoning is built on a collection of fuzzy production rules, which provide a formal approach to represent domain knowledge obtained from empirical experiences. For instance, a fuzzy rule may be defined as follows:

$$\text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2, \text{ Then } y \text{ is } B \tag{2.9}$$

This rule builds a relationship between premise variables x_1, x_2 and the consequent variable y. A direct interpretation of such fuzzy rule is the translation as a fuzzy relation. This is due to the fact the statement " *If* x, *Then* y" is written an implication $x \rightarrow y$ in case of classical propositional logic. Similarly, in the case of fuzzy rule, where classical (compound) propositions are replaced with fuzzy propositions, the fuzzy statement can also be interpreted as an implication. As $x \rightarrow y$ is equivalent to $\bar{x} \vee y$ or $(x \wedge y) \vee \bar{x}$ with same truth values, the specific interpretation of fuzzy statement may vary for a variety choices of fuzzy complement, fuzzy union, and fuzzy intersection operators. Supported with the argument that fuzzy production rules are local, Mamdani implications are the most widely used implications in fuzzy systems and fuzzy control [158]. In particular, the fuzzy production rule is interpreted as a fuzzy relation R_{MM} or R_{MP} in $X \times Y$ with the membership function

$$\mu_{R_{MM}} = min(\mu_X(x), \mu_Y(y)) \tag{2.10}$$

or

$$\mu_{R_{MM}} = \mu_X(x)\mu_Y(y) \tag{2.11}$$

where μ_X is the fuzzy set that describes the antecedent compound fuzzy proposition. A compound fuzzy proposition x is X is a composition of atomic fuzzy propositions connected with the connective and operator. In case of Rule 2.9, x_1 is A_1 and x_2 is A_2 are atomic fuzzy propositions valued in $[0, 1]$. Therefore, Rule 2.9 can be interpreted with the membership function as follows:

$$\mu_R(x_1, x_2, y) = min(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \mu_B(y)) \tag{2.12}$$

where the connective "and" in antecedent conditions is interpreted as min operator.

2.1.1.4 Fuzzy Rule-based Modelling

A fuzzy rule-based model is composed of a set of fuzzy rules in the form of if-then statement "IF (antecedent), THEN (consequent)". The if-then statements specify

what actions or behaviour should be taken under the circumstances, providing a means to incorporate and formulate human knowledge. This model structure is appealing in the sense that it is user-friendly and intuitively reflects natural human thinking. Depending on how the consequent is represented, fuzzy rule-based systems can be categorised into TSK fuzzy model [150] and Mamdani fuzzy model [103].

A TSK fuzzy rule is of the following form:

$$\text{If } x_1 \text{ is } A_1, x_2 \text{ is } A_2, \text{ Then } y = f(x_1, x_2) \tag{2.13}$$

where x_1 and x_2 are input variables, y is the output variable; A_1 and A_2 and B are fuzzy sets to describe the corresponding variables. Specifically, the consequent y is a crisp function represented as a polynomial in the input variable x_1 and x_2. A first order polynomial of the example can be:

$$\text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2, \text{ Then } y = px_1 + qx_2 + r \tag{2.14}$$

TSK models are computationally efficient and work well with optimisation and adaptive techniques. TSK fuzzy models have shown to be universal approximators in the sense that they are able to approximate any smooth nonlinear functions to any degree of accuracy in any convex compact region [48]. This provides a theoretical foundation of using TSK models to approximate complex nonlinear systems, and therefore they have been widely used in control problems [119], particularly for dynamic nonlinear systems.

One the other hand, a typical Mamdani fuzzy rule is defined as follows:

$$\text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2, \text{ Then } y \text{ is } B \tag{2.15}$$

Unlike the polynomial function used to calculate the consequent in Eqn. 2.13, the consequent y takes the value of B that is also a fuzzy set and can be attached with a linguistic label. A practical example of such rule may be: *If color is green and size is small, Then the tomato is unripe*. The fuzzy set *green* and *small* to describe tomato color and size respectively provide an interface between a numerical value and a symbolic description in terms of linguistic terms.

Unlike TSK models that describe rule consequent with a crisp function, a Mamdani rule consequent is also made of a descriptive fuzzy set that can be attached with a linguistic label. Due to the relatively simple structure and the interpretable nature

of Mamdani fuzzy rules, they have been more widely used than TSK rules which allow for more parameters to tune in the consequent [47]. This is also consistent with one of the most important incentives of introducing fuzzy sets for modelling complex systems that they can formulate the knowledge extracted from data with a more transparent way to gain insights into the complex systems [172]. The main aim of the thesis is therefore focused on the construction of Mamdani fuzzy rule-based systems, specifically, the fuzzy rule-based classification system (FRBCSs) where the rule consequent is crisp and discrete. An FRBCS can been as a special type of Mamdani fuzzy model with rule consequent being a singleton fuzzy set.

2.1.2 Fuzzy System Architecture

A key contribution of fuzzy systems is to formulate human knowledge in a systematic manner, together with information coming from sensory measurements and mathematical models. The transformation made by fuzzy systems makes it possible to map human knowledge onto mathematical formulas for systematic analysis. In order to better understand specifically what part the thesis is focused on, this section gives a summary of previously introduced building blocks by outlining the general architecture of a fuzzy rule-based system.

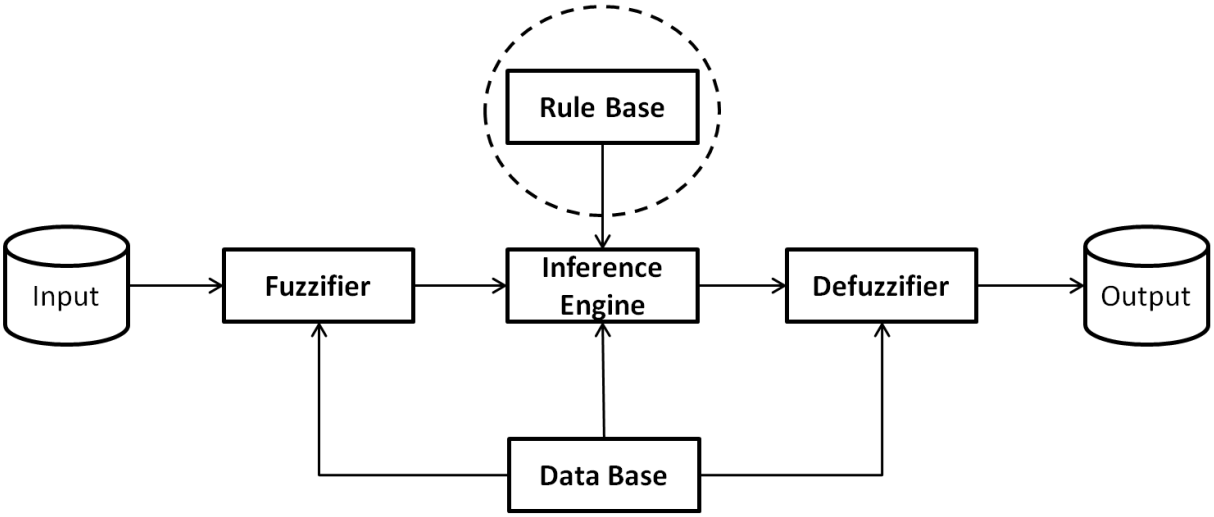

Figure 2.2: Architecture of a fuzzy system

There are five components in a classical knowledge-based fuzzy system as shown in Figure 2.2. These include the fuzzifier, the inference engine, the defuzzifier, the data base and the rule base.

- The *fuzzifier* provides an interface to transform crisp inputs that are usually obtained from physical measurement or derived from mathematical laws into fuzzy sets. The fuzzifier makes it possible to represent crisp numerical values in terms of linguistic words in natural languages with precise mathematical definitions. The transformation is implemented by searching through a collection of semantic mappings that relates a crisp value with a set of predefined fuzzy sets.

- The *inference engine* performs operations on the fuzzy values that are passed from the fuzzifier. These fuzzy values of individual features fire fuzzy rules stored in the rule base. Firing or matching degree with regard to each fuzzy rule is calculated by applying fuzzy logical operators on membership values of existing conditional antecedents. In case of multiple fuzzy rules fired simultaneously, these individual outputs are combined and aggregated before passing onto the defuzzifier.

- The *defuzzifier* maps an aggregated fuzzy set from the inference engine into a crisp set for subsequent operation. Often the required return for engineering systems should be in form of real-valued outputs, same as that of input. Therefore, the combined fuzzy set is mapped back into a crisp output through the defuzzifier. A general idea of deriving such crisp value from a fuzzy set is to find the representative point of a fuzzy set, e.g., centroid, maximum, etc.

- The *data base* stores the definitions of crisp values with regard to a number of overlapping concepts defined as fuzzy membership functions for individual attributes. The data base provides the mapping for converting crisp values into fuzzy values for subsequent computation and converting fuzzy values back to crisp values as engineering instructions for system operation. On top of the definitions of membership functions it stores a set of linguistic labels that are known to common human users and are fundamental in reasoning and decision making.

- The *rule base* stores a collection of linguistic fuzzy rules that are central to the fuzzy system in the sense that all other components are used to implement the rules in an efficient and systematic manner. The structure of fuzzy rules together with the inference procedure form the computation mechanism where fuzzy set theory and fuzzy logic are employed. Traditionally, the construction

of a fuzzy rule base may be done by directly consulting domain experts who can explicitly express their domain expertise by if-then statements. However, in case of very complex problems or where domain expertise is not sufficient, the information available may only be the input-output pairs. The thesis is therefore focused on the induction of a fuzzy rule base from a collected set of input-output data pairs.

2.1.3 Induction of Fuzzy Rule-based Systems

This section introduces the principles of two of the most commonly seen fuzzy rule induction methods, i.e., fuzzy decision tree and fuzzy association rule-based classifiers. As the thesis is working on the induction of interpretable fuzzy rules, where fixed and predefined fuzzy sets are utilised reflecting domain expertise as argued in Chapter 1, only the relevant literature that also holds onto this standpoint is reviewed.

Depending on how predefined membership functions of corresponding fuzzy sets are generated, this thesis categorises them into three cases.

1. Fuzzy sets that are defined by consulting domain experts. This is preferred given the original motivation of incorporating human knowledge into system design. However, it will have to take extra work consulting with domain experts, who may sometimes not be available.

2. Fuzzy sets that are uniformly partitioned in the universe of discourse. In case of no domain expertise available, membership functions could be built by dividing the universe of discourse into several equal partitioned intervals. This has also been considered being most interpretable from the shape point of view, as it simultaneously satisfies the properties [108] of normality, distinguishability, continuity, etc. Many interpretability indexes are constructed by measuring the difference between the standard fuzzy sets built this way and optimised fuzzy sets. However, the disadvantage is that the fuzzy sets may not reflect the distribution of the underlying training data, which would result in great performance loss.

3. Fuzzy sets that are defined by taking advantage of characteristics of underlying data. Again in case of no domain expertise available, it is also reasonable to assume common domain knowledge being similar to that obtained by utilising local data structure. A way to obtain such fuzzy sets is to discretise each feature space into a number of fuzzy sets a prior. However, depending on how the set of training instances is sampled from the problem domain, this may not necessarily reflect the real characteristics of the problem.

2.1.3.1 Fuzzy Decision Trees

A decision tree is a tree-like structure but with root node on top. Each internal node represents a test to the attribute of this node. Each branch so far corresponds to the outcome of the test, i.e., a value of the current node. The branch is then connected to next internal node or a leaf node, which comes with a decision label. A new example is classified by submitting a series of tests from the root for the acquisition of the decision label. Once the completion of decision tree construction, each path from the root node to the leaf node can be directly translated to a rule. Hence, the induction of decision trees can be seen as a straightforward means to obtain classification rules.

Basic Concepts

Central to the construction of a decision tree is the selection of a node that is of best discrimination classifying the examples. Entropy is a commonly used metric originally used in information theory that measures the impurity of a collection of examples. Given a set of examples S consisted of K decision classes, $S_i, i = 1, 2, \cdots, K$ is the subset of examples with decision label i, the entropy of the given set $Entropy(S)$ is defined as:

$$Entropy(S) = \sum_{i=1}^{K} -\frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \tag{2.16}$$

Given the definition of entropy, the classification capability of attribute A can be measured by taking the difference between entropy of original set S and an expected entropy after partitioning S with attribute A. The expected entropy is defined as the sum of entropies of each partitioned subset S_v, weighted by the fraction of examples $\frac{|S_v|}{|S|}$. The difference called information gain is defined as:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Dom(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{2.17}$$

where v is the possible value that attribute A can take from the domain $Dom(A)$; S_v is the subset of S partitioned with $A = v$.

Once the attribute is selected, the expansion process is repeated recursively for each nonterminal descendant node with examples associated with that node. The branch does not stop growing until either of the two conditions met, i.e., examples associated with this leaf node are pure with same class labels, or, all the attributes have been used down the path.

In order to deal with the possible shortcomings that crisp decision trees (such as ID3 [122], C4.5 [124]) encounter, the construction of fuzzy decision trees remains a topic of interest. In classical decision trees, the use of crisp sets for nominal attributes and crisp intervals for numerical attributes results in hard decision boundaries. Therefore, crisp decision trees are vulnerable to the situation that small changes in the attribute values of the examples being classified may result in sudden changes to the assigned class labels. This can be improved if attributes are described by fuzzy sets that permit gradual assessment of the membership of elements in the set.

The idea of building a fuzzy decision tree is the same as that of a crisp decision tree in the sense that an optimal attribute is selected recursively and partition the data based on the values of the attribute. This partition does not stop until some certain conditions are met. One major distinction between fuzzy decision tree and crisp decision tree is that the way the knowledge is represented has changed from numerical values (for continuous variables) to fuzzy terms. For classical decision tree, a numerical value can only fall into one of the partitioned intervals, while it can match against several fuzzy terms to various degrees, given the nature of overlapping fuzzy sets defined to describe concepts that are inherently vague or imprecise. Due to this, an instance will match against several tree branches when classifying, which leads to multiple terminal nodes. Therefore, the inference procedure embedded in fuzzy decision tree requires adjustment compared to crisp decision tree, where principles of fuzzy logic is necessary to be incorporated. Final decision comes from the aggregation of these matched terminal nodes through some defuzzification technique.

Another obvious extension to deal with fuzzy values is the fuzzy version of metric that is used to recursively select the attribute. Classic entropy measures the (im)purity of a node by calculating the weighted sum of instance counts that belong to different

decision classes. In case of fuzzy decision trees, individual instance is counted by a set of conjunctive fuzzy restrictions imposed from the root node to current node, each of which is often only partially matched against the fuzzy term. This is different from classical decision tree, where a instance can only belong to a certain tree branch.

In order to demonstrate the most natural metric extension, i.e., the adapted information gain in fuzzy decision tree, the technique used in [80] is reviewed here. For any node N to be expanded, the example count with regard to class k is calculated as:

$$P_k^N = \sum_j^{|E|} = f_2(\chi_j^N, \mu_{v_k^c}(y_j)) \tag{2.18}$$

where E is the set of training examples; f_2 calculates the satisfaction degree of the consequent v_k^c. This is propagated from the conjunctive restrictions from the root down to current node, therefore, χ_j^N is the satisfaction degree of the combination of these restrictions. Then the total example count is gathered summing example count with regard to all decision classes:

$$P^N = \sum_{k=1}^{|D_c|} P_k^N \tag{2.19}$$

where $|D_c|$ is the number of existing classes. As a result, the information of node N can be measured following traditional entropy as:

$$I^N = -\sum_{k=1}^{|D_c|} (\frac{P_k^N}{P^N} \cdot log \frac{P_k^N}{P^N}) \tag{2.20}$$

To further partition the instance space, an attribute V_i from the set of remaining attributes that have not been used in this branch is selected with the maximum information gain. Given a fuzzy set $v_p^i \in D_i$, where D_i is the term set defined for V_i, each node branch is weighted by the proportion of examples counts with corresponding attribute value in the training set as:

$$w_p^i = \frac{P^{N|v_p^i}}{\sum_{v_p^i \in D_i} P^{N|v_p^i}} \tag{2.21}$$

where $P^{N|v_p^i}$ counts the branch given that $D_i = v_p^i$, which is similar to that of 2.18. Similarly, the weighted information content with attribute V_i branching on node N is:

$$I^{S_{V_i}^N} = \sum_{v_p^i \in D_i} (w_p^i \cdot I^{N|v_p^i}) \tag{2.22}$$

where $I^{N|v_p^i}$ is the information content down the branch given that $D_i = v_p^i$; the calculation is similar to that of 2.20. The attribute V_{max} is then selected that brings the maximum information gain for current node.

$$V_{max} = \arg\max_i (I^{S_{V_i}^N} - I^N) \tag{2.23}$$

Induction of Fuzzy Decision Trees

Apart from Fuzzy ID3-based decision tree [80, 61] that directly generalises ID3, there are alternative approaches in literature. For example, it has been proposed in [168] to incorporates cognitive uncertainties such as vagueness and ambiguity into the induction of fuzzy decision trees, such that the attribute with the minimal ambiguity of a possibility distribution is selected for splitting. Optimisation principles of fuzzy decision trees based on minimising the total number and average depth of leaves has been discussed in [161], which proposed to use clustering to merge branches.

More recently, Gini index has been utilised as splitting measure for choosing the most appropriate attribute in [23], while fuzzifying the decision boundary without converting the numerical attributes into fuzzy linguistic terms. Alternatively, the method [22] is proposed to construct a fuzzy binary decision tree of significantly reduced size based on the adaption of a fuzzy supervised learning in Quest (SLIQ) decision tree. Furthermore, the paper [94] introduces the coherence membership functions to describe fuzzy concepts that build upon the Axiomatic Fuzzy Set (AFS) theory. The AFS decision tree is then built, in which the rules can be extracted and pruned afterwards, where potential subjective bias is eliminated due to the coherence membership functions and the underlying logic operators. A relatively dated review regarding fuzzy decision tree can be found in [29].

Also utilising the tree structure that can naturally be translated into a rule base, a fuzzy pattern-tree learning classifier as a novel machine learning method for classification has recently been introduced in [65, 135]. A pattern tree is a hierarchical, tree-like structure, whose inner nodes are marked with generalised logical operators and leaf nodes associated with fuzzy predicates on the input attributes. A pattern-tree classifier is composed of an ensemble of such pattern trees, each of which is built for one class. The fuzzy pattern-tree learning classifier is further generalised in [136], where Ordered Weighted Averaging operators (OWA) are used at the inner nodes to

increase flexibility. The improved algorithm works faster especially on large datasets with many instances or attributes, where only a fixed number of most promising candidates are considered instead of generating all expansions of a tree.

2.1.3.2 Fuzzy Association Rule-based classifier

Association rule mining (ARM) [1, 2] has been a very important approach in data mining that aims to discover interesting, hidden patterns among items in large database as an explorative data analysis. The relationship in association discovery are represented by frequent item sets and association rules with the general form being a $X => Y$ implication. The antecedent of the rule X is a collection of frequent item sets in the database, while the consequent Y is another set of frequent items that such that $X \cap Y = \phi$. An association rule-based classifier (ARC) is when the consequent contains only a class label. The hybridization of ARM and pattern classification results in a new rule induction approach for classification problems.

However, ARM algorithms deal with binary or categorical data. In real world, many data sets contain numerical features such that traditional ARM has to discretize them into a number of crisp intervals in order for ARM to work. This leads to abrupt transitions when the instances' values are on the boundaries of discretised intervals. As a result, the learnt rules are sensitive to small changes in attribute values, and are especially vulnerable to noisy data. In order to remedy this situation, fuzzy set theory has been incorporated into the ARM framework as the knowledge representation, such that the universe of discourse can be partitioned into a number of overlapping fuzzy sets. This could potentially relax the issue that arises from the sharp partitioned boundaries, making resultant rule base more robust to data that is noisy or imprecise. With the new knowledge granularity, the concepts from traditional ARM are required to be fuzzified in order to generate fuzzy association rules. The generation of fuzzy classification rules based on fuzzy association rule mining has attracted a lot of attention and been extensively studied in the literature.

Basic Concepts

Association rule mining originates from market basket analysis. Let $I = \{i_1, ..., i_k..., i_n\}$ be a set of items. A collection of one or more items $X \subseteq I$ is an itemset. Let

$T = \{t_1, ...t_j, ..., t_m\}$ be a set of transactions, each of which is a subset of items in I. An association rule is defined as an implication of the form $X => Y$ where $X, Y \subseteq I, and X \bigcap Y = \phi$.

In order to select interesting rules, a number of metrics have been used to evaluate rule significance. Among them, the most common ones are support and confidence. The support of an itemset X with respect to the transaction T measures the frequency of the itemset occurrence in the database and is defined as the proportion of transactions that contain X.

$$supp(X) = \frac{|\{t \in T : X \subseteq t\}|}{|T|} \tag{2.24}$$

An itemset whose support is greater than or equal to a *minisupp* threshold γ is called a frequent itemset. Confidence is an indicator that measures how often the rule is true. The confidence of a rule $conf(X => Y)$ with respect to transactions T, is the proportion of the transactions that contain X which also contain Y. Confidence is defined as:

$$conf(X => Y) = \frac{supp(X \bigcup Y)}{supp(Y)} \tag{2.25}$$

where $supp(X \bigcup Y)$ is the support of the union of items in X and Y. Thus confidence can be interpreted as an estimate of the conditional probability of finding the itemset on the right given that these transactions also contain itemset on the left. The task of association rule mining is to discover rules whose support and confidence are greater than the user defined threshold γ and *minconf* ϵ.

In order to deal with the possible abrupt transition resulting from the hard partitioning of numerical features, fuzzy set theory is utilised as the knowledge representation tool. To rephrase a general classification problem in the form of relational database, let $A_i, i = 1, \ldots, n$ denote the underlying domain variables, jointly defining the n-dimensional pattern space and respectively taking values from D_i. Let $A_{ji} \in A_i$ denotes a fuzzy set that the variable x_i may take, which can be done by consulting domain experts *a prior*, uniformly dividing the universe of discourse into a number equal intervals or partitioning the attribute into fuzzy sets utilising distribution of underlying feature space. Regardless of which approach to take, the fact that definitions of the fuzzy sets are assumed available for subsequent mining process forms a major approach for the induction of fuzzy rules with fixed and predefined fuzzy sets.

Formally, let $\hat{Z}$ be a fuzzy itemset such that $<\hat{Z}:A>=\{<A_{i_1}:A_{ji_1}>,...,<A_{i_q}:A_{ji_q}>\}$, where A_{i_1} is the first attribute of the itemset Z and $i_q <= n$ denotes the number of attribute-(fuzzy)term pairs in the itemset. The fuzzy support $fsupp(\hat{Z})$ of the fuzzy itemset $\hat{Z}$ with regard to the set of transactions T is defined as:

$$fsupp(\hat{Z}) = \frac{\sum_{t\in T}\prod_{A_i:A_{ji}}\mu_{A_{ji}}(t_i)}{|T|} \tag{2.26}$$

The fuzzy itemset $<\hat{Z}:A>$ is a frequent item set if $fsupp(\hat{Z})$ is higher than or equal to a user-defined minimum fuzzy support threshold $\hat{\gamma}$.

Similarly, a fuzzy association rule is an implication of the form $<\hat{X}:A>\Rightarrow<\hat{Y}:B>$, where the itemset $<\hat{X}:A>$ is the rule antecedent and $<\hat{Y}:B>$ is the consequent of the rule. The fuzzy confidence of the rule $fconf(<\hat{X}:A>\Rightarrow<\hat{Y}:B>$ is defined as

$$fconf(<\hat{X}:A>\Rightarrow<\hat{Y}:B>) = \frac{fsupp(<\hat{X}:A>\bigcup<\hat{Y}:B>)}{fsupp(\hat{X}:A)} \tag{2.27}$$

A fuzzy association rule is a strong rule if the support and confidence values are both higher than or equal the minimum fuzzy support $\hat{\gamma}$ and minimum fuzzy confidence threshold $\hat{\epsilon}$.

Principles of Association Rule Mining

In general, association rule mining is divided into two independent stages as below.

1. Search for all frequent itemsets whose support measures are bigger than or equal a user-defined threshold.
2. Generate strong association rules from the frequent itemsets, such that the confidence values of generate rules are bigger than or equal a user-defined threshold.

It usually is the first step that forms the bottleneck of the rule mining procedure, due to the exponential growth of the combination of attribute value pairs as the dimensionality increases. In literature, there are two main approaches for the efficient searching of frequent itemsets, i.e., Apriori algorithm being the initial approach to

tackle the problem and boost the development and popularity of association rule mining, and Frequent Pattern (FP) growth algorithm that significantly reduces the running time with only couple of scans of the database.

The idea of Apriori algorithm is based on the fact that if an itemset is frequent, then all of its subsets must also be frequent. Given that the support of an itemset never exceeds the support of its subsets, Apriori principle is known as the anti-monotone property of support. Specifically, Apriori algorithm uses a breadth first search strategy such that frequent itemsets with cardinality being one L_1 are first searched by scanning through the database, candidate itemsets with cardinality being C_k are then subsequently generated by a join step that combines the frequent itemset L_{k-1} with itself. Once the candidate set C_k is produced, the set of termsets stored in the candidate set will go through a prune step, such that those termsets whose subsets are not in the frequent itemset L_{k-1} will be deleted directly from C_k according to Apriori property. Remaining element of the candidate set will go through the support check, and are kept if their support values are greater than or equal to the user-defined threshold. The process goes on until L_k is empty. Association rules can then be generated with each of the mined frequent itemsets by imposing the minimum confidence constraint. The pseudocode of Apriori algorithm can be described in Algorithm 2.1.1.

Despite that Apriori algorithm can successfully find all frequent itemsets from the database, the overload of scanning the database repetitively with candidate generation significantly increase computational cost as the increase of problem dimensionality. To offset the candidate set generation-and-test approach, Frequent Pattern (FP) tree-based approach [60] is utilised for storing compressed information about frequent patterns, and FP-growth technique is proposed for efficient mining of frequent patterns in large databases. Different from Apriori, FP-growth avoids costly candidate generation and test by successively concatenating frequent 1-itemset found in the FP-trees, and it applies a partitioning-based divide-and-conquer approach which dramatically reduces the size of the subsequent conditional pattern bases and conditional FP-trees. The FP-tree can be constructed in the following steps:

1. Scan the database once to collect the set of frequent items F and their supports. Sort F in support descending order as L.

2. Create the root node of an FP-tree, T, and label it as 'null'. For each transaction t, do the following:

1: $L_1 = \{\text{frequent } 1-\text{itemset}\}$ (by counting the calculating the support of each item set)
2: **for** $k = 2; L_{k-1} \neq \emptyset; k++$ **do**
3: $C_k =$ Join operation by generating candidates from $L_{k-1}(p) \times L_{k-1}(q)$:
4: {
5: Insert into C_k;
6: Select $p.term_1, p.term_2, ..., p.term_{k-1}, q.term_{k-1}$
7: From p, q
8: Where $p.term_1 = q.term_1, p.term_2 = q.term_2, ..., p.term_{k-1} \neq q.term_{k-1}$
9: }
10: **for** termset $c \in C_k$ **do**
11: Check all the sub-termsets of all termsets in C_k, delete if they are not frequent termsets in L_{k-1}
12: **for** $(k-1)$ subset s of $c \in C_k$ **do**
13: **if** $s \ni L_{K-1}$ **then**
14: Delete c from C_k
15: **end if**
16: **end for**
17: **end for**
18: **for** termset $c \in C_k$ **do**
19: Calculate support value
20: **if** $supp(c) >= minisupp$ **then**
21: insert c into L_k
22: **end if**
23: **end for**
24: **end for**

Algorithm 2.1.1: Apriori algorithm

a) Select and sort the frequent items in t based on the order of L. Let the sorted frequent item list in t be $[p|P]$, where p is the first element and P is the remaining list. Call $insert\_tree([p|P], T)$

b) The function $insert\_tree([p|P], T)$ is performed as follows. If T has a child N such that $N.item-name = p.item-name$, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link be linked to the nodes with the same item-name via the the node-link structure. If P is nonempty, call $insert\_tree(P, N)$ recursively.

The FP-tree construction process needs exactly two scans of the database, i.e., to collect the set of frequent items as the first scan, and then constructs the FP-tree.

The algorithm for mining frequent patterns using FP-tree that has prove to be able to find complete set of frequent itemsets can be described as follows

1 α : frequent itemset in the database;
2 B : α's conditional pattern base;
3 β : an itemset in B;

```
1: FP-growth(Tree, α)
2: {
3: if Tree contains a single path P then
4:    for each combination beta of the nodes in the path P do
5:       generate pattern β ⋃ α with support = minimum support of nodes in β;
6:    end for
7: else
8:    for a_i in the header of Tree do
9:       generate pattern β = a_i ⋃ α with support = a_i.support
10:      construct β's conditional pattern base and then β's conditional FP-tree
         Tree_β
11:      if Tree_β ≠ ∅ then
12:         call FP-growth(Tree_β, β)
13:      end if
14:   end for
15: end if
16: }
```

Algorithm 2.1.2: Initialisation of fuzzy rule refinement

Induction of Fuzzy Association Rules

In order to smooth the abrupt transitions that come from crisp partitioning of continuous variables, the induction of fuzzy association rules have been studied in the literature, with the incorporation of fuzzy set theory into the framework of association rule mining. Given that the fuzzy set generation stage is done *a priori*, being independent of the rule mining process, this forms an alternative major category of inducing fuzzy rules for classification problems utilising fixed and predefined expertise in terms of fuzzy sets.

A fuzzy associative classifier is introduced in [30] to induce fuzzy classification association rules (CARs). The approach is based on the framework of Apriori algorithm, and extends the notions of support, confidence, redundancy and rule conflict for fuzzy knowledge representation. Similarly, [116] proposes an Apriori-based fuzzy

associative classification model with different methods for the initial partitioning of feature space. A fuzzy version of CBA [99], being a first association rule-based classifier that applies associative classification models to build recommender systems, is proposed in [98].

More recently, [100] proposes an fuzzy extension of gain-based association rule classifier, which learns the initial fuzzy partitioning utilising simulated annealing optimisation algorithm. A novel efficient fuzzy associative approach based on the framework of FP-growth is introduced in [10]. Another approach [7] presents an induction method to obtain fuzzy association rules consisting of three steps. Short fuzzy association rules are first mined following Apriori algorithm. A pre-selection process then selects most interesting rules, reducing the size of candidate rules. This is followed by a single objective genetic tuning process for the acquisition of a compact set of fuzzy association rules. This method is further extended in [46] with the use of multi-objective evolutionary algorithm for the post-processing stage, together with a new partitioning algorithm taking attribute partitioning interdependencies into consideration.

Finally, note that this section covers fuzzy association rule learning for its relevance to the learning mechanisms that are to be utilised in the subsequent development. However, the fuzzy association rule learning is not directly exploited. Further readings regarding this and other approaches to fuzzy learning (e.g., fuzzy neural networks) that are not included in the following review on evolutionary fuzzy systems can be found in [38, 159].

2.2 Evolutionary Algorithms

Evolutionary algorithms (EAs) refer to a set of generic population-based metaheuristic optimisation algorithms, which are inspired by the principles of Darwin's biological evolution. A generic EA typically works by randomly generating population of individuals for the first generation. Individuals with better qualities evaluated with a fitness function are then selected for reproduction. Offsprings are bred via genetic operations such as crossover and mutation in order to maintain diversity. These generated offsprings are then evaluated with better ones more likely to be selected

for the generation of next iteration. The process goes on until certain termination conditions are met.

EAs have been successfully and extensively applied to a broad range of combinatorial and search problems for two main reasons. First, a wide range of problems can be approximated well with EAs, for their powerful search capabilities without making assumptions of underlying fitness landscape. Second, it is simple and straight forward to encode specific problems with EAs that require little of domain knowledge. Due to the power of EAs and their being relatively problem-independent, EAs have been intensively studied and utilised for the identification of fuzzy rule-based systems. This sub-chapter reviews two particular approaches to EAs: Genetic Algorithm and Particle Swarm Optimisation that have been used in the thesis for the design for fuzzy rule-based systems for complete purpose, among many others [40, 42]. Further details regarding EAs in general can be found in [111], but are omitted herein.

2.2.1 Genetic Algorithm

Genetic algorithm (GA) [56] inspired by the process of natural selection is one of the most popular EAs. In a GA, the population is made up of a collection of individuals, each of which represents an underlying solution to the problem [24, 27]. A solution is consisted of a number of chromosomes, each of which encodes a certain trait of the problem, e.g., blood type of a person. Traditionally the solutions are represented with binary strings, i.e., 0s and 1s to indicate certain characteristics being on or off. Depending on the types of problems, it also works for GAs to deal with numerical values.

The GA evolution starts with a randomly generated population, and is an iterative process such that population from each generation is evolved towards better solutions, where the solutions are evaluated with objective functions that include targets, e.g., accuracy, complexity, to be optimised. Better solutions from generation to generation are created due to the use of genetic operators such as crossover and mutation. Crossover is performed on a pair of parent solutions such that a new child solution can be created by sharing characteristics of the parents. Different from crossover that inherit traits from parents, mutation randomly changes the states of some pieces of the chromosomes such that the generated solutions may be entirely different from previous ones. Individuals with better qualities from the pool of current iteration

are selected for future generation, thus guaranteeing average fitness will increase generation by generation especially when elitism is used.

The general process of a GA works as follows:

```
1  t = 0;
2  initialise P(t = 0);
3  evaluate P(t = 0);
4  while condition is true do
5      while |P_{t+1}| < |P| do
6          // select parents;
7          parents = selection_method(P_t);
8          if crossover_rate then
9              children = crossover_method(parents);
10         else
11             children = parents;
12         if mutation_rate then
13             children = mutation_method(children);
14         evaluate(children);
15         P_{t+1}.add(children);
16     t = t + 1;
```

Algorithm 2.2.1: Genetic algorithm

2.2.2 Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) was first introduced in [87], and was intended for simulating the flocking and schooling patterns of birds and fish. PSO is a meta-heuristic population-based algorithm, and has been successfully applied to various applications (e.g., [25, 24, 3]). PSO optimises a problem with a population of particles representing candidate solutions. These candidate solutions are updated stochastically with a guide towards the previously best known positions in the search space.

Two primary operations are involved in particular, for the update of PSO processes: velocity update and position update. During each updating iteration, termed

generation, every particle's movement is influenced by its local position as well as by the currently known best global position in the search space. A new velocity vector is then computed for each particle based on its current velocity, the distance from its previous best position, and the distance from the global best position so far. The new velocity is in turn used to calculate the next position for each particle in the search space.

More formally, the velocity update for each generation is implemented through the following assignment:

$$v_x = wv_x + c_1 r_1(x_{gBest} - x) + c_2 r_2(x_{pBest} - x) \tag{2.28}$$

where w is the so-called inertia weight that affects the trade-off between convergence and exploration-exploitation in the PSO updating process; c_1 and c_2 are two positive constants, termed social and cognitive scaling parameter in the literature, respectively; r_1 and r_2 are two random numbers within the range $[0, 1]$, introducing the stochastic nature during the update; x is the position of a certain particle dimension (or the fitness of the rule weight of a certain rule that leads to the current classification accuracy, in terms of the present application problem); x_{gBest} is the global best position of all particles (namely the fitness of the rule weights currently capable of achieving the highest classification accuracy overall); and x_{pBest} is the best individual position where the particular particle p achieves the current best position. The position is itself updated by the assignment:

$$x = x + \epsilon v_x \tag{2.29}$$

where ϵ is a further real-valued parameter used to control the evolving speed. The interaction between PSO positions and PSO velocities is illustrated in Figure 2.3.

Both the global best position and the best individual position are used during the update process, with the swarm collectively moving towards the overall best position. The process is iterated for a set of times or until a minimal error is achieved. The overall PSO process can be illustrated as shown in Algorithm 2.2.2.

2.3 Evolutionary Fuzzy Systems

Owing to the powerful capability of EAs for optimisation problems and their context-free encoding, EAs have been intensively utilised for the identification of fuzzy

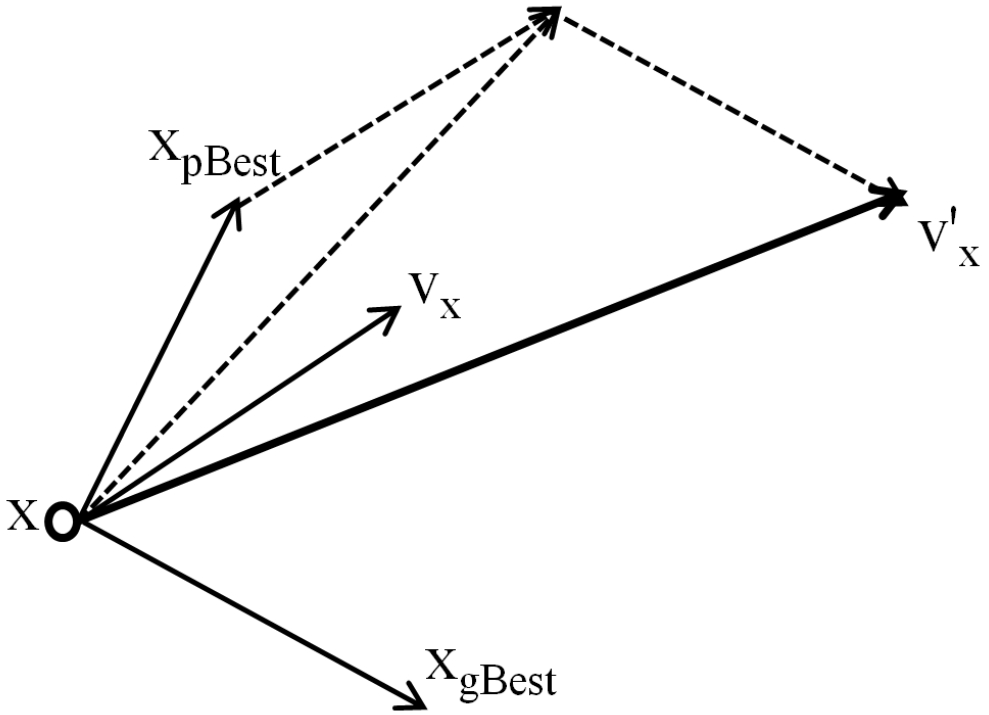

Figure 2.3: Update of PSO velocity and position

rule-based systems. An evolutionary-based fuzzy system (EFS) [32, 47, 49] is a fuzzy system that is augmented by a learning process for the identification of FRBCS components based on EA. A large amount of research has been developed to learn Mamdani type linguistic fuzzy models. Relating to this thesis is the literature that learns FRBCS based on a set of fixed and predefined fuzzy sets reflecting domain expertise.

In particular, MOGUL [34] is a genetic fuzzy rule-based system, following an iterative rule learning approach with additional simplification and fine-tuning processes. A fuzzy rule is generated for each example by evaluating all the global fuzzy rules. The obtained rule is added to the final set of fuzzy rules. The data covered by current rule set to a certain degree is removed and not considered for future iterations. The iterative process ends up when no more uncovered training data remains. Then the genetic simplification process is performed based on a binary-coded GA with fixed-length chromosomes, followed by a genetic tuning process based on a real-coded GA algorithm in which each individual represents a complete data base.

```
1: for each particle do
2:    Initialise particle
3: end for
4: while maximum iterations or minimum error not attained do
5:    for each particle do
6:       Calculate fitness value
7:       if the fitness value is better than pBest then
8:          Set pBest = current fitness value
9:       end if
10:      if pBest is better than gBest then
11:         Set gBest = pBest
12:      end if
13:   end for
14:   for each particle do
15:      Calculate particle velocity according to the velocity update equation (8)
16:      Calculate particle position according to the position update equation (9)
17:   end for
18: end while
```

Algorithm 2.2.2: PSO update process

The work of [76] formulates rule base learning problem as a combinatorial optimisation problem with a fixed fuzzy partition of the feature space. An initial fuzzy rule base is generated by a heuristic procedure such that individuals of first GA generation are already able to obtain a reasonable performance to speed up the subsequent evolvement. Optimisation is first achieved by a single objective GA that only takes accuracy into consideration. This is further enhanced by a two-objective GA [70] that considers a weighted combination of targets, i.e., the minimisation of rule numbers and maximisation of performance, followed by a three-objective GA approach [73] that additionally considers the total number of rule antecedents.

In order to deal with high dimensional problems that would result in a large number of initial rules with the heuristic rule generation method, it is proposed in [72] to use a 'don't care' fuzzy set that leads to the full matching degree regardless of the input value. This 'don't care' label adds an important option for genetic selection process, such that fuzzy rules with attributes being labels are shortened and the resultant fuzzy rule base exhibits more flexibility and higher interpretability. A prescreening procedure [69] of candidate rules is also proposed to filter the initial rule base based on the number of antecedents per rule.

More recently, in [78], a hybrid algorithm FH-GBML of two fuzzy genetics-based approaches is proposed. It uses the Pittsburgh style to encode a set of fuzzy

rules as an individual, while using the Michigan style for partially modifying each rule set as heuristic mutation. FH-GBML has been further expanded into a parallel distributed model [68] to decrease computation time significantly, where a population of individuals is divided into multiple islands based on the island model. The partitioned training data subsets are periodically rotated over the islands, with the best rule set in each island migrating periodically as well.

SLAVE [58, 59] is a well-known fuzzy rule induction approach in the literature that learns rules of a disjunctive normal form through an iterative algorithm. Each iteration a single fuzzy rule is extracted by a GA that best represents the system and incorporated into the final rule set. In order to obtain new and different rules, examples covered by the learned rules are removed. The iterative scheme is repeated until the set of rules obtained adequately represent the training data. SLAVE2 [57] is an improved version in the sense that it includes more information in the process of learning individuals rules, utilising the proposed calculus of the positive and negative examples, as well as new fitness functions and genetic operators. A number of different versions of SLAVE over the years have recently been reviewed in [54], but this is beyond the scope of this thesis.

GP-COACH [14] is a genetic programming-based learning approach, which also learns rules of a disjunctive normal form with a coding scheme that expresses one rule per tree. GP-COACH relies on the cooperative-competitive learning strategy, where the population constitutes the rule base. It uses a token competition mechanism to maintain the diversity of the population and this obliges the rules to compete and cooperate among themselves and allows the obtaining of a compact set of fuzzy rules. SGERD [105] proposes a novel steady-state GA-based algorithm to extract a compact set of fuzzy rules. The selection mechanism is nonrandom, such that only the best individuals can survive. To select the rules with high generalisation capabilities, SGERD makes use of rule and data dependent parameters, as well as an enhancing function that modifies the rule evaluation measures in order to assess the candidate rules more effectively before selection.

2.4 Summary

This chapter has first introduced a number of important concepts relating to fuzzy rule induction by decomposing a fuzzy rule base into finer granularities for explanation.

Linguistic variables and fuzzy sets that are defined with membership functions constitute atomic fuzzy propositions. A conjunctive set of fuzzy propositions forming the rule antecedent with a class label as the rule consequent makes up a typical fuzzy classification rule. With the aid of fuzzy logic, a conclusion can be drawn when a observation comes simultaneously firing multiple fuzzy rules.

In addition, this chapter also reviews fuzzy decision trees and fuzzy association rule-based classifiers, being two of the most popular fuzzy rule induction approaches, which directly induce fuzzy rules with the incorporation of fixed and predefined fuzzy sets reflecting domain expertise. Given recent popularity combining evolutionary algorithms into fuzzy system design, this chapter briefly introduces paradigms of GA and PSO in particular, given that both will be utilised in following chapters as optimisation algorithms to induce fuzzy rules. Well-known evolutionary fuzzy systems are then reviewed, a number of which will be compared with the proposed work in subsequent chapters.

Chapter 3

Fuzzy Rule Weight Modification with Particle Swarm Optimisation

A major challenge in learning FRBCSs often exists where the membership functions defining the antecedent fuzzy sets are prefixed, with each having a specific linguistic meaning pre-specified by domain experts (and typically also known to the user). Due to the need of maintaining the interpretability [96, 52, 19] of a learned model, any learned fuzzy classification rule is required to use one of these fuzzy sets to specify the value of each attribute. Yet, using a fixed quantity space consisting of such given fuzzy sets limits the accuracy of the learnt rules. Fortunately, this problem can be tackled by modifying the weights associated with the individual rules.

Rule weights intuitively reveal the relative importance amongst all the rules in a given rule base. The greater the rule weight of a fuzzy if-then rule is, the more likely it will be chosen to classify an unseen pattern amongst all the fuzzy rules that cover the subspace of that pattern. The modification of a rule's weight is in effect equivalent to the adjustment of the membership functions of those antecedent fuzzy sets in the rule [113]. Interestingly, the adjustment of rule weights is much easier than directly modifying the antecedent fuzzy sets (which would involve the learning of a number of parameters for each membership function), since there is only one single parameter (namely, the weight itself) per rule to learn [79]. This also has the benefits where a practical fuzzy rule-based system has already been in use while a set of newly collected data needs to be promoted into rules, such that the

promotion can be independently done without disrupting the existing rule base. A method is thus required to improve the performance of such constructed rule-base by carefully adjusting the rule weights, instead of learning a set of dynamic membership functions.

In [115], a seminal method of leaning rule weights is proposed by the use of an error correction-based learning procedure with post pruning, through a "Reward and Punishment (R&P)" scheme. It works by increasing the weight when a pattern is correctly classified by the current rule, and decreasing the rule weight otherwise. Another weighting approach is reported in [104], by dividing the covering subspace of each fuzzy rule into two subdivisions based on a given threshold. The association degree of any pattern with a so-called compatibility grade above the threshold is enhanced by increasing the weight. The splitting threshold for each rule is found by exploiting the distribution of patterns in the subspace covered by that rule. Other rule weight learning methods for building FRBCSs include [79] and [174]. The importance and effects of learning rule weights in FRBCSs have been discussed and highlighted in [71], and a number of heuristic methods for fuzzy rule weight specifications can also be found in [77].

The performance of a particular fuzzy rule may be improved by directly adjusting its rule weight. However, the performance of its neighbouring fuzzy rules (i.e., those that also cover the same given pattern) may be deteriorated or even become useless due to the propagation of such modifications to the rest of the rule base. The overall consequence is thus unpredictable when all the rule weights are changing successively. Instead of solely using heuristic weighting functions to tune fuzzy if-then rule weights, this chapter proposes an evolutionary algorithm-based approach to modifying rule weights in FRBCSs. In particular, Particle Swarm Optimisation (PSO) [87] is employed as the evolutionary algorithm to evolve rule weights in order to improve the classification accuracy.

The reminder of this chapter is organised as follows. Section 3.1 presents the method for the generation of an initial fuzzy rule base. Section 3.2 demonstrates the approach of fuzzy rule weight refinement with PSO. Section 3.3 conducts a number of experiments demonstrating the efficacy of the proposed approach. Section 3.4 concludes this approach.

3.1 Generation of An Initial Fuzzy Rule Base

The task of learning from or generalising a given problem description, by the use of fuzzy logic and fuzzy sets, is to find a finite set of fuzzy if-then rules capable of reproducing the input-output behaviour of a given system (or process). Without losing generality, the system to be learnt is herein assumed to be a multiple-input-single-output, containing n inputs and one output and involving m patterns for an M-class problem. A fuzzy if-then rule $R_j, j = 1, 2, ..., N$, for such a system is represented as follows:

$$\text{If } x_1 \text{ is } A_{j1} \text{ and } ... \text{ and } x_n \text{ is } A_{jn} \text{ then class is } C_h \text{ with } w_j \tag{3.1}$$

where $x_1, x_2, ..., x_n$ are the underlying linguistic variables, jointly defining an n-dimensional pattern space (with N denoting the number of such fuzzy rules); A_{ji}, $i \in \{1, 2, ..., n\}$, is the fuzzy value of the corresponding antecedent x_i; C_h, $h \in \{1, 2, ..., M\}$, is the consequent class for the M class problem; and w_j is the rule weight of fuzzy rule R_j indicating the strength that any input pattern $X_p = [x_{p1}, x_{p2}, ..., x_{pn}]$, $p \in \{1, 2, ..., m\}$ within the fuzzy subspace delimited by the given antecedent values is deemed to belong to the consequent class C_h.

In order to generate an initial set of fuzzy if-then rules, each dimension of the pattern space is divided into K ($K \geq 2$) subsets $\{A_1^K, A_2^K, ..., A_K^K\}$. Practically speaking, partitioning the input space and defining the corresponding fuzzy sets are typically done by the domain experts (even though such specification may reflect a certain biased view of particular individuals). In many cases [79, 104, 74, 75], simple fuzzy grid partition of input space is adopted in order to generate an initial rule base. Of course, the performance of a resulting learnt classifier may vary in relation to the variation of the partition of the input space, especially regarding the number of the partitions made. When the fuzzy partition is too coarse in the sense that the number of generated fuzzy subspaces is too small, the performance of the corresponding fuzzy classifier may be low. On the other hand, if the partitioning of the fuzzy subspace is too fine such that the number of generated fuzzy subspaces is too large, the testing data may not be fully covered by the resulting rules, due to there being not sufficient data points at the training phrase [74]. Moreover, the finer the partition is, the more likely that more rules will be generated in the initial rule base, which will in turn lead to more complex computation in achieving the classification task using the resultant

rules. Note that the impact of the size of a rule base upon the performance of a learning classifier will be further investigated later by examining the effects of using different partitions of a given problem domain.

In this work, for simplicity and also for unbiased comparison, other than the two delimiting values (which are defined as rectangular triangular fuzzy sets) each dimension is simply divided equally into K fuzzy regions with the corresponding fuzzy membership values being determined by the symmetric triangular functions as shown in Figure 3.1, where a and b represent the minimum and maximum value of x_{pi} taken from the training examples, respectively. The vertex location of a symmetric triangular is calculated according to its position within the K partitions. Membership values of x_{pi} in a new pattern below a or above b are set to 1. Each partition is identified by a fuzzy rule if there is at least one training pattern in that pattern subspace [74]. That is, given an input partitioning of pattern space, a fuzzy rule will be generated only when there is a training pattern covered by this rule. Thus, a problem with m training patterns, m rules will be generated at most.

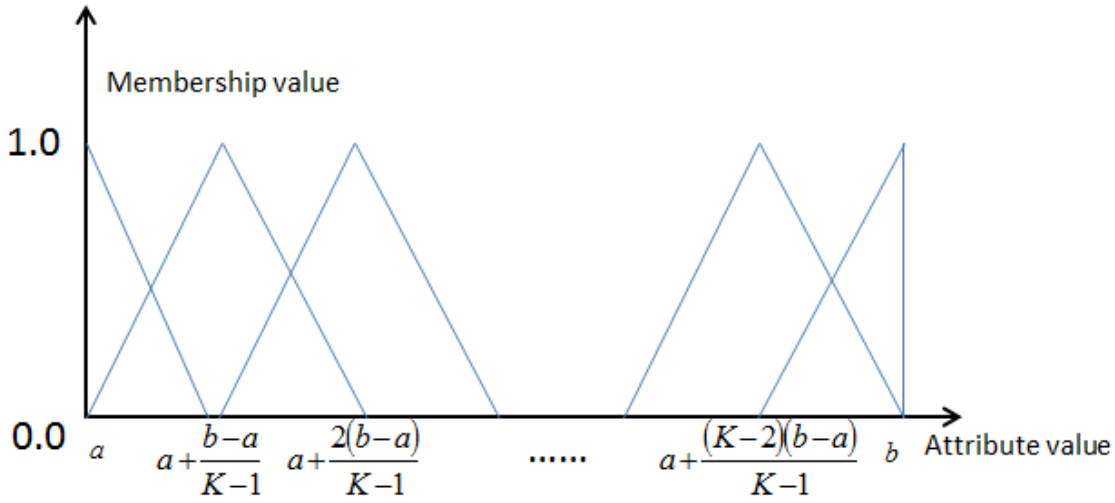

Figure 3.1: Partitioning of each pattern space dimension

There are a number of different approaches to specifying fuzzy rule weights [77]. This work adopts the classical method of [75] owing to its maturity. Following this approach, the consequent class C_h of fuzzy rule R_j and the corresponding rule weight w_j are determined by the following procedure, where rule generation is a direct by-product:

1. Calculate the matching degree for each class C_h with respect to the possible antecedents such that

$$\beta_{C_h} = \sum_{X_p \in C_h} \prod_{i=1}^{n} \mu_{A_{ji}}(x_{pi}) \tag{3.2}$$

where X_p are the training patterns defined with the corresponding n-dimensional fuzzy subspace $A_j = A_{j1} \times A_{j2} \times \cdots \times A_{jn}$, and $\mu_{A_{ji}}(\cdot)$ is the membership function of the antecedent fuzzy set A_{ji}.

2. Find $\beta_{C_T}, T = 1, 2, ..., M$, such that

$$\beta_{C_T} = max\{\beta_{C_1}, \beta_{C_2}, ...\beta_{C_M}\} \tag{3.3}$$

where C_T is the class of the maximum matching degree with regard to the antecedent fuzzy sets, forming a candidate if-then rule relating the antecedents and the class.

3. Set the rule weight w_j to a candidate rule with the following value if its class C_T is the unique one that takes the maximum matching degree in Eq. (3.3):

$$w_j = (\beta_{C_T} - \beta) / \sum_{h=1}^{M} \beta_{C_h} \tag{3.4}$$

$$\beta = \sum_{C_h \neq C_T} \beta_{C_h} / (M-1) \tag{3.5}$$

where β is the sum of the matching degrees for all training patterns belonging to the same fuzzy subspace, except those covered by C_T. Otherwise, discard the corresponding candidate rule when two or more classes take the maximum value in Eq. (3.3) or all the β_{C_T} are zero, since it cannot be uniquely determined or there is no training pattern in support of this rule.

4. Promote all remaining candidate rules as the members of the learnt rule base, with their corresponding rule weights assigned.

Note that the above method for rule generation and rule weight specification is straightforward when a two-class problem is considered. For instance, assuming that $\beta_{C_1} > \beta_{C_2}$, the consequent class is determined to be Class 1 and its weight will be $(\beta_{C_1} - \beta_{C_2})/(\beta_{C_1} + \beta_{C_2})$. Interestingly, suppose that there are almost no Class 2

patterns in the training data set, the result will be $\beta_{C_1} >> \beta_{C_2} \approx 0$ and $w_j \approx 1$. If however, the total matching degrees of patterns for Class 1 and Class 2 are very similar to each other $\beta_{C_1} \approx \beta_{C_2}$, then $w_j \approx 0$.

A popular and easy to understand, and perhaps also the simplest method for classifying a new pattern is based on the strategy of “single winner rule" or “winner taking all" [70]. This is employed in this work (but others can be used alternatively if preferred which can be found in [33]). The class C_{X_p} of pattern X_p is determined by

$$C_{X_p} = \underset{C_h, h=1,2,...M}{\arg\max}\ \alpha_{C_h} \tag{3.6}$$

where α_{C_h} is

$$\begin{aligned}\alpha_{C_h} = \max\{(\prod_{i=1}^{n} \mu_{A_{ji}}(x_{pi}))w_j | w_j \text{ is associated with } R_j, \\ R_j \text{ is associated with } C_h, j = 1, 2, ..., N\}\end{aligned} \tag{3.7}$$

The inferred class is the consequent of the fuzzy rule that has the maximum value of antecedent matching degree by the corresponding rule weight. If two or more classes take the maximum value in Eq. (4.10) or the matching degree is zero at X_p, then the pattern cannot be uniquely classified. To force a classification (if desired), such a pattern may be assigned with a default class label that is associated with most training instances.

3.2 Rule Weight Refinement with PSO

This section first illustrates how the classification boundary may be affected with a set of rule weights taking different values, reinforcing the need for the development of the current work. It then introduces how PSO is employed to refine rule weights for FRBCSs, followed by a summarised description of the general structure of the present work, including a brief analysis of the algorithm complexity.

3.2.1 Influence of Rule Weights on Classification Boundaries

A simple example will help demonstrate the effects of adjusting rules weights on the accuracy of the resulting classification boundary. Consider the following case

with a two-dimensional input space. For each of the two input variables, x_{p1} and x_{p2}, suppose that three descriptive fuzzy sets are defined such that x_{p1} may take a value on either $A_{11} = Small$, $A_{12} = Medium$, or $A_{13} = Large$, and x_{p2} on either $A_{21} = Short$, $A_{22} = Medium$, or $A_{23} = Long$. The two-dimensional pattern space is then divided into $3^2 = 9$ fuzzy subspaces, as shown in Figure 3.2. Each of the input subspace forms a possible fuzzy if-then rule. The dotted lines in Figure 3.2 also show the classification boundaries.

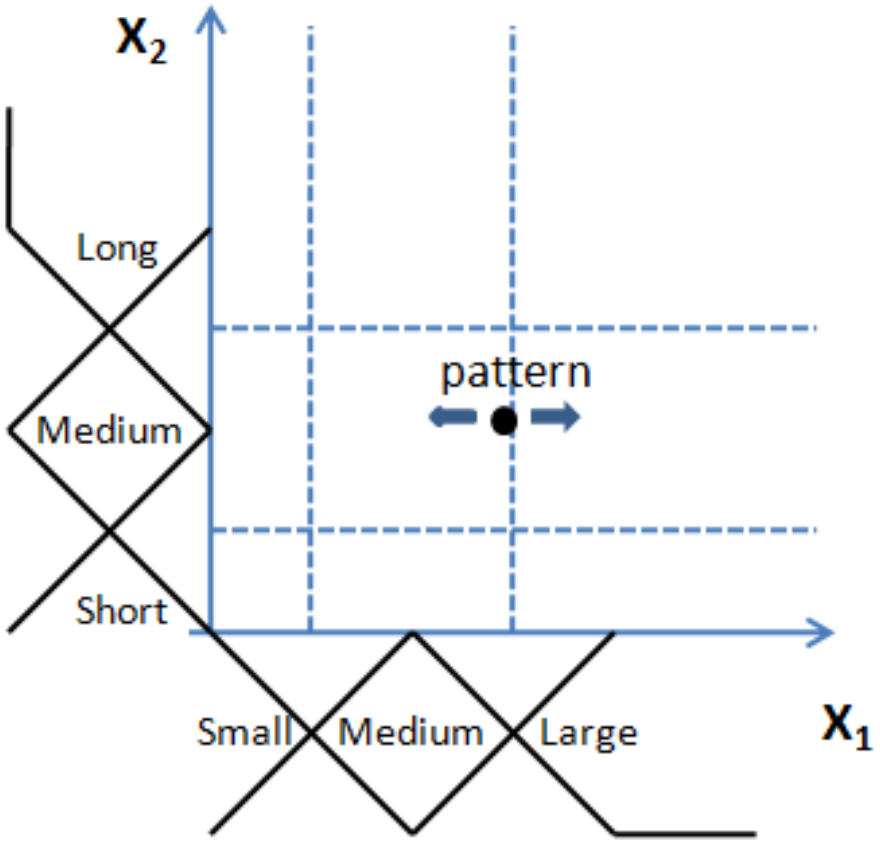

Figure 3.2: Fuzzy subspace of a two-dimensional pattern space

A newly collected pattern X_p is classified by first fuzzifying the attribute values using the corresponding fuzzy membership functions, and then checking if there is any match between the fuzzified value and the antecedent fuzzy sets of each given rule. Based on the single winner rule principle, the pattern is identified with the class label from the rule that is of the following maximum matching degree:

$$C_{X_p} = \underset{C_h, h=1,2,\ldots M}{\arg\max}\ \gamma_{C_h} \tag{3.8}$$

where γ_{C_h} is of the same value as α_{C_h} that is obtained from Eq. 3.7 when the value of every rule weight is set to 1. This is generally depicted in Figure 3.3 (adapted from [53]), where $\alpha(X_p, R_{hj})$ stands for the matching degree of the pattern X_p and the subspace of which is covered by those rules whose consequent is C_h.

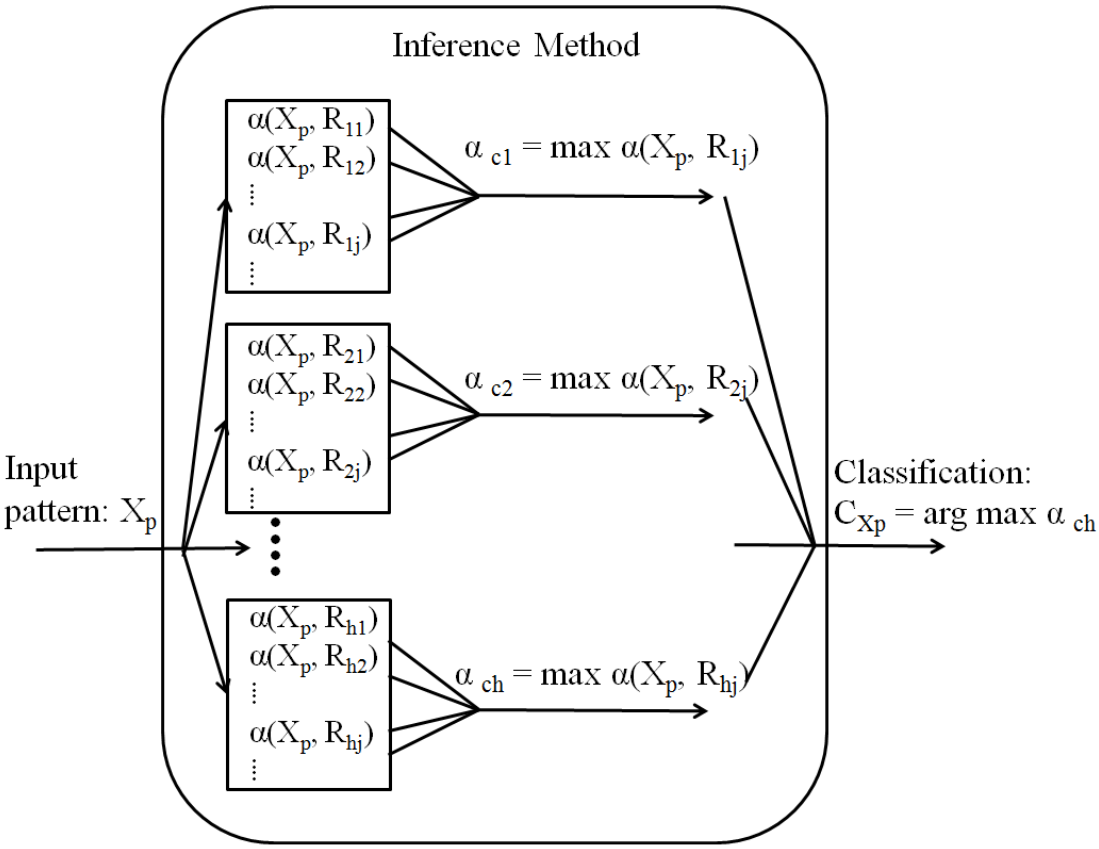

Figure 3.3: Single winner rule

When there are patterns misclassified, classification boundaries can be adjusted to recover the system performance by modifying the membership functions of the linguistic values. Figure 3.4 shows the classification boundary is adjusted by modifying the membership functions of fuzzy sets on x_1 axis. Although modifying potential membership functions can adjust the classification boundary and improve the performance of a fuzzy rule-based system [35], it may destroy the potential linguistic meanings given by the domain experts and hence, the interpretability of the learnt model. Also, the entire learning process needs to be rerun when knowledge derived from newly collected patterns is required to be combined with the existing rule base.

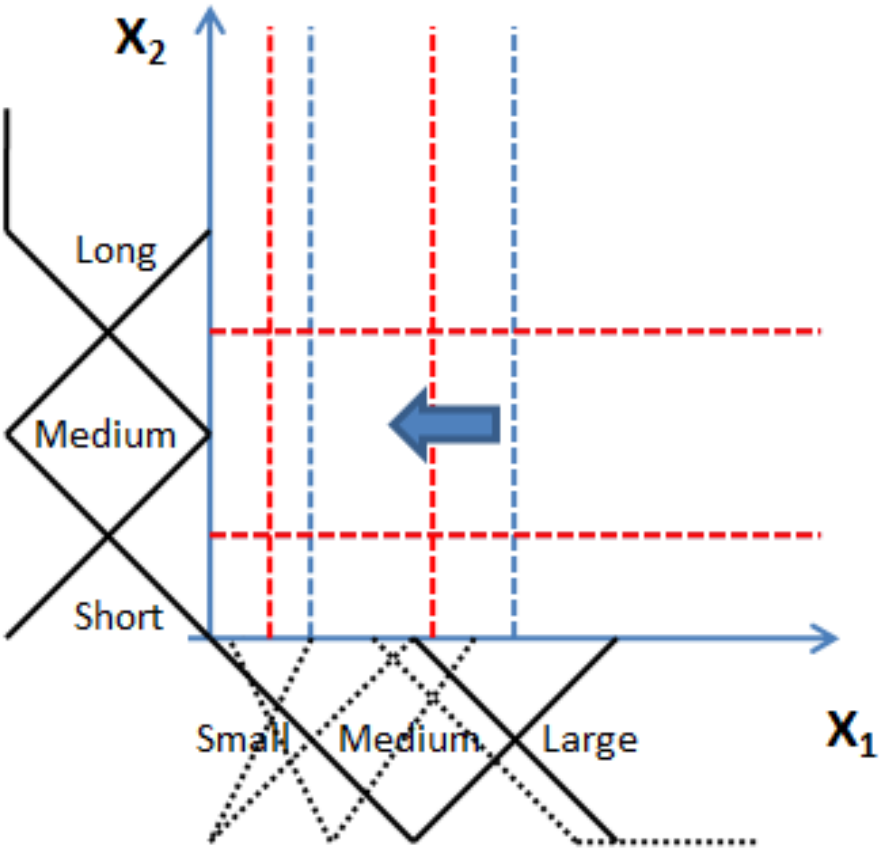

Figure 3.4: Modification of classification boundary on membership functions

According to Eq. (3.7), the class label for a new pattern X_p is determined by both the matching degree of its fuzzified value with the antecedent fuzzy sets and its corresponding rule weight. It is possible for a pattern to be misclassified, however. This is because a pattern may fall into one of the different neighbouring classes implied by certain fuzzy rules, as shown in Figure 3.2 where the black dot is on the edge of two fuzzy subspaces. For a two-dimensional problem, for instance, the equation $\mu_{A_j}(X_p)w_j = \mu_{A_{j'}}(X_p)w_{j'}$ holds while deciding on which class a given pattern may belong to. Thus, the classification boundary is determined by the ratio of w_j and $w_{j'}$ only. Consequently, the areas dictated by any two neighbouring classification rules may be linearly expanded or narrowed by the ratio of their rule weights. Consider rules R_{j1}, R_{j2}, and R_{j3} as an example in Figure 3.5. Instead of modifying membership functions, keeping the rule weights of w_{j1}, w_{j3} unchanged but reducing the value of w_{j2}, the areas covered by R_{j2}'s neighbouring rules R_{j1} and R_{j3} will be expanded while the area covered by R_{j2} is contracted.

Heuristic rule weights indicate the exact decision areas originally depicted by the

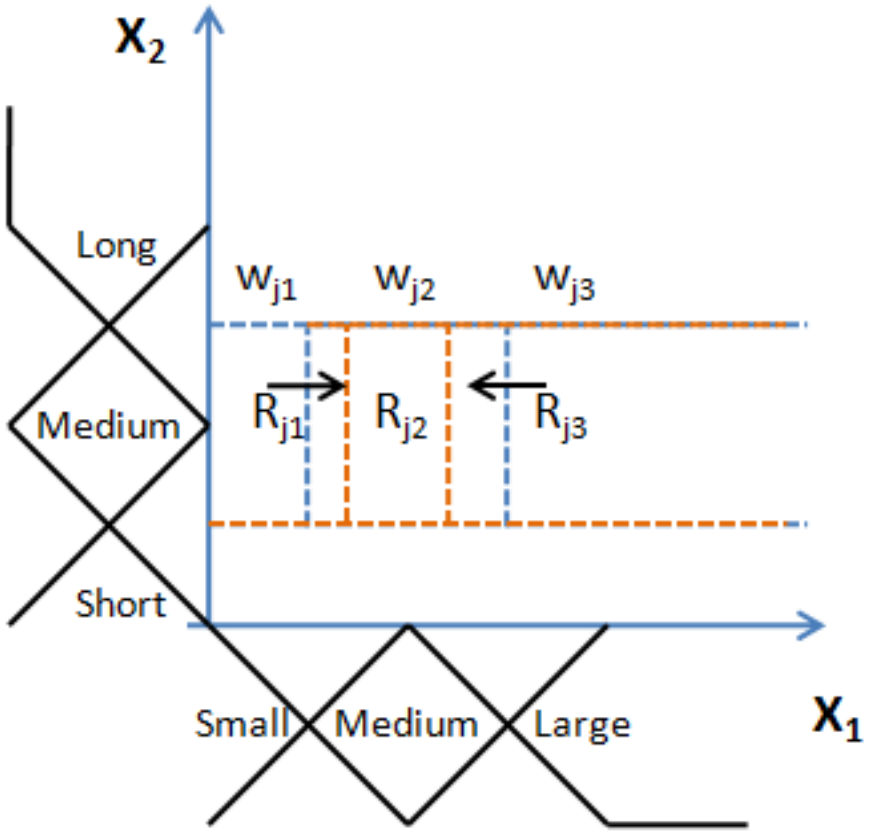

Figure 3.5: Modification of classification boundary by rule weights

predefined fuzzy sets [75]. The closer the value of a rule weight is to 1, the more reliable or more significant the rule is. With the single winner rule as the reasoning strategy, any modification of rule weights, through increasing or reducing the weight value, is in fact equivalent to adjusting the reliability of the relevant individual rules. This is in turn equivalent to reshaping the overall classification boundaries. The adjustment of any two neighbouring rule weights is linear in determining new classification boundaries, but the situation will become much more complicated if the modification of all rule weights is performed simultaneously. Figure 3.6 shows an example of one possible irregular classification boundary with various rules weights [71].

In order to obtain a higher classification performance the rule weight of the wining rule may be required to increase. However, adjusting the rule weight for any individual rule also affects the classification boundaries of its neighbouring rules. That is, whilst the performance of a certain fuzzy rule may be improved by directly changing its rule weight, the performance of its neighbouring fuzzy rules may be

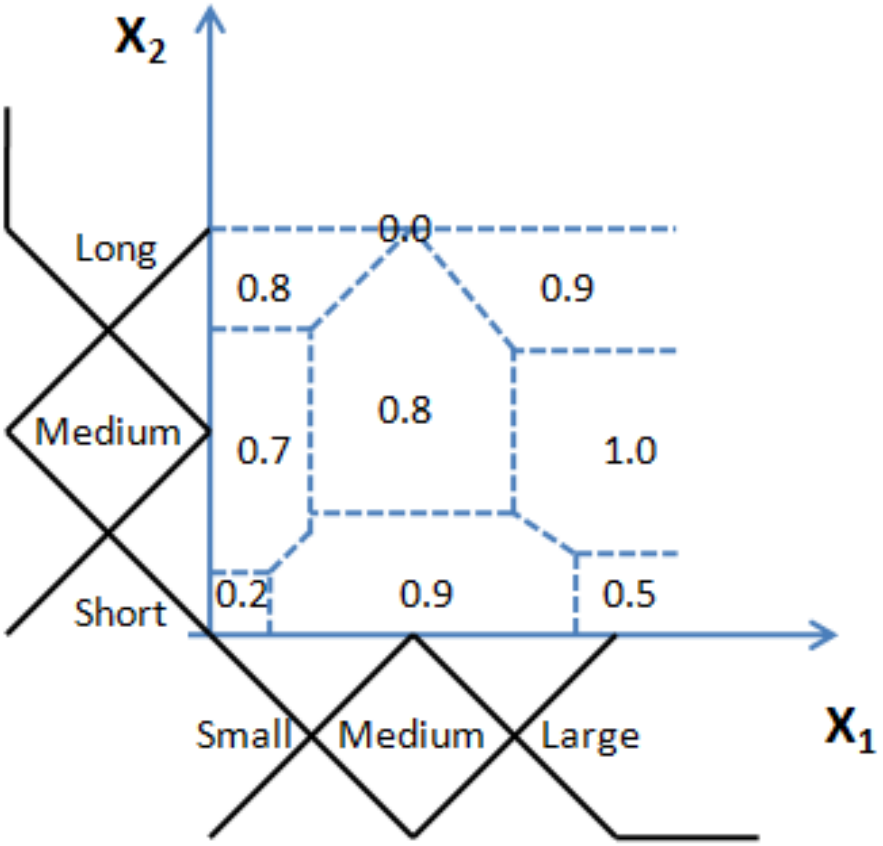

Figure 3.6: Classification boundary of an irregular shape

deteriorated as a consequence. The overall consequence is thus unpredictable when all the fuzzy rules are changing successively. A method is therefore required to deal with all existing rule weights in a synchronised manner to achieve overall optimal classification performance.

Broadly speaking, the process of finding an optimal combination of a full set of rule weights appears similar to the behaviour of a particle swarm going towards the best solution with each particle's movement influenced by both its local best position and the currently best known position amongst all rules, as with typical applications of Particle Swarm Optimisation (PSO) [87]. Inspired by this observation and the success of PSO in obtaining optimal solutions in multi-dimensional search space, PSO is employed below to evolve the weights of a fuzzy rule set.

3.2.2 Rule Weight Refinement with PSO

Further to the power of searching for optimal solutions in a discrete search space, PSO can also deal with real numbers directly (and hence the term optimisation is used). When it comes to adjusting real number encoded rule weights, inherent PSO mechanism of updating particle positions and velocities (Eqn. 2.29 and 2.28) can also make straightforward changes regarding rule weights. The dimensionality that each particle can have is herein set to be the same as the number of the variables considered in the problem. In utilising PSO for tuning the rule weights in an FRBCS, the PSO only needs to maintain a single static population whose members are tweaked in response to new discoveries about the search space. Each particle typically starts at a random location [88], and is accelerated during the iterations towards the particles that have achieved the previous best position and the global best position so far. The position of a particle corresponds to the fitness measure that determines the quality of the emerging solution.

For the present application, an initial fuzzy if-then rule base is firstly built with a number of predefined fuzzy sets, each having a predefined meaning given by domain experts. This is done via the use of Eq. (3.2) and (3.3) first, in order to obtain a consequent class for a certain rule and then, Eq. (3.2), (3.4) and (3.5) are used to create initial rule weights for the resulting rules. Tuning the rule weights is regarded as an optimisation problem of concurrently finding the best combination of them.

To obtain an optimal set of rule weights with PSO, the problem needs to be interpreted in terms of PSO specification. In particular, each of the existing weights is encoded as one particle dimension, and one particle then represents the entire set of the rule weights in the existing fuzzy if-then rules. Positions of the particles in the first generation are initialised with the rule weights obtained by the use of Eq. (3.2), (3.4) and (3.5). Particles are then iteratively modified towards the best solution with regard to a given quality measure over the set of rule weights. The fitness function of each particle is herein gauged by the classification accuracy that is entailed by the renewed fuzzy if-then rules. In summary, the algorithm using PSO to evolve the rule weights of an existing fuzzy classification system is presented in Algorithm 3.2.1, supported by Algorithm 3.2.2.

1 MAX_IT : number of maximum iterations;
2 $GOAL$: desired fitness value.
1: Initialisation
2: **repeat**
3: **for** each particle $i \in S$ **do**
4: **if** $f(x_i) < f(pBest_i)$ **then**
5: $pBest_i = x_i$
6: **end if**
7: **if** $f(pBest_i) < f(gBest)$ **then**
8: $gBest = pBest_i$
9: **end if**
10: **end for**
11: **for** each particle $i \in S$ **do**
12: **for** each dimension $d \in D$ **do**
13: $v_{i,d} = wv_x + c_1 r_1(x_{gBest} - x) + c_2 r_2(x_{pBest} - x)$
14: $x_{i,d} = x + \epsilon v_{i,d}$
15: **end for**
16: **end for**
17: $it{+}{+}$
18: **until** $it > MAX\_IT$ or $GOAL$ is achieved

Algorithm 3.2.1: Fuzzy rule refinement

1 S : number of particles;
2 D : number of dimensions equal to number of rules;
3 r_d : dth rule weight from existing rule base;
4 $f()$: fitness function used to evaluate particles.
1: **for** each particle $i \in S$ **do**
2: **for** each dimension $d \in D$ **do**
3: $x_{i,d} = r_d$
4: $v_{i,d} = Rnd(-v_{max}/3, v_{max}/3)$
5: **end for**
6: $pBest_i = x_i$
7: **if** $f(pBest_i) < f(gBest)$ **then**
8: $gBest = pBest_i$
9: **end if**
10: **end for**

Algorithm 3.2.2: Initialisation of fuzzy rule refinement

3.2.3 Learning Classifiers with PSO Refined Rule Weights

As a summary, Figure 3.7 shows the general framework of the proposed approach, for situations where the interpretability of fuzzy sets pre-defined by domain experts is required to remain unchanged. The initial rule base can be obtained by simple fuzzy grid partitioning [74] or other data-driven based methods [138, 157]. Specification of the initial rule weights can be obtained from a range of methods [77]. PSO is then directed to perform rule weights modification with the aim of improving the overall performance of the fuzzy classifier under consideration.

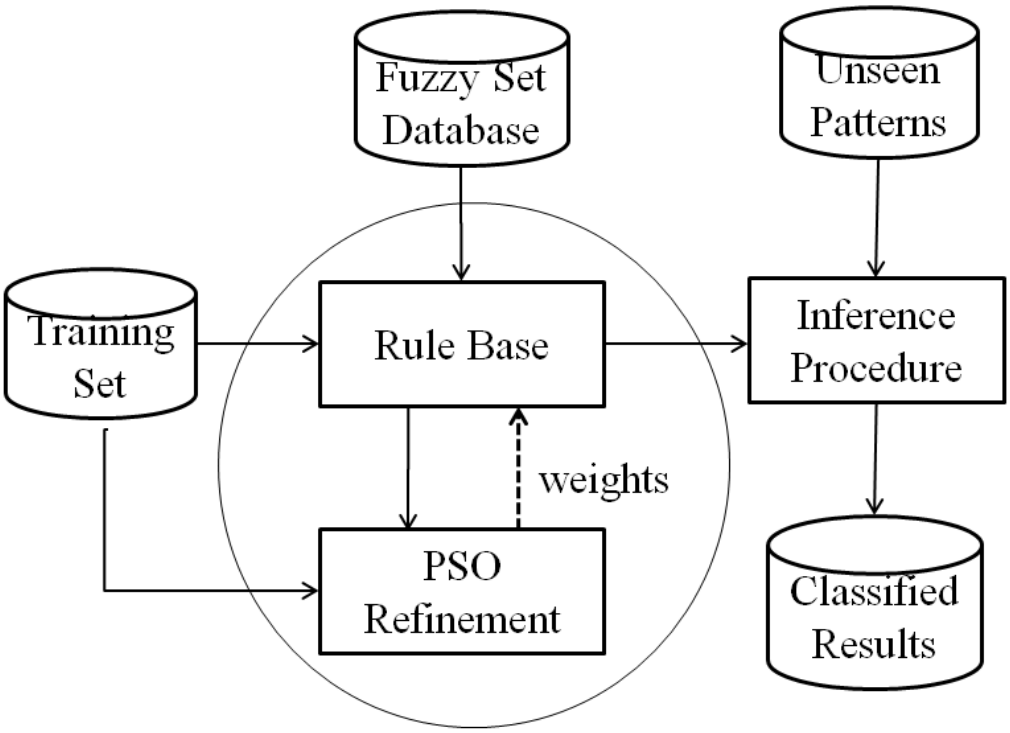

Figure 3.7: Framework of FRBCS with PSO refined rule weights

In terms of core algorithm complexity, during each PSO iteration a given set of rules, each of which is associated with an updated rule weight (which may remain the same as its original for certain rules), is reevaluated with regard to a global best set of weights achieved. Each training sample is required to match against each fuzzy rule with the updated rule weight to determine its classification result by the use of single winning rule strategy. The total computation effort required to accomplish

reevaluation is therefore in proportion to the product of the number of training data by the number of fuzzy rules, denoted as m and N respectively, namely $O(mN)$.

Obviously, in developing an FRBCS this way, a training data set is needed as input to this learning system, for both the generation of the initial rule base and the process of the rule weight refinement. For a given training set, the greater the number of initially built fuzzy rules, the more computation is needed to complete a PSO update process. The training of the FRBCS completes once the PSO-based refinement process terminates. Unseen patterns can then be classified by the trained classifier. Although the single winner rule strategy is adopted to classify patterns here, other inference methods (e.g., weighted vote) may also be employed if preferred [33]. Note that if after a training process is completed, a newly collected set of data becomes available then this set can also be utilised to train the classifier, with new rules integrated into the existing rule base. If implemented, the application of this idea would make the resulting FRBCSs dynamically adaptive, but this implementation remains as further work.

3.3 Experimentation and Validation

To demonstrate the potential of the proposed approach, a number of comparative experiments are carried out. The results are reported and discussed here, in terms of the effects of: (a) rule weighting schemes, (b) rule base sizes, and (c) rule learning methods used.

3.3.1 Experimental Setup

The PSO parameters are empirically specified in Table 3.1. Similar settings can be found in [40], with PSO parameter selection discussed in [129]. Note that as the main aim of this study is to examine the efficacy of applying PSO for fuzzy rule refinement instead of that of PSO itself, only the basic version of PSO is used in the experiments. The parameter specification for PSO is not carefully adjusted, therefore, simulation results could be further improved where more sophisticated versions of PSO are used with carefully modified parameters.

Table 3.1: Parameter values of PSO

w	c_1	c_2	ϵ	$Max\_Generation$	$Particle\_Numbers$
0.8	2.0	2.0	1.0	200	30

Initial rule weights are calculated via Eq. (3.2), (3.4) and (3.5), classification accuracies are computed with and without any initial heuristically produced rule weights respectively, in order to show how the rule weight refinement may affect the performance of the learned rules' accuracy. The purpose of this experimental design is to test how additional rule weight may affect the performance of a potential classifier, and how the proposed method may help improve such performance. Note that several popular rule-based learning classifiers are also selected for comparison. This is to demonstrate that simple FRBCSs which employ a rule base whose individual rule weights are modified with a PSO process are competitive in their performance as with popular rule-based classifiers available in the literature.

In order to examine the effect of using PSO-refined rule weights upon the improvement of fuzzy partition quality, four different fuzzy partitions are tested, where each of the pattern spaces is uniformly divided into K ($K = 2, 3, 4, 5$) triangular fuzzy subsets in the same way as that shown in Figure 3.1. This allows the performance of the proposed method to be investigated for fine fuzzy partitions as well as coarse fuzzy partitions. In particular, the case of $K = 2$ represents a very rough partition, while that of $K = 5$ represents a very detailed partition. Similar partitions can be found in [79]. Note that given a K, in theory, the total number of fuzzy if-then rules for each fuzzy partition would be K^n, where n stands for the number of input attributes, however a fuzzy rule will only be generated when there is a training pattern covered by an emerging rule. So, the total number of rules produced is typically smaller.

Owing to a large amount of systematic experimental investigation being carried out, only stratified twofold cross-validation (2-CV) is employed for data validation in this work. In 2-CV, a given data set is partitioned into 2 subsets. One of the subsets is used to train a fuzzy classifier, where the proposed approach is used to refine corresponding fuzzy rule weights. Another divided subset is retained to testing data to produce a single accuracy value. The process is then repeated 30 times by initialising different, randomly assigned seeds to produce the final average outcomes. Pairwise t-tests are run with $p < 0.05$. Results are thus measured in terms of the

significance of differences between different learning classifiers, with the achieved accuracy of PSO-FR (i.e., the proposed approach) as the reference in each experiment. Those results that are significantly better, worse or of no difference are marked with "(v)", "($*$)", or "(—)", respectively.

3.3.2 Effect of Rule Weighting Scheme

In Table 3.2, PSO-FR, FR, and H-FR stand for the application of fuzzy rules with PSO-refined rule weights, that of fuzzy rules without rule weights, and that of fuzzy rules with heuristic rule weights initially provided, respectively. Experiments are performed on 12 real-valued benchmark data sets [11], the characteristics of which can be found in Appendix B.

As shown in Table 3.2, H-FR outperforms FR in terms of average classification accuracy, regardless of the number of fuzzy partitions, for 7 out of 12 data sets (including ecoli, iris, image, liver-disorders, new-thyroid, parkinsons, and prnn-synth). For the other 5 data sets, the results of H-FR are competitive to those of FR. This is not surprising since the rule weights used in H-FR are heuristically initialised, reflecting the fact that general information on the significance of the corresponding rules has been exploited in building the rule-based learning classifier. This conforms to what is explained in Chapter 3.2.1 regarding the influence of rule weights upon classification boundaries.

Although H-FR generally achieves better results than FR, the performance of H-FR is still far from ideal. Fortunately, as illustrated in Table 3.2, the results of PSO-FR are significantly better those achievable by H-FR for 33 times and worse for just once, with 14 ties. This superior performance of learnt fuzzy classifiers with PSO refined rule weights is reinforced by Figure 3.8, which systematically depicts the relation between the PSO iteration number and the accuracy of a learnt classifier for each of the simulated data sets. In this figure, 12 sets of plots are shown each representing the results on one data set for both training and testing performance using 4 different fuzzy partitions, namely $k = 2, 3, 4, 5$. Generally, for both training and testing data, each FRBCS with the current PSO-returned rule weights starts from their initial performance, through an oscillatory process, and then reaches a steady state with a noticeable degree of improvement.

Table 3.2: Comparison using 30 × 2 cross-validation with respect to classification accuracy (%), where v, $-$ or $*$ indicate statistically better, same or worse results, respectively, and bold figures signify overall best results for each data set with a certain partition number.

Data Sets	K	Rule Number	PSO-FR	FR	H-FR	J48	PTTD	QSBA
ecoli	2	25.97	**78.28**±1.97	72.83±1.29(*)	75.98±1.65(*)	74.97±1.29(*)	76.21±2.19(*)	23.29±6.54(*)
	3	39.30	**80.49**±2.10	72.21±1.39(*)	74.99±1.49(*)	77.40±1.93(*)	76.24±1.48(*)	19.59±7.86(*)
	4	56.58	**81.53**±1.75	80.78±1.86(*)	81.47±1.51(-)	75.34±2.40(*)	78.53±2.02(*)	58.64±3.09(*)
	5	84.42	79.58±1.84	79.04±2.37(*)	**79.65**±2.18(-)	75.65±1.90(*)	77.97±1.92(*)	69.03±3.19(*)
glass	2	23.98	**60.67**±4.77	49.42±3.22(*)	52.23±4.11(*)	52.13±2.62(*)	59.14±2.88(*)	28.30±4.56(*)
	3	31.48	61.12±2.50	57.90±2.40(*)	55.51±4.21(*)	59.03±3.11(*)	**61.57**±4.01(*)	36.08±3.88(*)
	4	40.82	54.25±4.41	48.33±3.44(*)	49.10±3.37(*)	57.99±3.37(v)	**59.31**±4.00(v)	37.88±3.90(*)
	5	56.70	58.07±3.03	54.31±2.95(*)	58.47±2.91(-)	57.54±3.89(-)	**63.93**±3.63(v)	45.83±4.34(*)
haberman	2	3.57	**74.07**±1.07	73.10±0.27(*)	73.27±0.43(*)	73.35±0.48(*)	72.41±1.63(*)	72.57±4.23(*)
	3	6.50	74.02±1.47	73.28±1.05(*)	73.14±0.74(*)	73.33±0.67(*)	73.35±1.43(*)	**74.32**±1.16(-)
	4	8.87	73.65±1.39	**75.52**±1.17(v)	74.18± 1.10(v)	73.24±0.67(-)	74.90±1.38(v)	73.77±2.87(-)
	5	13.47	**74.18**±1.64	73.33±1.51(*)	73.77±1.39(*)	73.16±0.80(*)	73.81 ±1.08(-)	73.29±1.29(*)
image	2	37.05	72.49±3.33	69.41±3.53(*)	70.37±3.23(*)	**74.78**±2.13(v)	64.98±3.92(*)	55.32±1.55(*)
	3	65.15	74.68±2.38	70.86±2.59(*)	73.54±2.45(*)	76.49±2.70(v)	**80.98**±2.57(v)	59.14±6.81(*)
	4	86.60	76.44±2.74	76.57±2.89 (-)	76.57±2.74(-)	**82.70**±2.28(v)	80.97±2.37(v)	72.09±7.56(*)
	5	93.07	72.41±2.11	72.40±2.06(-)	72.51±2.11(-)	80.46 ±2.53(v)	**83.68**±2.15(v)	74.45±6.92(-)
iris	2	7.98	**92.33**±2.90	72.04±2.00(*)	84.58±2.64(*)	76.65±2.76(*)	77.49±2.27(*)	66.67±0.00(*)
	3	14.75	95.16±1.60	91.56±1.37(*)	93.89±0.91(*)	**95.33**±1.19(-)	92.18±0.89(*)	62.11±1.63(*)
	4	22.38	**93.02**±1.93	78.18±2.36(*)	85.60±2.75(*)	90.09±3.12(*)	90.78±3.80(*)	62.11±1.71(*)
	5	30.60	93.09±1.66	93.00±1.33(-)	93.22±0.89(-)	91.53±2.37(*)	94.73 ±0.98(v)	**94.91**±0.93(v)
liver-disorders	2	13.97	**58.45**±2.07	56.10±1.40(*)	57.72±1.53(*)	56.98±1.62(*)	58.20±2.01(-)	47.18±3.03(*)
	3	35.80	59.07±2.89	52.10±2.59(*)	56.45±2.41(*)	56.91±1.40(*)	**59.71**±3.16(-)	46.07±1.54(*)
	4	56.53	59.52±3.08	54.64±2.93(*)	56.26±2.80(*)	56.25±2.26(*)	**60.03**±2.55(-)	53.07±3.22(*)
	5	82.30	58.14±2.67	56.24±2.55(*)	56.75±1.91(*)	56.11±2.34(*)	**63.51**±2.78(-)	57.92±3.33(-)
new-thyroid	2	6.87	**91.18**±1.49	83.97±1.19(*)	85.13±1.26(*)	84.85±1.79(*)	83.71±0.68(*)	87.21±2.76(*)
	3	16.88	**91.13**±2.31	88.34±1.40(*)	89.30±1.41(*)	86.25±1.62(*)	87.06±1.09(*)	89.71±2.61(*)
	4	25.78	91.92±1.77	90.32±1.54 (*)	91.60±1.03(-)	88.50±2.41(*)	92.34±0.87(-)	**93.38**±0.70(v)
	5	33.33	91.16±1.44	88.65±1.95(*)	91.09±1.34(-)	90.90±1.56 (-)	89.61±1.35(*)	**92.67**±0.80(v)
parkinsons	2	57.07	86.10±2.19	79.15±2.12(*)	83.85±2.29(*)	82.37±2.47(*)	**86.19**±1.35(-)	54.75±5.06(*)
	3	79.67	81.47±2.45	79.83±2.11(*)	80.72±2.48(*)	**86.61**±2.01(v)	83.72±1.44(v)	77.09±0.86(*)
	4	88.12	84.74±3.27	84.80±2.82(-)	84.86 ±2.92(-)	84.35±2.89(-)	**85.39**±1.87(-)	76.97±0.90(*)
	5	93.38	84.78±3.77	84.71±3.80(-)	84.75±3.77(-)	84.48±2.02(-)	**86.46**±1.87(v)	77.88±1.74(*)
pima-diabetes	2	36.32	**73.19**±1.57	66.57±1.13(*)	69.34±0.92(*)	68.09±1.28(*)	69.65±1.40(*)	70.53±0.66(*)
	3	84.08	70.30 ±1.38	70.72±1.56(-)	69.83±1.41(*)	72.70±1.00(v)	**73.83**±0.44(v)	61.52±1.60(*)
	4	201.70	69.67±1.66	68.14±1.34(*)	69.51±1.60(-)	**73.94**±0.82(v)	71.85±1.66(v)	58.29±1.34(*)
	5	269.70	67.59±1.86	66.65±1.85(*)	67.50±1.86(-)	**74.40**±1.04(v)	73.55±1.05(v)	69.05±0.71(*)
prnn-synth	2	4.00	**83.67**±1.83	80.97±0.20(*)	81.25±1.54(*)	81.83±1.12(*)	81.83±1.12(*)	51.68±1.99(*)
	3	7.90	**83.45**±1.29	70.08±3.46(*)	80.47±1.13(*)	77.03±2.19(*)	72.04±2.47(*)	71.20±3.40(*)
	4	12.52	83.32±2.21	82.28±0.91(*)	84.23±1.15 (v)	83.28±1.78(-)	**84.24**±1.21(v)	83.19±1.29(-)
	5	16.52	82.65±1.63	79.36±2.35(*)	83.29±1.21(v)	82.76±1.52(-)	80.60±2.12(*)	**83.52**±1.04(v)
seeds	2	16.22	90.06±1.86	88.21±0.98(*)	86.49±1.26(*)	87.33±1.79(*)	**92.16**±1.34(v)	33.39±0.35(*)
	3	41.92	**89.98**±1.83	79.67±1.55(*)	85.95±2.26(*)	84.44±2.06(*)	86.35±1.55(*)	41.13±1.27(*)
	4	56.17	**89.49**±1.74	88.08±1.24(*)	88.95±1.38(*)	87.84±2.47(*)	88.35±1.55(*)	62.22±1.45(*)
	5	75.57	88.70±1.61	87.89±1.71(*)	88.06±1.67(*)	86.94±1.56(*)	**88.78**±1.52(-)	73.08±1.61(*)
yeast	2	34.40	47.58±1.84	38.49±0.51(*)	38.79±2.53(*)	39.57±1.79(*)	**50.78**±0.92(v)	14.22±3.87(*)
	3	83.73	**54.90**±0.95	51.56±0.66(*)	53.17±0.97(*)	52.34±1.49(*)	50.63±1.67(*)	10.45±2.15(*)
	4	90.42	**52.70**±1.16	42.05±0.92(*)	49.76±1.23(*)	51.32±1.68(*)	52.19±1.21(-)	32.53±2.36(*)
	5	164.65	50.33±1.45	47.31±1.29(*)	48.30±1.04(*)	49.42±1.49(*)	**51.90**±1.98(v)	46.89±1.66(*)

3.3.3 Effect of Rule Base Size

From Figure 3.8, further observations can be obtained. For better viewing, the accuracy of each classifier is displayed for every 3 iterations within a total of 100 iterations, each point is the average of the results from 30 runs of 2-CV. As can be seen, after an initial period of oscillations, generally the trend of the training performance for all FRBCSs tend to converge at around $40th$-$60th$ iteration regardless of the number of fuzzy partitions. In terms of testing accuracies, the curves are generally more oscillatory than the training ones. Although the testing accuracies do not reach so high as that is achievable over the training phase, they are significantly improved over the original performance.

Looking more carefully, it is interesting to note that in general, fuzzy classifiers modelled with a lower number of partitions tend to have a poor performance at the beginning for both training and testing curves, likely due to the coarse partitioning of the input spaces. However, in terms of testing curves, although coarse partitioned ones (e.g., $K = 2$) have a lower start, their performance can outperform the finer partitioned ones (e.g., $K = 5$), not just catching up with them, particularly for diabetes, glass, haberman, liver-disorders, parkinsons, prnn-synth, seeds, and thyroid (8 out of 12 data sets). For finer partitions, which generally have a better start in performance, the classification accuracy does not improve so much as lower partitioned ones when converged, and even underperformed than those models with a lower number of partitions in 11 out of 12 data sets: diabetes, ecoli, glass, iris, image, liver-disorders, parkinsons, prnn-synth, seeds, thyroid, and yeast. Although a finer partitioned fuzzy classifier with more initial rules is likely to have a head start regarding performance, the resultant larger search space may in turn make final solutions converge in a local minimal, thus achieving even worse results than those of coarsely partitioned ones.

With simple fuzzy grid partitioning in the generation of the initial rule base, finer partitions of the input spaces lead to more fuzzy rules, as clearly indicated from the rule numbers in Table 3.2. The more fuzzy rules generated initially, the more rule weights need to be modified, and hence the larger the search space is and the higher the computational complexity the PSO process involves. Besides, as observed above, a finer partitioned fuzzy classifier normally achieves worse performance. One possible reason for such seemingly unintuitive results is overfitting during the

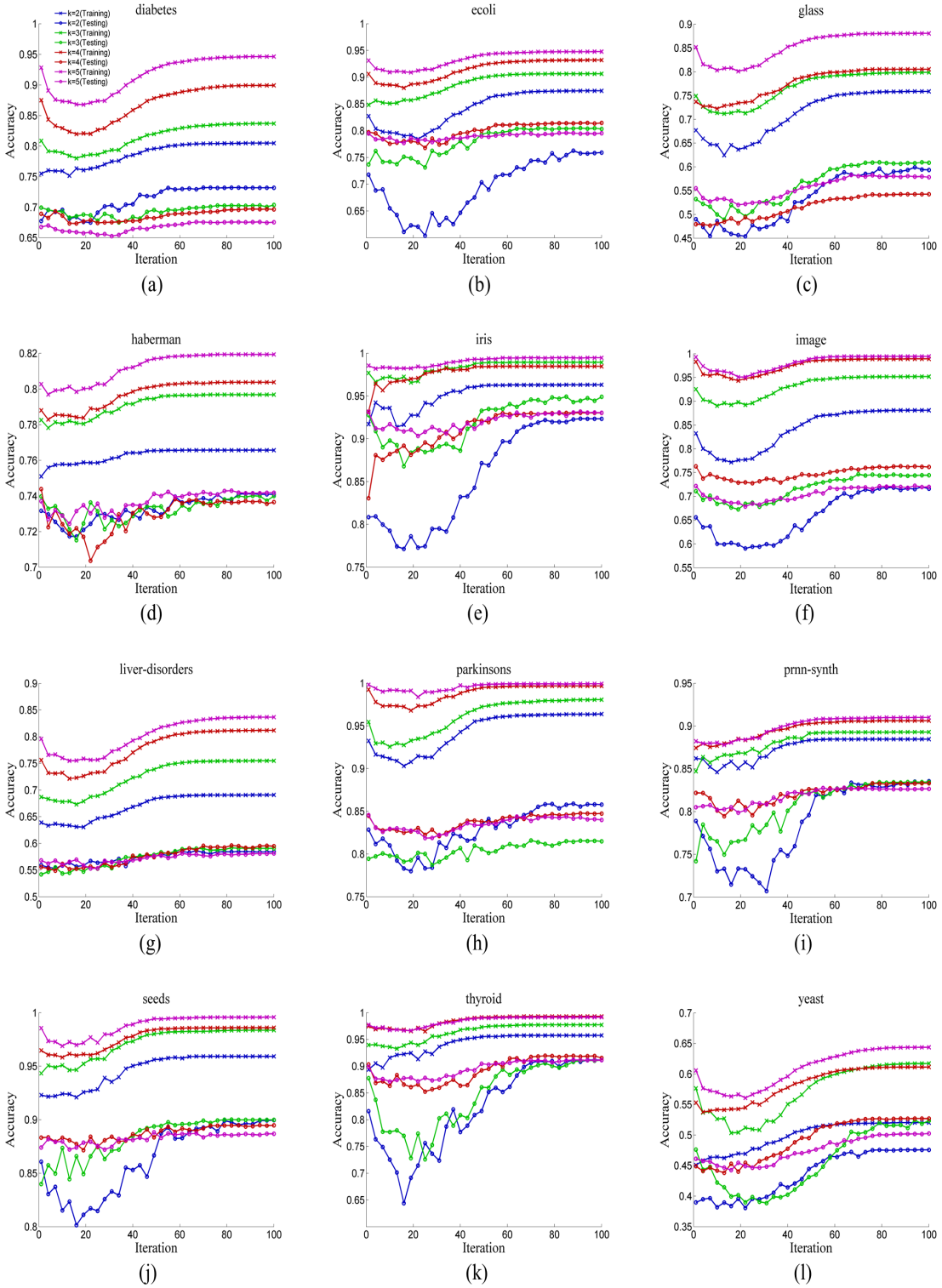

Figure 3.8: Relation between PSO iteration number and classification performance

training. Therefore, it would be worthwhile to consider the number of fuzzy rules as part of the criteria in constructing the fitness function, by penalising emerging models that consist of more rules or by filtering poor quality individual rules (e.g., low coverage or low performance). The implementation of such ideas remains as future work.

3.3.4 Effect of Rule Learning Method

Three classifier learning algorithms that generate models in the form of a rule set are chosen to perform classification tasks for comparison purpose. These are: the popular C4.5 decision tree learner (J48) [124], the top-down fuzzy pattern trees (PTTD) [135], and the fuzzy subsethood-based rule models with quantifiers (QSBA) [127]. In order to reduce the runtime of PTTD and to have a fair comparison, only the algebraic t-norm and maximum s-norm are chosen as fuzzy operators in this work, which are similar to the operators used in the proposed approach herein (see Eq. (4.10), (3.7)). Fuzzy quantifier-based models are generated using fuzzy quantification to replace crisp weights in subsethood-based fuzzy rule models, which are not only interpretable but also practically applicable [125], [126]. Note that the same fuzzy pre-partition of the input space is adopted for both PTTD and QSBA as that for the proposed method, whereas the same partitioning interval is chosen as the corresponding variable discretisation for J48. All these algorithms are implemented within the WEKA machine learning framework [162] with default parameter setting unless otherwise stated previously.

The winning results in terms of achieving the highest classification accuracy per learning classifiers are highlighted in boldface in Table 3.2. It is important to note that the proposed method (PSO-FR) has 18 wins, compared with 17 wins by PTTD, 6 wins by J48, and 5 wins by QSBA. Obviously the proposed approach significantly outperforms J48 and QSBA, and is competitive to PTTD. Between the two better performers, PSO-FR and PTTD, a specific comparison can be made from the results obtained. Statistically, the proposed method wins 22 times and loses 16 times with 10 ties over PTTD. These results jointly demonstrate that the present work is at least competitive to the state-of-the-art rule-based classifiers in the literature regarding classification accuracy.

3.3.5 Effect of Imbalanced Data

Conventional evaluation of classification performance using a criterion like the overall accuracy does not always provide adequate assessment in cases that involve imbalanced data. In order to examine the performance of the proposed approach with regard to imbalanced data sets (e.g., the yeast dataset seen earlier), confusion matrices (also known as contingency tables) are typically used. A confusion matrix is an $M \times M$ matrix with each column representing the number of instances in a predicted class, and each row representing the number of instances in an actual class.

Motivated by the above, this section further reports, as an example, a further experimental investigation of the present work, based on the confusion matrices computed over the yeast dataset. This dataset is selected because it includes 10 imbalanced classes. Tables 3.3 and 3.4 show the results obtained from the runs with the partition number (K) set to 2, for the input space using H-FR and PSO-FR, respectively.

As can be seen, the proposed method improves the results greatly mainly in classes of majority instances (e.g., Class 1 with total 463 instances from 177 to 310, and Class 2 with total 429 instances from 191 to 229). For classes with minority instances, results remain almost the same or even deteriorate (e.g., Class 8 with total 30 instances from 0 to 0, Class 7 with total 35 instances from 1 to 0). This is due to the fact that the proposed algorithm employs overall accuracy as the fitness function. One possible way to resolve this problem is to embed a cost matrix into the calculation of accuracy as the fitness function, such that the objective of PSO refinement becomes to develop a set of weights that minimise the overall cost on the training set[62].

3.4 Summary

This chapter has proposed an approach for fuzzy rule weight refinement by the use of PSO. The approach works for situations where an initial rule fuzzy rule-base has been built with predefined fuzzy sets, which are required to be maintained for the purpose of consistent interpretability, both in the learned models and in the inference

Table 3.3: Confusion matrix of $H-FR$ on yeast data set with a random seed and $K=2$ for input space

Class	1	2	3	4	5	6	7	8	9	10	Total
1	**177**	156	86	1	40	0	0	0	1	2	463
2	146	**191**	67	2	21	1	0	0	0	1	429
3	35	57	**143**	3	4	0	0	0	2	0	244
4	82	47	20	**2**	12	0	0	0	0	0	163
5	15	12	19	0	**2**	1	2	0	0	0	51
6	1	28	9	1	2	**2**	1	0	0	0	44
7	2	12	16	0	2	2	**1**	0	0	0	35
8	13	8	7	1	1	0	0	**0**	0	0	30
9	1	2	8	0	1	0	0	0	**8**	0	20
10	0	0	0	0	0	0	0	0	0	**5**	5
Total	472	513	375	10	85	6	4	0	11	8	1484

Table 3.4: Confusion matrix of $PSO-FR$ on yeast data set with a random seed and $K=2$ for input space

Class	1	2	3	4	5	6	7	8	9	10	Total
1	**310**	112	31	0	0	9	0	0	1	0	463
2	161	**229**	31	0	0	6	0	0	0	2	429
3	72	26	**134**	0	0	10	0	0	2	0	244
4	68	78	7	**0**	0	10	0	0	0	0	163
5	29	3	10	0	**0**	9	0	0	0	0	51
6	0	0	7	0	0	**37**	0	0	0	0	44
7	6	1	3	0	1	24	**0**	0	0	0	35
8	16	6	2	0	0	6	0	**0**	0	0	30
9	6	1	4	0	0	0	0	0	**9**	0	20
10	0	0	0	0	0	0	0	0	0	**5**	5
Total	668	456	229	0	1	111	0	0	12	7	1484

results using such models. Systematic experimental results have demonstrated the following:

1. The performance of a fuzzy rule-based classifier can be significantly improved with rule weight refinement implemented by PSO.

2. The size of an initially built rule base may affect the performance of the proposed method, although optimisation of the initial fuzzy quality space will help reduce such influence.

3. The proposed approach is at least competitive to typical state-of-the-art learning classifiers even when only simple fuzzy grid partitioning is used to create the initial rule base.

Chapter 4

Induction of Quantified Fuzzy Rules with Particle Swarm Optimisation

QUANTIFICATION has been regarded as an important topic in fuzzy theory and its applications [37]. The use of fuzzy quantifiers by attaching semantic labels to fuzzy sets can be seen as flexible tools for the representation of natural language, making the existing fuzzy models more readable and accurate [107]. Fuzzy quantifiers could also be used to deal with situations where the information dealt with is not equally important. For example, when evaluating a student performance, scores of assignments and final exams will probably take different weights in determining the student's final grade. A certain weighting strategy to represent the degrees of significance among antecedent attributes may therefore be necessary [63].

Crisp weights attached to fuzzy linguistic variables could be used to improve the classification accuracy of fuzzy models. Yet the use of non-fuzzy values with fuzzy terms may lead to confusion regarding the linguistic interpretation of a given fuzzy model. Replacing crisp weights with fuzzy quantifiers also helps improve the interpretability of the learned models, while guaranteeing any inferred results to remain in consistent fuzzy representation. In the literature, a number of definitions of fuzzy quantifiers have been proposed [37, 95], including both absolute and relative quantifiers. An example of this is subsethood-based fuzzy rule modelling that has been developed for classification tasks [127]. Furthermore, quantifier-based fuzzy classification systems have been successfully applied in addressing different

problems, including the evaluation of student academic performance [128] and medical diagnosis [142]. Such applications not only provide promising classification performance but also practically understandable rule sets for further reference.

Motivated from the potential of quantifier-based fuzzy classification, this chapter proposes a metaheuristic algorithm-based approach that can learn a set of rules with continuous fuzzy quantifiers. The work allows a set of quantified fuzzy rules to be combined and evaluated simultaneously during the learning process. In particular, Particle Swarm Optimisation [87] is employed as the metaheuristic algorithm to evolve rule sets and fuzzy quantifiers subject to the overall quality of an emerging rule base. As an initial implementation to test the ideas, the performance of resulting fuzzy rules with and without fuzzy quantifiers is assessed on various UCI benchmark data sets, in comparison to popular rule based learning classifiers. Experimental results demonstrate that rule bases generated by the proposed approach can boost classification performance as compared to those without fuzzy quantifiers while being competitive to those popular rule based classifiers.

The reminder of this chapter is organised as follows. Section 4.1 introduces fuzzy classification rules and the representation of continuous fuzzy quantifiers, together with the class-dependent simultaneous rule induction strategy. Section 4.2 demonstrates how PSO particles may be used to encode fuzzy rule bases and how each rule base is evaluated and updated. Section 4.3 presents and discusses the experimental results. Section 4.4 concludes the chapter and outlines ideas for further development.

4.1 Preliminaries

4.1.1 Fuzzy Quantifiers

The task of learning an FRBCS is to find a finite set of fuzzy if-then rules capable of reproducing the input-output behaviour of a given system or process. Without losing generality, the system to be modelled is herein assumed to be a multiple-input-single-output, containing n inputs and one output and involving m patterns for an M-class problem. A fuzzy if-then rule $R_j, j = 1, 2, ..., N$, for such a system is represented as follows:

$$\text{If } x_1 \text{ is } A_{j1} \text{ and } ... \text{ and } x_n \text{ is } A_{jn} \text{ then class is } C_h \tag{4.1}$$

where $x_1, x_2, ..., x_n$ are the underlying linguistic variables, jointly defining an n-dimensional pattern space (with N obviously denoting the number of such fuzzy rules); A_{ji}, $i \in \{1, 2, ..., n\}$, is the fuzzy value of the corresponding antecedent x_i; C_h, $h \in \{1, 2, ..., M\}$, is the consequent class for the M class problem.

As with many existing techniques for representing weights, measures of weighting are limited to the normal range of 0 to 1, with 0 representing the lowest weight and 1 the highest. Weights attached to linguistic terms provide a multiplication factor in the compound fuzzy propositions. They reveal relative contributions made by different linguistic terms (and thereby the underlying antecedent variables) towards the conclusion drawn. Such a rule is represented as follows:

$$\text{If } x_1 \text{ is } w_{j1}A_{j1} \text{ and } ... \text{ and } x_n \text{ is } w_{jn}A_{jn} \text{ then class is } C_h \tag{4.2}$$

where w_{ji}, $i \in \{1, 2, ..., n\}$ is a crisp weight of the corresponding linguistic term A_{ji}.

The interpretation of the compound fuzzy proposition $(w_{j1} \times A_{j1})$ is restricted with respect to a practical application. In [127], fuzzy quantifiers are used to replace crisp weights to improve the transparency of the learnt fuzzy systems. In general, quantification in logic may be expressed as $Q(x)A(x)$ where $Q(x)$ is a quantifier and $A(x)$ is a predicate for variable x [37]. As small changes in the training set can cause a change to the entire rule set, fuzzy models that employ continuous fuzzy quantifiers may therefore be more appropriate compared to the use of two or multi-valued crisp quantifiers.

In particular, a fuzzy relative quantifier Q, where $\mu_Q(q) \in [0, 1]$ with q defined on real interval $[0, 1]$, processes the non-decreasing behaviour: $\forall q_1, q_2 \in Q, q_1 < q_2 \rightarrow \mu_Q(q_1) \leq \mu_Q(q_2)$. An example of a relative quantifier is "Most students who get a high score are young", where the "most" is the quantifier, and the "high" and "young" are fuzzy values. In [156] a continuous fuzzy quantifier is proposed which applies linear interpolation between the two classical, extreme cases of the existential quantifier $\exists$ and the universal quantifier $\forall$, such that:

$$Q(E, A) = (1 - \lambda_Q).T_{\forall, A/E} + \lambda_Q.T_{\exists, A/E} \tag{4.3}$$

In this definition, Q is the quantifier for the fuzzy set A relative to fuzzy set E and λ_Q is the degree of orness [167] of the two extreme quantifiers. Following this, two

popular quantifiers, the existential quantifier $T_{\exists,A/E}$ and the universal quantifier $T_{\forall,A/E}$ can be represented as:

$$T_{\exists,A/E} = \triangle_{K=1}^{N} \mu(e_k) \nabla \mu(a_k) \tag{4.4}$$

$$T_{\forall,A/E} = \nabla_{K=1}^{N} (1 - \mu(e_k)) \triangle \mu(a_k) \tag{4.5}$$

where a_k and e_k are the membership functions of fuzzy sets A and E respectively, ∇ denotes the *t-norm* and $\triangle$ denotes the *t-conorm*.

The use of such fuzzy quantifiers enables the representation of a fuzzy rule in a more natural way:

$$\text{If } x_1 \text{ is } Q_{j1}A_{j1} \text{ and ... and } x_n \text{ is } Q_{jn}A_{jn} \text{ then class is } C_h \tag{4.6}$$

where Q_{ji}, $i \in \{1, 2, ..., n\}$ is a fuzzy quantifier modifying the linguistic term A_{ji}. Importantly, the use of *t-norm* operators to interpret $\nabla(Q_{ji}, A_{ji})$ guarantees that the inference results are also fuzzy sets.

4.1.2 Strategy of Simultaneous Rule Induction

The sequential covering algorithm or separate-and-conquer strategy is one of the most widespread approaches to learning disjunctive sets of rules for classification problems [50]. Generally, this covering algorithm takes each class in turn and seeks a set of rules covering positive instances for a certain class. Positive instances covered by a learned rule are removed, and subsequent rules are learnt based on the remaining training instances. This procedure is iterated until all positive instances are covered by the rules created so far. In literature, well-known rule induction approaches for the generation of crisp rule bases developed on the basis of this strategy include PRISM [20], FOIL [123], and RIPPER [31].

Despite the popularity of this strategy, a problem may be encountered when trying to extend it to learning fuzzy rules for FRBCSs. Unlike crisp rules, all fuzzy rules may match or cover all cases within a training set if fuzzy sets with an infinite support such as those of a Gaussian form are used, but to varying degrees. This may lead to a situation where a case requiring classification is closely matched by two or more rules with different conclusions. Having a final rule base of complementary

rules is therefore potentially beneficial to the fuzzy inference or classification process. To reflect this observation, an approach is proposed in [53], which runs a number of Ant Colony Optimisation (ACO) algorithms simultaneously, with each focusing on finding descriptive rules for a specific class. After each class has had its associated rules created during one iteration, all possible combinations of rules, with one from each class are formed into a rule base and is re-tested on the training set. The rules in the best performing rule base are used to update the pheromone levels, within the underlying iterative ACO process. Such a simultaneous fuzzy rule induction strategy is also adopted in this paper and is generally described in Algorithm 4.1.1.

```
1:  for numInterations do
2:     for each class  do
3:        each agent constructs a fuzzy rule
4:     end for
5:     for each combined rule base do
6:        evaluate each rule base
7:     end for
8:     update agents with best rule base
9:  end for
10: output best rule base
```

Algorithm 4.1.1: Strategy of Simultaneous Rule Induction

4.2 Induction of Quantified Fuzzy Rules with Particle Swarm Optimisation

4.2.1 Encoding Quantified Fuzzy Rules with PSO Particles

As reviewed in Section 2.2.2, PSO [87] is able to provides a simple but effective mechanism for conducting global search that requires minimum understanding of the problem domain, while involving only simple real number encoding. It is therefore is employed as the metaheuristic algorithm to evolve rule sets and fuzzy quantifiers. To suit the present application, of obtaining an optimal set of rules, the PSO specification needs adaptation. In particular, each of the PSO particles is set to encode a quantified fuzzy rule base, with each particle dimension representing a single fuzzy rule with quantifiers. As the first attempt to evaluate this initial work, a simplified version of

the simultaneous rule induction strategy is adopted by just encoding one rule for each class. This is based on the assumption that one rule is sufficient to describe a class [53]. This assumption is realistic as all fuzzy rules may be presumed to match or cover all cases, but to varying degrees. In theory, however, various particle dimensions may be used to encode multiple fuzzy rules for each class.

Generally speaking, the proposed method adopts Pittsburgh-style representation, which encodes an entire fuzzy rule base as a PSO particle. Each quantified fuzzy rule is encoded by one PSO particle dimension. The dimensionality of a single PSO particle is set to the same as that of rule base, which consists of multiple fuzzy classification rules for different classes. As a simplified version with the assumption of one rule per class, the dimensionality is simply set to the same as the number of classes given for the problem domain. Each PSO particle dimension is initialised with an array of positive real numbers, where each array element encodes a certain linguistic term and its associated fuzzy quantifier, corresponding to a compound fuzzy proposition. The dimensionality of a PSO particle dimension is therefore dependent on the number of attributes provided.

In representation, each positive real number $r \in \mathbb{R}^+$ initialised as an element of a PSO particle dimension, is separated into the integer part $int(r) = \lfloor x \rfloor$ and the fractional part $frac(r) = r - \lfloor r \rfloor$. The hierarchical structure of encoding a quantified fuzzy rule with a PSO particle is specified in Figure 4.1. A certain PSO particle P^l denotes a fuzzy rule base, with P^l_h being a PSO particle dimension representing a quantified fuzzy rule for the consequent class h given an M-class problem, where $h \in \{1, 2, ..., M\}$. Each PSO particle dimension representing rule R_j is then initialised with an array of positive real numbers, such that each array element P^l_{hi} encodes information for both the fuzzy set A_{ji} and its quantifier Q_{ji}. In particular, the integer part denotes a corresponding fuzzy set A_{ji}, where $i = int(P^l_{hi}), i \in \{1, 2, ..., n\}$, and the fractional part represents the quantifier Q_{ji} with regard to the fuzzy value A_{ji} as the degree of orness of one of the two extreme quantifiers (i.e., the existential and universal quantifier). Such an encoding scheme therefore transforms the task of finding an optimal set of fuzzy rules into that of obtaining an optimal PSO particle regarding a linguistic rule base.

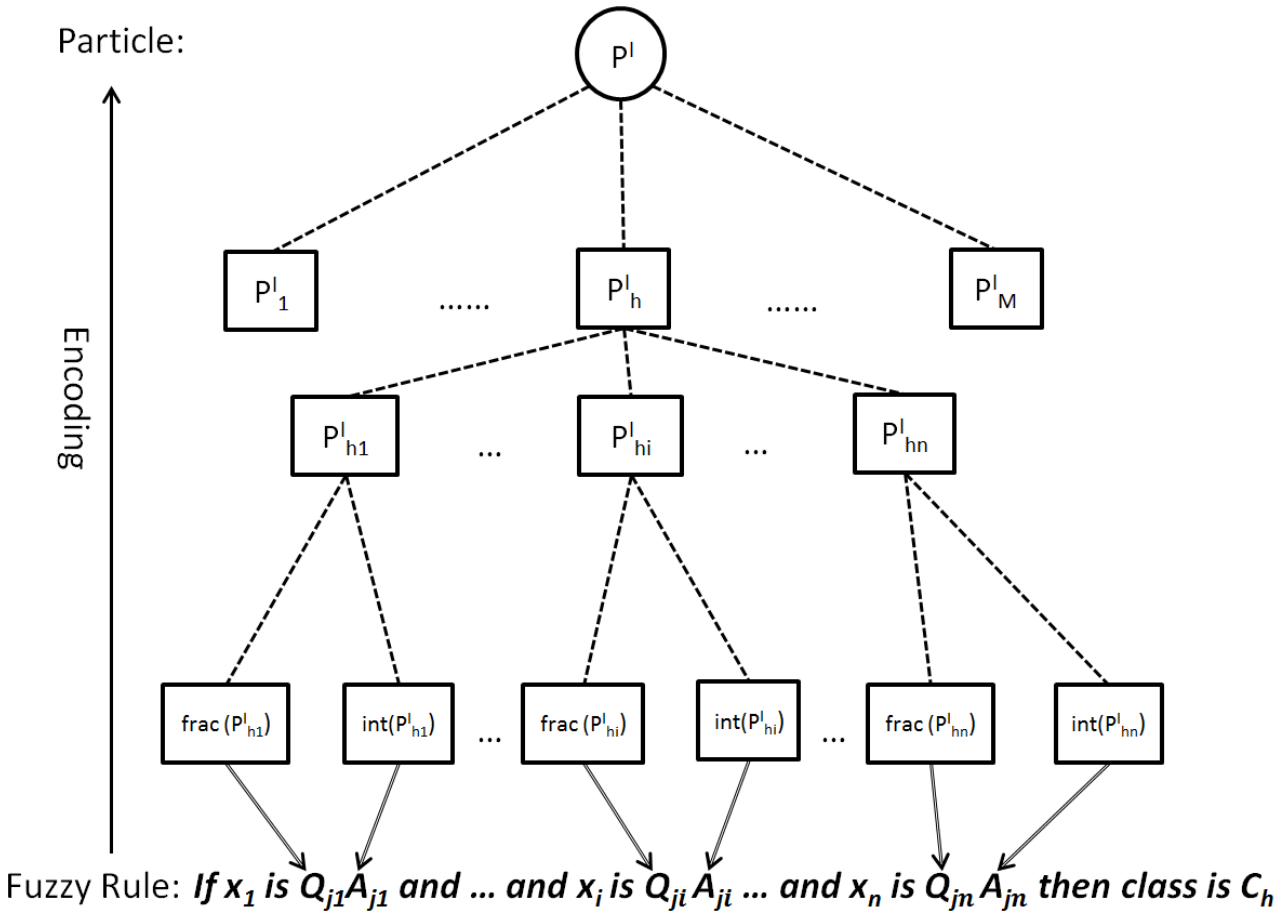

Figure 4.1: Hierarchical structure of a quantified fuzzy rule encoded with PSO

4.2.2 Evaluating Quantified Fuzzy Rules

The matching degree of each instance is calculated, with regard to each quantified fuzzy rule R_j, initialised via an array of real numbers. The range of such a real number is set to $r \subseteq [0, K+1)$, where K denotes the number of pre-defined fuzzy sets for an antecedent attribute, with the interpretation that each integer $int(r)$ corresponds to the $int(r)$-th pre-defined fuzzy set. The membership degree of an attribute A_{ji} is retrieved from the quantity space with respect to the given integer. This situation is excluded when $int(r) = 0$, indicating the absence of that particular attribute, with the corresponding attribute matching degree set to 1, with the interpretation being that the attribute is irrelevant. In terms of calculating the quantifier Q_{ji} associated with A_{ji}, the fractional part is then used to represent the orness of the two extreme quantifiers $\exists$ and $\forall$, following Eqn. 4.3. Thus, the quantifier Q_{ji} is updated such that:

$$Q(E,A) = (1 - frac(x_{ji}))T_{\forall,A/E} + frac(x_{ji})T_{\exists,A/E} \tag{4.7}$$

where $frac(x_{ji})$ is the fractional part of that position, and the truth value of the existential quantifier $T_{\exists,A/E}$ and that of the universal quantifier $T_{\forall,A/E}$ are calculated according to Eqn. 4.4 and Eqn. 4.5, respectively. *T-norm* operators are then used to interpret $\nabla(Q_{ji}, A_{ji})$, which guarantees that the inference results are also fuzzy sets.

In general, the system to be modelled is assumed to involve n antecedent attributes and M classes, with m training instances provided. The compatibility matching degree of each training pattern $x_p = [x_{p1}, x_{p2}, ..., x_{pn}]$, with respect to rule R_{hj} for class $C_h, h \in \{1, 2, ..., M\}$ is defined within the n-dimensional fuzzy subspace $A_j = A_{j1} \times A_{j2} \times \cdots \times A_{jn}$, such that:

$$\alpha(x_p, R_{hj}) = \nabla_{i=1}^{n} \nabla(\mu_{A_{ji}}(x_{pi}), Q_{ji}) \tag{4.8}$$

where ∇ represents a predefined *t-norm* operator.

In complete implementation of the proposed approach, where multiple rules $R_{h1}, R_{h2}, ..., R_{hj}$ may be used to describe the same class C_h, the final matching degree regarding to this sub-rule base R_h can be calculated as:

$$\beta_{C_h} = f(\alpha(x_p, R_{hj})) \tag{4.9}$$

where $f(.)$ is an aggregation operator (e.g., weighted vote, min or max), and $\alpha(x_p, R_{hj})$ is the matching degree for each rule describing C_h from sub-rule base R_h. As for the current simplified version, each sub-rule base for class C_h only contains one single rule without the need of aggregation among rules from R_h.

To determine the final class label of a testing pattern, the popular single winner rule policy is adopted, such that the pattern is identified with the class label from the sub-rule base that is of the following maximum matching degree:

$$C_{x_p} = \underset{C_h, h=1,2,...M}{\arg\max} \ \beta_{C_h} \tag{4.10}$$

If two or more classes take the same maximum value or the total compatibility degree is zero at a certain variable x_p, no pattern can be uniquely classified. To force a classification (if desired), such a pattern may be assigned with a default class label that is associated with most training instances. Such an inference strategy is generally depicted in Figure 4.2 [33].

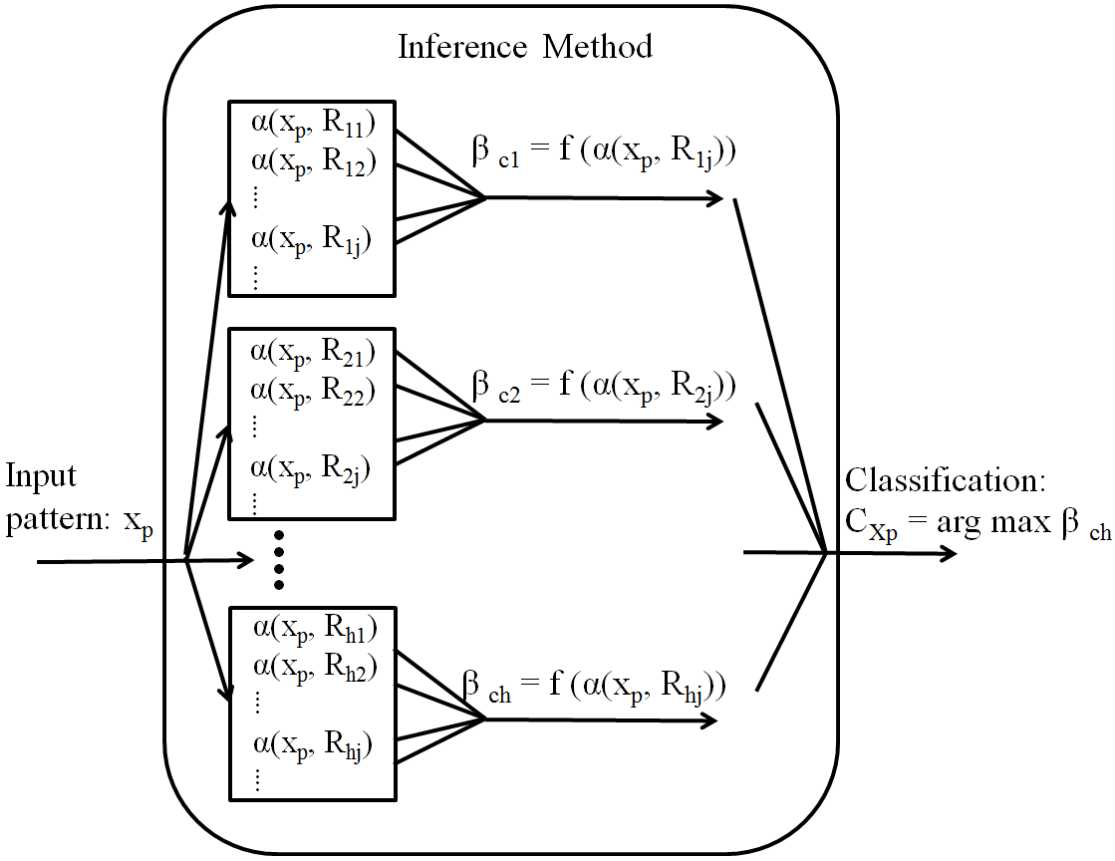

Figure 4.2: Single winner rule

4.2.3 Updating Quantified Fuzzy Rules

The quality of each PSO particle is then gauged by the overall quality of its encoded fuzzy rule base as reflected by the fitness function. In particular, as the entire rule base is designed to consist of sub-rule bases classifying instances from different classes, the overall quality of the whole rule base can thus be measured by decomposing it into qualities of individual sub-rule bases.

In order to measure the quality of a fuzzy sub-rule base, *F-score* is adopted, which combines a measure of both the sensitivity of a sub-rule base (its accuracy among instances of the same class, namely, recall r) and the specificity of the sub-rule base (its accuracy among instances of different classes, namely, precision p). Formally, *F-score* F is interpreted as a weighted average of the precision and the recall, such that:

$$F = 2 \cdot \frac{p \cdot r}{p + r} \tag{4.11}$$

F may achieve its best value of 1 and worse score 0. From this, the overall quality of a fuzzy rule base is deemed to be the sum of *F-score* values of individual fuzzy sub-rule bases each describing a class label C_h, $h \in \{1, 2, ..., M\}$, weighted by the fraction of instances m_h with class label C_h among all training instances m. The fitness value for a PSO particle is thus calculated as follows:

$$\text{fitness} = \sum_{h=1}^{M} \frac{m_h}{m} F_h \tag{4.12}$$

where F_h is the *F-score* value for the fuzzy sub-rule base describing class C_h.

Particles are then iteratively modified towards the best solutions with regard to a given quality measure over the set of fuzzy rules. For each generation, the so-called particle velocity is calculated by the following assignment:

$$v_x = wv_x + c_1 r_1 (x_{gBest} - x) + c_2 r_2 (x_{pBest} - x) \tag{4.13}$$

where w is the inertia weight affecting the trade-off between convergence and exploration-exploitation in the PSO process; c_1 and c_2 are two positive constants, termed social and cognitive scaling parameters in the literature, respectively; r_1 and r_2 are two random numbers within the range [0, 1]; x is the position for one particle dimension; x_{gBest} is the global best position of all particles, namely the rule weights currently capable of achieving the highest classification accuracy overall; and x_{pBest} is the best individual position where the particular particle p achieves the current best classification accuracy. The position is updated by the assignment: $x = x + \epsilon v_x$, where ϵ is an additional real-valued parameter used to control the evolving speed.

In summary, the algorithm using PSO to evolve and obtain an optimal set of fuzzy rules with quantifiers is presented in Algorithm 4.2.1, supported by Algorithm 4.2.2.

4.3 Experimentation and Validation

4.3.1 Experimental Setup

The PSO parameters are specified in Table 4.1. Note that as the main aim of this study is to examine the efficacy of applying PSO for the induction of a quantified

1 MAX_IT : number of maximum iterations;
2 $GOAL$: desired fitness value.
1: Initialisation
2: **repeat**
3: **for** each particle $l \in S$ **do**
4: **if** $f(x^l) < f(pBest_l)$ **then**
5: $pBest_l = x^l$
6: **end if**
7: **if** $f(pBest_l) < f(gBest)$ **then**
8: $gBest = pBest_l$
9: **end if**
10: **end for**
11: **for** each particle $l \in S$ **do**
12: **for** each dimension $d \in D$ **do**
13: **for** each sub-dimension $i \in n$ **do**
14: $v^l_{d,i} = wv^l_{d,i} + c_1 r_1(x_{gBest} - x^l_{d,i}) + c_2 r_2(x_{pBest} - x^l_{d,i})$
15: $x^l_{d,i} = x^l_{d,i} + \epsilon v^l_{d,i}$
16: **end for**
17: **end for**
18: **end for**
19: $it{+}{+}$
20: **until** $it > MAX\_IT$ or $GOAL$ is achieved
Algorithm 4.2.1: Induction of Quantified Fuzzy Rules with PSO

fuzzy rule base instead of that of PSO itself, only the basic version of PSO is used in the experiments. The parameter specification for PSO is not carefully adjusted, therefore, simulation results could be further improved where more sophisticated versions of PSO are used with carefully modified parameters.

In this work, for simplicity, each dimension of input space is divided into 5 fuzzy regions with the fuzzy membership values calculated by corresponding triangular/trapezoid functions as shown in Figure 4.3. As can be seen, the parameters that define these membership functions include the mean μ, standard deviation σ of the corresponding input dimension and a threshold θ, such that $a = \mu - 2\sigma; b = \mu - \sigma; c = \mu - \theta\sigma; d = \mu + \theta\sigma; e = \mu + \sigma; f = \mu + 2\sigma$. In particular, the threshold is empirically set to 0.7 consistently for all fuzzy sets, regardless of the problem domain. Simulation results may also be further improved with more carefully adjusted parameters for pre-defined fuzzy sets, with regard to different data sets. Similar methods of initialising fuzzy sets for each input space can be found in [85].

S : number of particles;
D : number of dimensions equal to number of rules;
n : number of antecedent attributes;
K : number of predefined fuzzy sets;
F_d : *F-measure* score of dth particle dimension;
$f()$: fitness function used to evaluate particles.

for each particle $l \in S$ **do**
  for each dimension $d \in D$ **do**
    for each sub-dimension $i \in n$ **do**
      $x^l_{d,i} = Rnd(0, K+1)$
      $v^l_{d,i} = Rnd(0, 1)$
    end for
    $F_d = 2 \cdot \frac{precision(d) \cdot recall(d)}{precision(d) + recall(d)}$
  end for
  $pBest_l = f(l)$
  if $f(pBest_l) < f(gBest)$ **then**
    $gBest = pBest_l$
  end if
end for

Algorithm 4.2.2: Initialisation of PSO particles

Stratified tenfold cross-validation (10-CV) is employed for validation. In 10-CV, a given data set is partitioned into 10 subsets. One single subset is maintained as the validation data for testing, and the remaining subsets are used for training. The process is then repeated 30 times by initialising different, randomly assigned seeds to produce the final average outcomes. In Table 4.2, PSO-QFR and PSO-FR stand for PSO evolved fuzzy rules with quantifiers and PSO evloved fuzzy rules without quantifiers, both of which are based on the class-dependent simultaneous rule learning strategy. As an initial implementation to test the proposed approach, only one fuzzy rule is generated for both PSO-QFR and PSO-FR. Pairwise t-tests are run to measure results in terms of the significance of differences between different learning classifiers with $p < 0.05$. Those results that are significantly better, worse or of no difference are marked with "(v)", "$(*)$", or "$(-)$", respectively, with the achieved accuracy of PSO-QFR as the reference in each experiment.

Table 4.1: Parameter values of PSO

w	c_1	c_2	ϵ	$Max\_Generation$	$Particle\_Numbers$
0.85	2.0	2.0	1.0	500	30

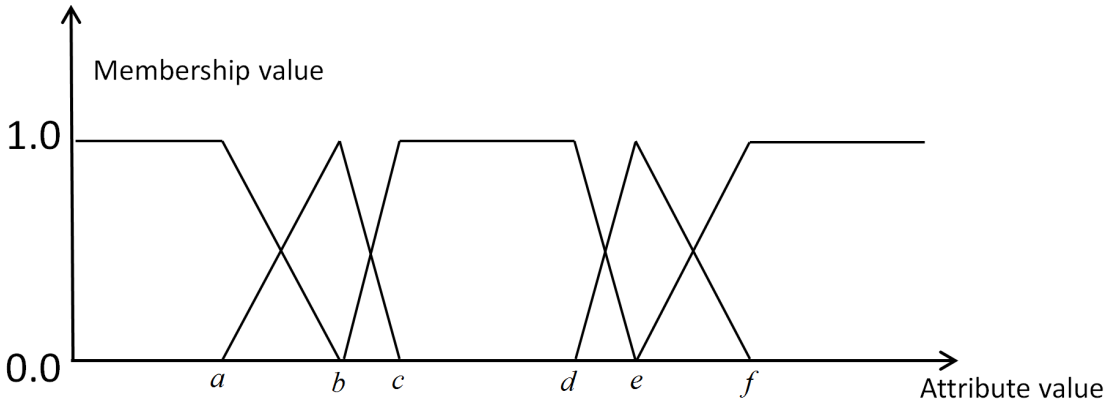

Figure 4.3: Partitioning of each pattern space dimensions

4.3.2 Results and Discussion

Experiments are performed on 6 real-valued benchmark data sets, the characteristics of which can be found in Appendix B. As shown in Table 4.2, PSO-QFR achieves better results than PSO-FR in terms of average classification accuracy, for 4 out of 6 data sets (including breast-cancer, glass, haberman, iris), together with one tie (new-thyroid) and one loss (blood). Generally speaking, fuzzy rules with quantifiers outperforms those without quantifiers in the experiment. This is not surprising since fuzzy rules with quantifiers associated with linguistic terms provide a richer information with regard to the relative importance among the antecedent attributes, thereby affecting the classification performance of the learned fuzzy rule base.

Table 4.2: Comparison using 30 × 10 cross-validation with respect to classification accuracy (%), where v, $-$ or $*$ indicate statistically better, same or worse results, respectively, and bold figures signify overall best results for each data set.

	PSO-QFR	PSO-FR	QSBA	QuickRules
blood	74.13 ± 0.49	**76.21** ± 0.17 (v)	66.72 ± 1.24 (*)	75.87 ± 1.12 (v)
breast-cancer	92.33 ± 0.98	83.69 ± 2.01 (*)	95.65 ± 0.15 (v)	**96.25** ± 0.28 (v)
glass	43.18 ± 2.60	39.12 ± 2.81 (*)	35.06 ± 1.55 (*)	**44.89** ± 1.91 (v)
haberman	**74.80** ± 0.49	73.26 ± 1.05 (*)	74.31 ± 1.56 (*)	71.54 ± 1.58 (*)
iris	**91.86** ± 2.04	73.27 ± 4.20 (*)	91.67 ± 0.34 (-)	89.60 ± 1.63 (*)
Thyroid	87.76 ± 1.73	87.28 ± 2.16 (-)	**93.15** ± 0.52 (v)	77.21 ± 8.48 (*)

From Figure 4.4, further observations can be obtained. For better viewing, the accuracy of each classifier is displayed for every 5 iterations within a total of 200

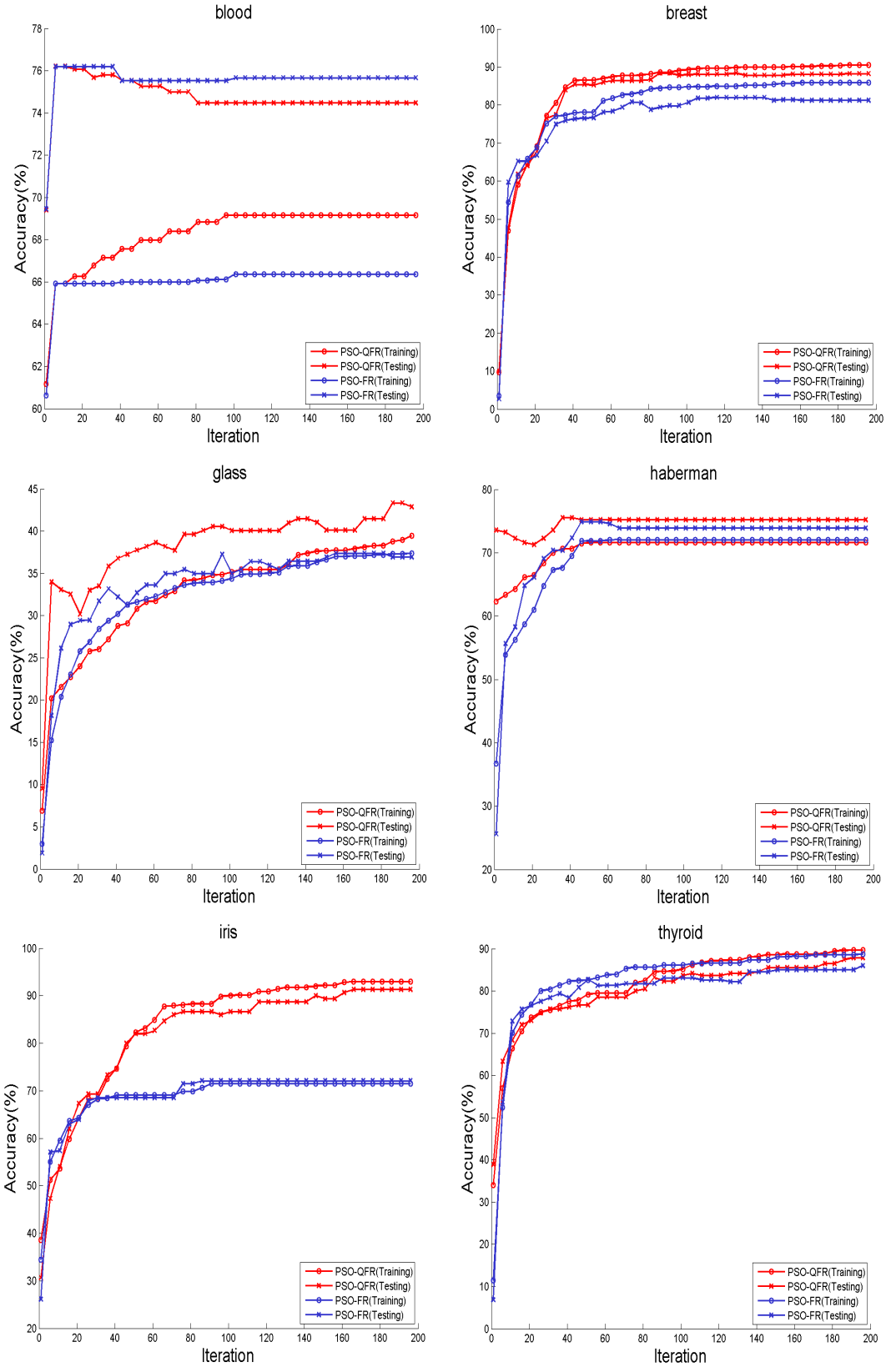

Figure 4.4: Relation between PSO iteration number and classification performance

iterations, and each point is the average of the results from a single tenfold cross validation run. As can be seen, after an initial period of oscillations, generally the trend of the training performance for all FRBCSs tend to converge at a certain iteration, with PSO-QFR outperforming PSO-FR, which also conforms to the experiment results from Table 4.2. It is also interesting to note that the final point for each of the testing curves on the blood, glass, and haberman datasets is even better than that achieved during training for both classifiers. This is probably because the assumption for the current implementation that one fuzzy rule is sufficient to adequately describe a class, such that fuzzy rules generated from training set may not fit the training data very well, especially with 10-CV when most of the data is used in the training phrase.

Two learning classifier algorithms that generate models in the form of a rule set are chosen to perform classification tasks for comparison. They are: QSBA, a fuzzy subsethood-based rule model with quantifiers [127]; QuickRules [82], a recently proposed hybrid fuzzy-rough rule induction. Both classifiers are implemented within the WEKA machine learning framework [162] with default parameter setting.

With only one rule per class in the implementation, the proposed method (PSO-QFR) has 3 wins, 1 tie and 2 losses, compared to QSBA, and wins and loses 3 times each compared to QuickRules. The winning results in terms of achieving the highest classification accuracy per learning classifiers are also highlighted in boldface in Table 4.2, showing that the proposed approach achieves 2 best results out of 6 data sets among the 4 classifiers. These results jointly demonstrate that the present work is at least competitive to popular rule-based classifiers in the literature regarding classification accuracy, and that the rule bases generated by the proposed approach boost classification performance as compared to those without fuzzy quantifiers.

4.4 Summary

This chapter has proposed a PSO-based approach that can learn a set of rules with continuous fuzzy quantifiers, such that all fuzzy rules can be combined and evaluated simultaneously. The approach works for situations where the information dealt with is not equally important, better capturing the relative importance among antecedent attributes by fuzzy continuous quantifiers. As an initial implementation of the proposed method, only one rule is generated per class. Experimental results show

that the rule bases generated by this method help boost the classification performance when compared to those generated without the use of fuzzy quantifiers. In addition, the performance of the proposed approach is at least competitive to popular rule-based learning classifiers.

Chapter 5

Induction of Accurate and Interpretable Fuzzy Rules with Preliminary Crisp Representation

One of the most important advantages of fuzzy systems lies in their inherent interpretability as they support the explicit formulation of, and inference with, domain knowledge, gaining insights into the complex systems and facilitating the explanation of their operations. In order to maintain the transparency in both the learned models themselves and the inferences performed by running the learned models, this chapter presents an approach that promotes an alternative approach, where a fuzzy model is initialised by utilising preliminary existing crisp rules that have been generated by a certain crisp rule-based learning mechanism.

This is motivated by the observation that such a data-driven rule generation method is able to omit the empty parts of the input space, which usually leads to curse of dimensionality as number of input feature increases, when fixed and predefined quantity space is required to maintain the exactly prescribed meaning of given labels and interpretability of the overall rule models. Being fundamentally data-driven, each of the generated crisp rules forms a certain partition of the entire problem space, and points to those parts in which desirable fuzzy rules may potentially exist, instead of considering all of the possible combinations of input and class variables.

Each crisp rule is then locally mapped onto a compact set of interpretable fuzzy rules involving only predefined meaningful fuzzy labels. This is followed by a global

genetic rule generalisation and selection procedure to produce a fuzzy model that is of high performance and interpretability (in both model semantics and model complexity). Note that the proposed approach is different from what is often done when utilising crisp rule-based classifiers to initialise potential fuzzy classifiers, which works by simply selecting the relevant input variables through the use of feature selection techniques [9, 121], or by directly fitting and fine tuning the generated crisp intervals into certain parameterised MFs [66, 120] (which would of course result in semantic loss).

The reminder of this chapter is organised as follows. Section 5.1 maps individual crisp rules into preliminary fuzzy rules for subsequent operations. Section 5.2 selects a subset of preliminarily transformed fuzzy rules that collectively generalise the corresponding original crisp rule. Section 5.3 performs global genetic rule and condition selections of all locally selected fuzzy rules from previous step. Section 5.4 conducts complexity analysis for this approach.

5.1 Mapping Crisp Rules to Fuzzy Rules

5.1.1 Heuristic Mapping

To generate an accurate and compact set of interpretable fuzzy rules effectively and efficiently, it is useful to have an initial focus on where the potentially meaningful rules may reside without going through an exhaustive search. An easily conceived way to implement this is to make use of an initial set of if-then crisp rules available (e.g., generated by a certain learning mechanism or provided by domain experts), even though such rules might not be very accurate. Without losing generality, suppose that a crisp rule $C_j, j = 1, 2, ..., N$ (with N denoting the number of all crisp rules available) is given as follows:

$$\text{If } x_1 \text{ is } I_{j1} \text{ and } ... \text{ and } x_n \text{ is } I_{jn}, \text{ Then class is } y^{C_j} \tag{5.1}$$

where $x_1, x_2, ..., x_n$ represent the underlying domain variables, jointly defining an n-dimensional input pattern space; I_{ji}, $i \in \{1, 2, ..., n\}$, is the crisp interval of the antecedent variable x_i; and y^{C_j} is a class label, acting as the rule consequent (which may be encoded as an integer for simplicity in implementation).

In general, a fuzzy if-then rule F_j can be represented as follows:

$$\text{If } x_1 \text{ is } D_{j1} \text{ and ... and } x_n \text{ is } D_{jn}, \text{Then class is } y^{F_j} \tag{5.2}$$

where $j = 1, 2, .., N$, with N denoting the number of all such fuzzy rules within the system; $x_i, i = 1, \ldots, n$ are the underlying domain variables, jointly defining the n-dimensional pattern space and respectively taking values from X_i; $D_{ji} \in X_i$ denotes a fuzzy set that the variable x_i may take; and $y^{F_j} \in Y$ is the consequent of the fuzzy rule F_j that is to be assigned to one of the M possible output classes.

Note that fuzzy rules adopted in this chapter do not involve the use of rule weights. Their involvement could further improve classifier performance as can be seen from previous two chapters, but may pay the price of affecting some semantic transparency, as rule weights change the normality of antecedent fuzzy sets [5]. Importantly, unless otherwise stated, in this work, each D_{ji} in the above description is a semantic fuzzy set for the variable x_i, which is predefined and fixed throughout both the modelling and inference processes.

In order to approximate the modelling problem with a set of fuzzy rules as of Eqn. (5.2), where variables are described with predefined fuzzy sets instead of crisp intervals, a procedure is required to convert crisp intervals into the corresponding fuzzy terms. The idea to implement such a mapping is to use a similarity measure between a crisp interval and each of the predefined fuzzy sets describing the same variable, such that only those fuzzy sets are considered valid whose similarity values are above a user-defined threshold η.

A heuristic is employed herein to obtain the set of potentially useful interpretable rules by mimicking the method of [107]. It builds up a layered graph, where a node in a certain layer contains a number of predefined fuzzy sets in association with each existing crisp interval per variable. A node is only generated if any of its predefined fuzzy sets has a similarity measure with the original crisp interval above a given threshold or confidence level. This process iterates until all the corresponding crisp sets that are associated with all the nodes within each layer have been successfully replaced by predefined fuzzy sets. A path from one layer to another can be built by connecting one and only one node from each layer. As such, each resultant path can be interpreted as a possible interpretable fuzzy rule which coarsely approximates the given crisp rule under mapping.

Note that crisp intervals in a crisp rule are themselves crisp sets, each of which can be seen as a special case of fuzzy sets. Thus, the similarity between a crisp set and a fuzzy set can be generalised as the similarity between two fuzzy sets. There are many such similarity metrics available in the literature. The following set-theoretic based similarity measure is adopted in this work (owing to its popularity though others may be used as an alternative):

$$S(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|} \tag{5.3}$$

where A and B denote two fuzzy sets; $|.|$ represents the cardinality of a fuzzy set; and $\bigcap$ and $\bigcup$ denote set intersection and union operator, respectively.

From the above, the similarity between a predefined fuzzy set D_{ji} and a crisp set I_{ji} regarding the i-th variable within a given rule C_j can be rewritten as:

$$S(D_{ji}, I_{ji}) = \frac{\sum_{\bar{x}^p \in E_j^i} [\mu_{D_{ji}}(x_i^p) \wedge \mu_{I_{ji}}(x_i^p)]}{\sum_{\bar{x}^p \in E_j^i} [\mu_{D_{ji}}(x_i^p) \vee \mu_{I_{ji}}(x_i^p)]} \tag{5.4}$$

where $\wedge$ and $\vee$ represent the minimum and maximum operator, respectively, and

$$E_j^i = \{\bar{x}^p \mid \prod_{k \neq i} \mu_{I_k}(x_k^p) > 0, \bar{x}^p \in E_{\text{trn}}\} \tag{5.5}$$

where $\bar{x}^p$ stands for an instance from the training data set E_{trn}; and the check of $\mu_{I_k}(x_k^p) > 0$ is to ensure that the training instance intersects with all antecedent variables, except the one under consideration for mapping.

The computation effort required for this similarity measure is significantly lighter than what it may appear at the first sight. This is because in general, the set of training instances used for calculating the similarity is not the entire training set, but the subset of training data specified by Eqn. (5.5). However, it does not necessarily ensure a good coverage of the original crisp rule unless the threshold value is set very low. Yet, a low threshold implies many matching nodes to be retained and hence, many potential fuzzy rules to be created. A large number of rules not only increases computational complexity but also deteriorates the interpretability of the learned model. A way to reduce the impact of this sensitivity in parameter setting is to introduce another user-defined parameter T such that a very low threshold value may be set, but only those T most similar fuzzy sets may be retained per variable.

5.1.2 Illustrative Example

To illustrate the basic idea of the above heuristic process, consider a crisp rule C under mapping as follows:

$$\text{If } x_1 \text{ is } I_1 \text{ and } x_2 \text{ is } I_2, \text{ Then class is } y^C \tag{5.6}$$

where I_1 and I_2 are two crisp sets describing the two input variables x_1 and x_2, respectively. Suppose that a collection of predefined fuzzy sets $\{D_{ji} | j = 1, 2, .., k_i\}$ per variable $(x_i, i = 1, 2)$ is provided. For simplicity, let $k_i = 3, i = 1, 2$. In particular, the three semantic fuzzy sets are defined for each variable such that x_1 may take a value on either of $D_{11} =$ low, $D_{21} =$ medium, $D_{31} =$ high, and x_2 on either of $D_{12} =$ small, $D_{22} =$ medium, $D_{32} =$ large.

Following the heuristic approach, the first layer of the hierarchical graph is set to work on the crisp set of the first antecedent variable first, i.e., I_1 in this case (assuming the strategy of first come first served). Then, a node is created for each of the predefined corresponding fuzzy sets $D_{j1}, j = 1, 2, 3$ if it has a similarity value greater than a given threshold η (which is here set to 0 by default) to I_1. Suppose that $S(I_1, D_{11}) = 0$, $S(I_1, D_{21}) = 0.75$, and $S(I_1, D_{31}) = 0.3$. With the default threshold, the nodes representing the two valid fuzzy sets of D_{21} and D_{31} are retained in the graph. The similar process is repeated for the next antecedent variable. From which, all retained nodes in a preceding layer are connected to those in the immediate subsequent layer. The result of this mapping process for the example is shown in Figure 5.1.

Once such a graph is generated, each path becomes an emerging fuzzy rule, with the antecedent variables described by corresponding fuzzy sets, while the rule consequent remains to be the same as that of the original crisp rule. This leads to a set of possible fuzzy rules involving the use of only predefined fuzzy sets. For this example, the resultant rules are:

Rule F_1**:** If x_1 is medium and x_2 is small, Then y^C

Rule F_2**:** If x_1 is medium and x_2 is medium, Then y^C

Rule F_3**:** If x_1 is high and x_2 is small, Then y^C

Rule F_4**:** If x_1 is high and x_2 is medium, Then y^C

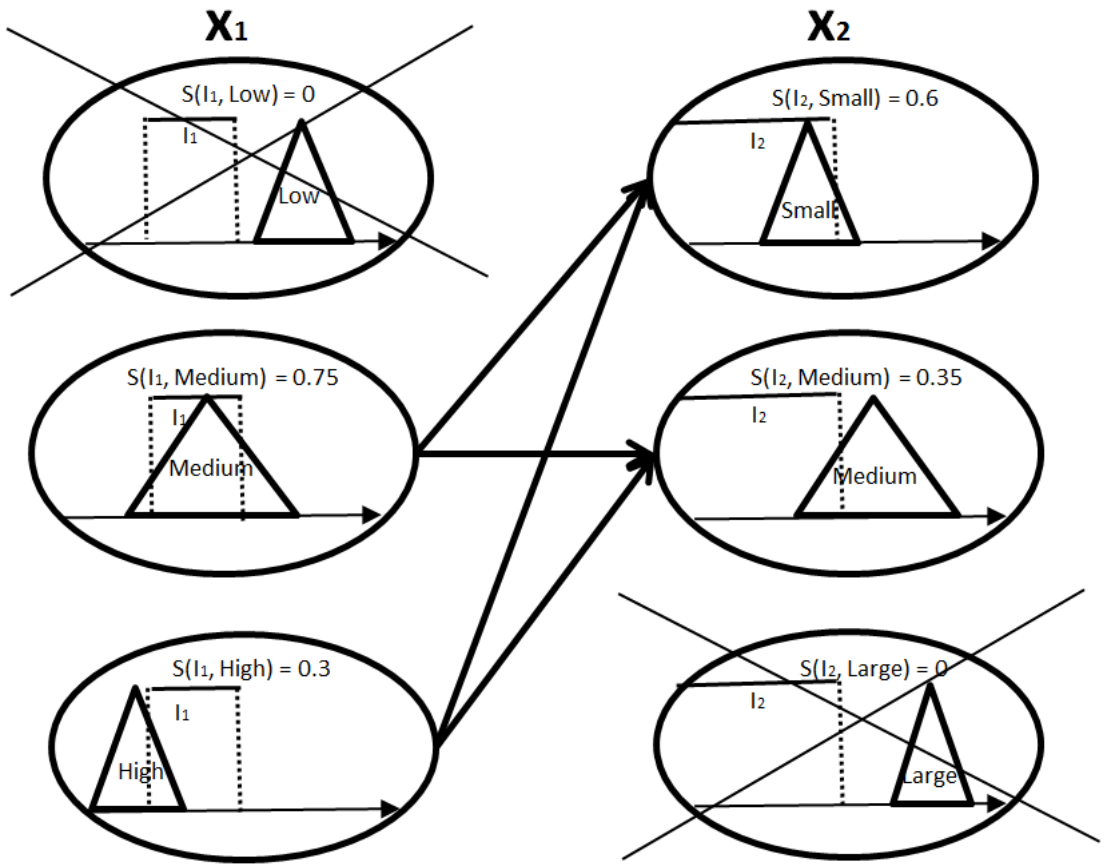

Figure 5.1: Example on heuristic mapping

5.2 Local Rule Selection

5.2.1 Functional Generalisation

With the use of similarity measure, the heuristic method generates a set of interpretable fuzzy rules with respect to each existing crisp rule. However, the employment of all such preliminarily mapped fuzzy rules does not necessarily optimally mimic the capability of the original crisp rule. Unlike crisp rule-based environment, where an instance is only covered by one crisp rule, each instance may now match with multiple fuzzy rules to various degrees. Unfortunately, certain mapped fuzzy rules may be conflicting with each other, and certain rules may be very similar with each other (resulting in duplications). These issues must be addressed, not just to increase computational efficiency but also to decrease potential model inconsistency and complexity.

A local rule selection procedure is proposed here to tackle these issues, by introducing the constraint of *functional generalisation*. This constraint imposes that in searching for a subset of initially mapped fuzzy rules to replace the full set of the (possibly inconsistent and/or redundant) preliminary rules, the subset must collectively generalise the capability of the original crisp rule from which they are mapped while avoiding inconsistency and redundancy for the given data.

Suppose that there are N crisp rules $C_j, j = 1, 2, ..., N$, and that K_j preliminary fuzzy rules $F_{ji}, i = 1, 2, ..., K_j$ are mapped from C_j using the heuristic method. For each input pattern $\bar{x}^p \in E_{trn}$, the rule firing degree $\mu_{F_{ji}}(\bar{x}^p)$ with respect to the entire set of fuzzy rules F_{ji} is intuitively defined as the largest matching degree amongst all:

$$\mu_{F_{ji}}(\bar{x}^p) = \max\{\mu_{F_{j1}}(\bar{x}^p), ..., \mu_{F_{ji}}(\bar{x}^p), ..., \mu_{F_{jK_j}}(\bar{x}^p)\} \tag{5.7}$$

Let E_j denote the set of instances selected to measure the quality of a selected subset of fuzzy rules $F_{ji'}, i' = 1, 2, ..., S_j, S_j \leq K_j$, which satisfies the following:

$$E_j = \{\bar{x}^p | \mu_{F_{ji}}(\bar{x}^p) > 0, \bar{x}^p \in E_{trn}, i = 1, \ldots, K_j\} \tag{5.8}$$

To ensure the desired functional generalisation, there are five cases to consider regarding the different instances of a given E_j:

(i) Instances that are covered and correctly classified by the original crisp rule C_j:

$$E_{j1} = \{\bar{x}^p | y^p = y^{C_j}, \mu_{C_j}(\bar{x}^p) = 1, \bar{x}^p \in E_j\} \tag{5.9}$$

where y^p is the underlying label of the instance $\bar{x}^p$, and y^{C_j} is the rule consequent of C_j. It is desirable to maximise the firing degrees over these instances when using the selected fuzzy rules, by imposing the requirement that such instances be still correctly classified, while avoiding influence from other mapped fuzzy rules, especially those whose rule consequents are inconsistent with the selected rules.

(ii) Instances that are covered, but wrongly classified by C_j:

$$E_{j2} = \{\bar{x}^p | y^p \neq y^{C_j}, \mu_{C_j}(\bar{x}^p) = 1, \bar{x}^p \in E_j\} \tag{5.10}$$

It is desirable to minimise the firing degrees over these instances when using the selected fuzzy rules, as much as possible, while improving the opportunity for them to be classified by other mapped fuzzy rules with consistent class labels.

(iii) Instances that are not covered by the original crisp rule C_j, but by an alternative rule $C_{j'}$ with correct classification which happens to be of the same consequent as C_j, and that are now to a certain extent matched with the fuzzy rules F_{ji} that are mapped from C_j with consistent classification:

$$E_{j3} = \{\bar{x}^p | y^p = y^{C_j} = y^{C_{j'}}, \mu_{C_{j'}}(\bar{x}^p) = 1, j' \neq j, \bar{x}^p \in E_j\} \tag{5.11}$$

It is natural not to do anything in this case since the fuzzy rules mapped from C_j will provide the same correct class label as that inferred by certain other fuzzy rules mapped from the other original crisp rule $C_{j'}$.

(iv) Instances that are otherwise regarded as the same as those in Case (iii), except that they are incorrectly classified by $C_{j'}$:

$$E_{j4} = \{\bar{x}^p | y^p = y^{C_j} \neq y^{C_{j'}}, \mu_{C_{j'}}(\bar{x}^p) = 1, j' \neq j, \bar{x}^p \in E_j\} \tag{5.12}$$

It is desirable to maximise the firing degrees over these instances when using the fuzzy rules selected from those mapped from C_j, as much as possible, while providing additional support for those instances of Case (ii).

(v) Instances whose class labels are inconsistent with those of the original crisp rule C_j, but either they are correctly classified by an alternative rule $C_{j'}$ with a consistent rule consequent:

$$E_{j5a} = \{\bar{x}^p | y^{C_j} \neq y^p = y^{C_{j'}}, \mu_{C_{j'}}(\bar{x}^p) = 1, j' \neq j, \bar{x}^p \in E_j\} \tag{5.13}$$

or they are incorrectly classified by an alternative rule $C_{j'}$:

$$E_{j5b} = \{\bar{x}^p | y^p \neq y^{C_j}, y^p \neq y^{C_{j'}}, \mu_{C_{j'}}(\bar{x}^p) = 1, j' \neq j, \bar{x}^p \in E_j\} \tag{5.14}$$

It is desirable to minimise the firing degrees over these instances when using the selected fuzzy rules, as much as possible, given that the consequents of such fuzzy rules are not to be consistent with the true classes of these instances, while improving the opportunity for them to be matched with rules that are mapped from other crisp rules with correct classification. For simplicity in description later, introduce the notion of E_{j5} such that $E_{j5} = E_{j5a} \cup E_{j5b}$.

Figure 5.2 summarises the five types of instance and their associated appropriate actions.

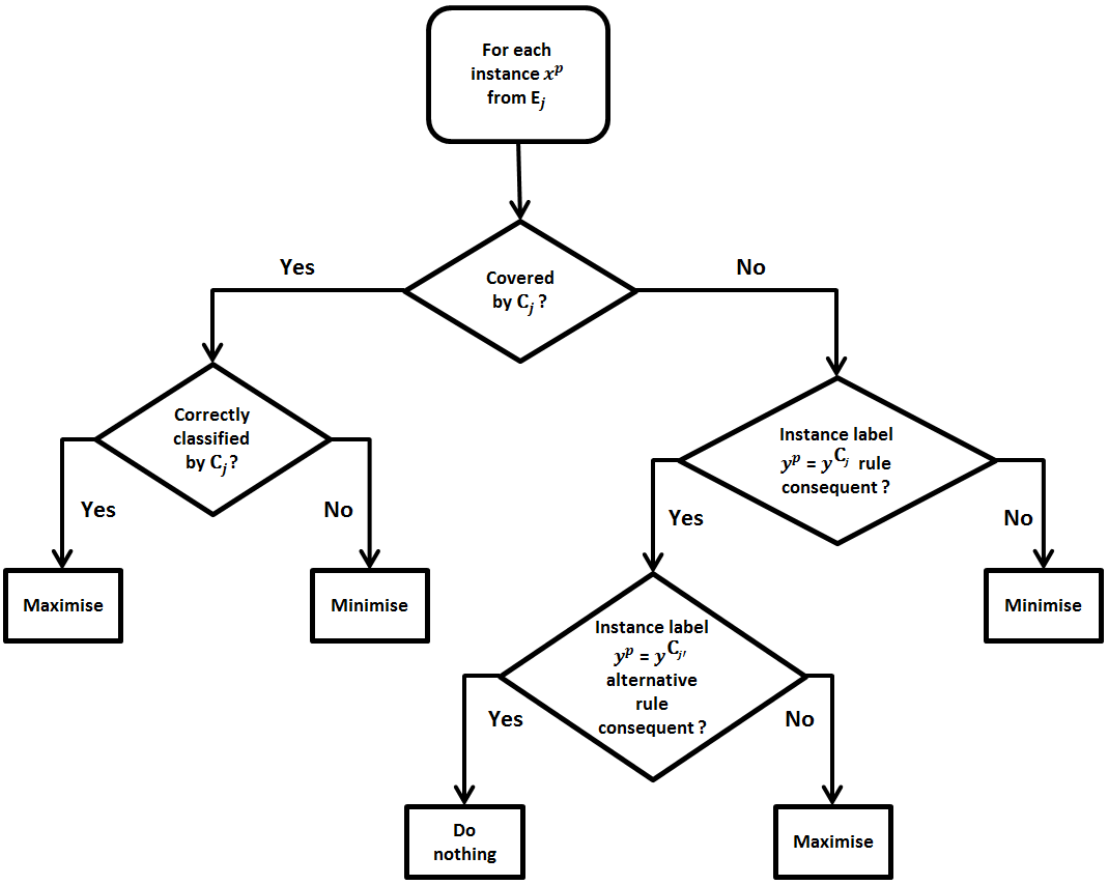

Figure 5.2: Functional generalisation regarding instances from different cases

5.2.2 Search for Subset of Quality Mapped Rules

Given the above discussion, the quality $Q(F_{ji'})$ of a subset of the fuzzy rules $F_{ji'}, i' = 1, 2, ..., S_j, S_j \leq K_j$, selected from the K_j rules $F_{ji}, i = 1, 2, ..., K_j$, mapped from the preliminary crisp rule C_j in relation to the data set E_j, can be evaluated as follows:

$$Q(F_{ji'}) = \sum_i Q_{E_{ji}}(F_{ji'}) \tag{5.15}$$

where $Q_{E_{ji}}(F_{ji'}) \in [0, 1], i = 1, 2, 4, 5$, denote the quality measures of the same sets of fuzzy rules over the data instances that belong to Case i. Note that Case iii is not included due to its nature as indicated previously.

The component quality measures $Q_{E_{ji}}$ can be computed by adopting that often applied in conventional classification techniques, using the following biased mean

squared error:

$$Q_{E_{ji}}(F_{ji'}) = 1 - \frac{1}{|E_{ji}|} \sum_{\bar{x}^p \in E_{ji}} (\mu_{F_{ji'}}(\bar{x}^p) - \theta)^2, \tag{5.16}$$

where $|E_{ji}|$ is the cardinality of instances from Case i; $\mu_{F_{ji'}}(\bar{x}^p)$ denotes the largest matching degree of the instance $\bar{x}^p$ with the selected subset of fuzzy rules $F_{ji'}$; $\theta \in \{0.0, 1.0\}$ represents the desired value (depending on whether it is for maximisation or minimisation) regarding the instance $\bar{x}^p$, that is, $\theta = 1.0$ if $\bar{x}^p \in E_{j1} \cup E_{j4}$, $\theta = 0.0$ if $\bar{x}^p \in E_{j2} \cup E_{j5}$.

Following the above approach the generalisation capability of the selected fuzzy rules that are mapped from a given crisp rule C_j is assessed with regard to an equal weight over the five types of training data instance. This may not be the ideal in general because not only the number of instances from different types can vary, the matching degrees of individual instances are not the same either, where higher matching degrees ought to be considered contributing more to the overall quality than the lower ones.

To better address this issue, a weighted approach is taken here. In particular, the weight w_{ji} that is associated with an individual quality measure is specified as the ratio between the sum of the matching degrees of the instances belonging to that given type E_{ji} and the total of the matching degrees of all instances in E_j such that

$$w_{ji} = \frac{\sum_{\bar{x}^p \in E_{ji}} \mu_{F_{ji}}(\bar{x}^p)}{\sum_{i=1,2,4,5} \sum_{\bar{x}^p \in E_{ji}} \mu_{F_{ji}}(\bar{x}^p)} \tag{5.17}$$

where $\mu_{F_{ji}}(\bar{x}^p)$ is the matching degree of the instance $\bar{x}^p$ regarding all K_j preliminary fuzzy rules as defined in Eqn. (5.7). In addition, to minimise redundant rules, the following relative size $S(F_{ji'})$ of the resultant fuzzy rules is also factored into the overall quality measure:

$$S(F_{ji'}) = 1 - \frac{|F_{ji'}|}{|F_{ji}|} \tag{5.18}$$

Thus, the quality $Q(F_{ji'})$ of a selected subset of the fuzzy rules $F_{ji'}$ mapped from a given crisp rule C_j will be assessed as follows:

$$Q(F_{ji'}) = \sum_{i=1,2,4,5} w_{ji} Q_{E_{ji}}(F_{ji'}) + w_s S(F_{ji'}) \tag{5.19}$$

where $w_s \in [0, 1]$ is a parameter that allows for the adjustment of the relative contribution of the size of the subset of selected fuzzy rules towards the quality of that subset (which can be set to 1 by default in implementation).

5.3 Tuning of Interpretable Fuzzy Rule Base

The above work ensures that a subset of fuzzy rules can be selected that collectively generalise a given crisp rule. However, globally, the combination of all such locally selected fuzzy rules does not necessarily result in an optimal and compact interpretable rule base, especially from the ruleset complexity viewpoint. Although each subset of rules may be optimised separately, the quality of any neighbouring subsets (which share antecedent variables) may be deteriorated if they are not optimised at the same time. The overall performance of the entire rule base is thus unpredictable when all crisp rules are mapped simultaneously. With the aim to obtain a compact ruleset with high performance, when given all of the selected fuzzy rules in response to all existing crisp rules, a method is therefore required to search for an optimal set of fuzzy rules globally.

For aforementioned purpose, genetic algorithms (GAs) are employed in this work to implement the required global search owing to their practical popularity and conceptual simplicity. GAs realise a population-based search meta-heuristic inspired by the process of natural selection. Of course, other stochastic population-based techniques may be adopted as alternative for implementation, if preferred.

Generally speaking, in applying GAs, a set of possible solutions are represented as chromosomes, with better emerging solutions more likely to be selected as offsprings according to their fitness, where new solutions are generated mainly based on crossover and mutation operators. In order to allow more flexibility for ruleset tuning, each encoded fuzzy rule is assumed to always include n antecedents, with a *don't care* label in place of void in the corresponding variable location within the rule. Obviously, an emerging rule will be eliminated if *don't care* appears as the value for all antecedent variables. In so doing, for a problem involves an n-dimensional pattern space, each variable $x_i, i \in \{1, 2, ..., n\}$ may take any fuzzy set from its domain $\{D_0, D_1, ..., D_{d_i}\}$ (whose cardinality is d_i), with D_0 representing the notion of *don't care* (that has a specifically fixed membership value of 1). In implementation within this work, the GA used adopts Pittsburgh style encapsulation, whereby the combination of all selected fuzzy rules returned by the local rule selection process are encoded within a single chromosome, where individuals of the first population are initialised with an exact copy of the selected fuzzy rules.

Recall that the ultimate goal of this tuning process is to obtain an accurate fuzzy rule base that is interpretable in terms of both semantics and complexity. As the semantic interpretability is already ensured by the consistent use of predefined fuzzy sets, the fitness function takes both the accuracy and complexity of a resultant fuzzy rule base into account, such that

$$Q = Q_p - w_i Q_i \tag{5.20}$$

where Q_p measures the performance of the resultant rule base, defined as the accuracy rate of correctly classified instances; Q_i measures the structural complexity of the rule base, defined as the size of the resulting rule base, penalising rule base with a large number of rules or rules of many compound conditions; and w_i is a weighting factor to balance the expected contributions of the two quality indicators. As such, this work follows a conceptually simple method that converts multiple objectives into a compound single objective.

5.4 Complexity Analysis

Given a set of crisp rules $\{C_j | j = 1, 2, ..., N\}$ (returned by a certain data-driven existing crisp rule learner), and a fixed linguistic term set with underlying semantics defined as fuzzy sets reflecting the domain expertise, the process of generating an interpretable fuzzy rule base can be summarised into the following three-stage process, as outlined in Figure 5.3.

1) Mapping crisp rules into interpretable fuzzy rules. For each crisp rule C_j:
    a) Generate the (sub-)data set E_j^i relevant to each antecedent variable x_i.
    b) Compute similarity between crisp interval I_{ji} and each of the predefined fuzzy set D_{ji} of x_i.
    c) Retain those fuzzy sets whose similarity values surpass user-defined threshold η, resulting in a set of emerging interpretable fuzzy rules $F_{ji}, i = 1, 2, ..., K_j$.

The cost incurred in this stage to generate the initial sets of fuzzy rules is $O(N \times N_{intl} \times d)$, where N denotes the number of given crisp rules, N_{intl} is the maximum number

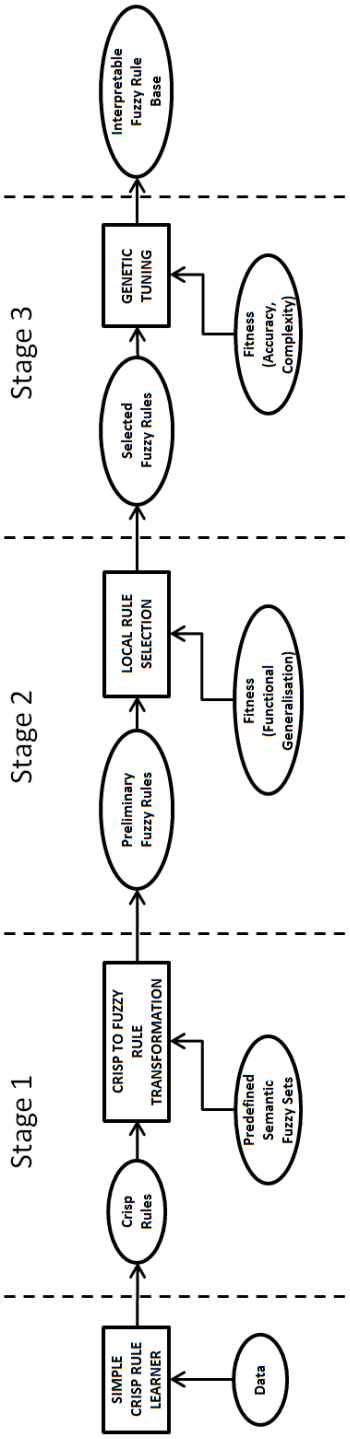

Figure 5.3: Generation of accurate and interpretable fuzzy model from a crisp rule learner: Three stages

of the existing crisp intervals for any crisp rule, and d is the maximum number of predefined fuzzy sets for any attribute. In practice, N_{intl} is set to a small number to allow for more general rules [66] whilst d is not large, which is typically at most 9 owing to psychological theory for the learned rules to be interpretable (although in the experimentation later, this may be set to 14 in an effort to demonstrate that the proposed method works even with larger than usual variable domains).

2) Selecting mapped fuzzy rules with functional generalisation. For each set of fuzzy rules $F_{ji}, i = 1, 2, ..., K_j$ mapped from C_j:

   a) Categorise instances from E_j into five types.
   b) Compute weights for each type.
   c) Obtain a locally optimal selected subset of fuzzy rules $F_{ji'}, i' = 1, 2, ..., S_j, S_j \leq K_j$ with functional generalisation (which is also implemented with a simple GA in this work).

The cardinality of possible fuzzy rules generated in response to each crisp rule is bounded by $N_{intl} \times T$, where T is the maximum number of similar fuzzy sets that are allowed per crisp interval. In practice, as with N_{intl}, T is set to a small number to avoid potentially generating too many redundant rules. For each crisp rule, the cost for rule evaluation over a subset of initially mapped fuzzy rules is bounded by $2^{N_{intl} \times T}$. The total computational effort at this stage is therefore, $O(N \times 2^{N_{intl} \times T})$, which can be practically resolved by GA given that N_{intl} and T are both a small number.

3) Computing a globally compact and accurate fuzzy rule base with GA.

   a) Encode all locally optimised fuzzy rules together in Pittsburgh style.
   b) Optimise the interpretable fuzzy rule base, with performance and complexity jointly encoded as the fitness function.

Suppose that the cardinality of the family of all selected fuzzy rules is N_r, then, the cost for the final generic tuning is $O(d^n \times N_r)$, where n is the number of antecedent attributes in the domain. In practice, as the outcome of Stage 2 has already provided a good solution and d is not large, the GA often converges very quickly at this stage (which is also supported by experimental results as to be shown in Section IV-G).

Finally, note that at the end of each stage, appropriate conventional rule-pruning mechanisms may be employed if desired, but this is beyond the scope.

5.5 Experimentation and Validation

Systematic experiments using benchmark data sets are reported here to demonstrate the efficacy of the proposed approach. Section 5.5.1 introduces the experimental setup. Section 5.5.2 shows the generation of interpretable fuzzy rules, which are initialised from crisp rules generated by two distinct learning mechanisms, and compares the generated rules with those directly fuzzified by the use of the popular FURIA algorithm [66]. Section 5.5.3 compares performance of the generated rule bases with alternative fuzzy rule-based learning classifiers that only use fixed and predefined fuzzy sets, with rule bases complexity analyses as shown in Section 5.5.4. For completeness, Section 5.5.5 compares the proposed work with non-fuzzy-rule-based learning approaches. Section 5.5.6 investigates the effect of local rule selection in relation to functional generalisation.

5.5.1 Experimental Setup

To demonstrate the proposed approach at work, experiments are performed on 16 real-valued benchmark data sets, the characteristics of which can be found in Appendix B. Stratified tenfold cross-validation (10-CV) is employed for result validation. In 10-CV, a given data set is partitioned into ten subsets. Of the ten, nine subsets are used to perform training, where the proposed approach is used to generate an interpretable fuzzy rule base, and the remaining single subset is retained as the testing data for assessing the learned classifier's performance. This cross-validation process is then repeated ten times in order to lessen the impact of random factors; these 10×10 sets of evaluations are then averaged to produce each final experimental outcome reported below (except for the particular investigation into the effect of local rule selection as reported in Section 5.5.6).

Table 5.1: Parameter specifications of GA

Stage 2	$w_s = 0.1, Pop = 100, P_c = 0.95, P_m = 0.005, maxItr = 100, itr\_no\_improve = 10$
Stage 3	$w_i = 0, Pop = 100, P_c = 0.95, P_m = 0.005, maxItr = 500, itr\_no\_improve = 30$

For fair and systematic comparison, fixed and uniformly divided fuzzy sets are used in the experiments. As the partition granularity for each variable is unknown in

Table 5.2: Parameter specifications of the learning classifiers used for experimentation

Approach	Parameter Specification
PTTD	$\epsilon = 0.0025, numCandidates = 5, maxDepth = 0$
GP-COACH	$Labels = 5, Eval = 20000, Pop = 200, \alpha = 0.7, P_c = 0.5, P_m = 0.2, P_{dp} = 0.15, P_i = 0.15, Tournament = 2, w_1 = 0.8, w_2 = w_3 = 0.05, w_4 = 0.1$
SLAVE2	$Pop = 20, Iter_{\text{change}} = 500, P_{bm} = 0.5, P_{bc} = 0.1, P_{rm} = 1.0, P_{rc} = 0.2, \lambda = 0.8$
MOGUL	$Labels = 5, \omega = 0.05, K = 0.1, \epsilon = 1.5, repeat\_rules = 1, rule\_type = 2, Iter_{\text{selection}} = 500, Pop_{\text{selection}} = 61, \tau = 1.5, \beta = 0.5, P_{cs} = 0.6, P_{ms} = 0.1, Iter_{\text{tuning}} = 1000, Pop_{\text{tuning}} = 61, a = 0.35, b = 5, P_{ct} = 0.6, P_{mt} = 0.1$
FH-GBML	$Rules = 30, Sets = 200, Gens = 1000, P_c = 0.9, P_{\text{dont-care}} = 0.5, P_{\text{michigan}} = 0.5$
SGERD	$Q_{\text{rules}} = 0$(calculate heuristically)$, RuleEval = 2$
QSBA	$Labels = 5, thres = 0.7, Tnorm =$ Algebraic
C4.5	$Pruned =$ yes$, confidence = 0.25, minNumObj = 2, numFolds = 3, reduced_error_pruned =$ yes
RIPPER	$Pruning = yes, Folds = 3, N_{\text{optimisations}} = 2$
NB	default
SMO	$c = 1.0, \epsilon = 1.0 \times 10^{-12}, tolerance = 0.001$
IBk	$kNN = 1, search\_algorithm =$ linear search$, window = 0$
FRNN	$kNN = 10, TNorm =$ KD$, Implicator =$ KD$, Similarity = 1$
NFC	$epoch = 100, \sigma = 5.0e^{-5}, \lambda = 5.0e^{-7}$
C45-IFRC	$maxDepth = 3, T = 3, \eta = 0, w_i = 0$
UR-IFRC	$maxDepth = 5, T = 3, \eta = 0, w_i = 0$

advance, in this work, without any bias and for simplicity, four types of homogeneous fuzzy partition with uniformly divided triangular MFs are employed, as shown in Figure 5.4. That is, each antecedent variable may take one fuzzy set from the domain $\{D_1^2, D_2^2, D_1^3, ..., D_5^5\}$ (in addition to the value that stands for *don't care*). Given such underlying value domains, 4 bits are required for encoding each variable in the binary encoded chromosomes, with 0000 and 1111 reserved for the *don't care* label, and the rest for the 14 distinct fuzzy sets. The total length of a chromosome required is $4nN_r$, where N_r is the cardinality of the family of all selected fuzzy rules after Stage 2 of the learning process. The fitness function is defined as given in Eqn. (5.20). Each implemented GA utilises the steady-state with elitism selection strategy.

As the main aim of this investigation is to examine the efficacy of the proposed approach for the acquisition of an interpretable rule base, instead of the performance

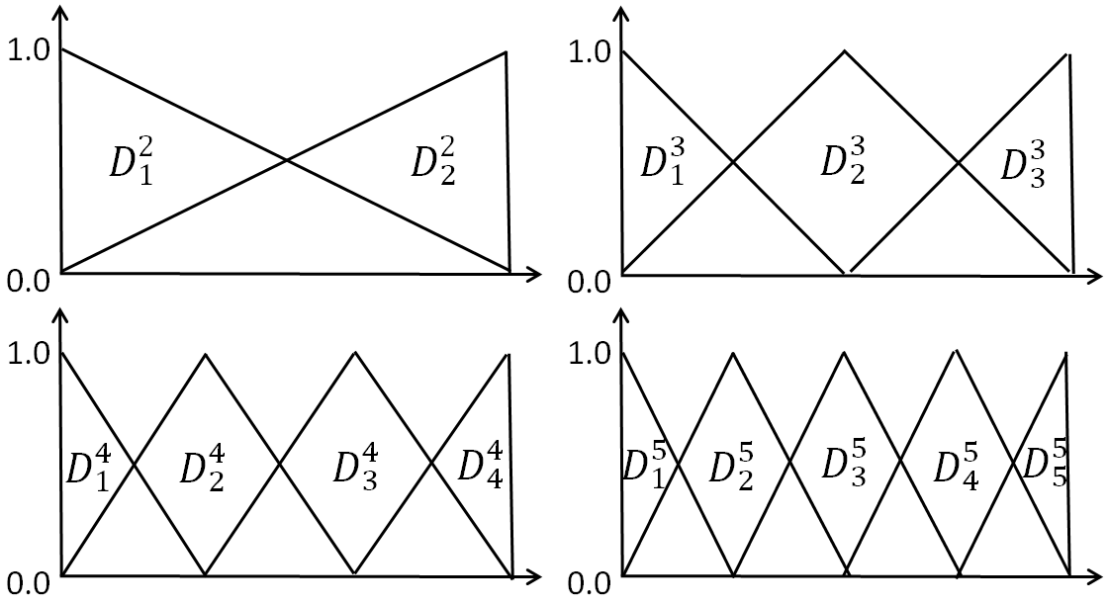

Figure 5.4: Partitioning of pattern space

of a GA itself, only the basic version of GA is used in the experiments. The parameter specification for GA is not purposefully adjusted and therefore, the experimental results could be further improved where a more sophisticated version of GAs is employed with carefully modified parameters. In fact, GAs with the same parameter specifications as detailed in Table 5.1 are applied to generate fuzzy rules that are initialised by two distinct crisp rule-based learning mechanisms. Parameter specification involved in the proposed approach and alternative learning classifiers is summarised in Table 5.2. Note that the implementation of the compared approaches can be found in WEKA [162] or KEEL [4].

5.5.2 Generating Fuzzy Rules with C4.5 and Unordered RIPPER

Two highly popular crisp rule-based classifier learners are each employed here to act as the initial crisp rule generator to enrich the comparison. These are C4.5, a classical decision tree learning algorithm, and an unordered version of RIPPER (UR) [66]. Comparison is also made with FURIA [66], commonly served as the benchmark that greedily transforms crisp rules into fuzzy rules by fitting initially generated crisp intervals into parameterised trapezoid MFs, where C4.5 and UR are also separately used as the initial rule generator. For conciseness, the resulting learned rule sets are

shorthanded as C45-IFRC and UR-IFRC, for C4.5- and UR-initialised interpretable fuzzy rule-based classifiers, and as C45-FURIA and UR-FURIA for C4.5- and UR-initialised FURIA, respectively. Note that UR-FURIA is the exact FURIA algorithm itself that converts UR rules directly into fuzzy rules, and is renamed purely for meeting the eyes.

Table 5.3 presents the results with C4.5 used as the initial rule generator, where the top performer in terms of classification accuracy is highlighted in boldface for each data set, and pair-wise t-test ($p = 0.05$) results are identified to reflect their statistical significance. As can be seen, performance improvement using the present approach is statistically very significant with 10 wins, 4 ties and only 1 loss with an average increased margin over 0.6%. In particular, C45-IFRC works well generally across the data sets with a different dimensionality, achieving 12 top results out of 16. Superiority in performance of the fuzzy rules produced using the proposed approach over those generated by FURIA is also statistically reflected in the last column of Table. 5.3, where C45-IFRC clearly beats C45-FURIA with 7 wins, 7 ties and only 1 loss. In contrast, the performance of the fuzzy rule bases generated by FURIA is even worse than its original crisp counterpart, with t-test results barely being equal.

Table 5.4 lists the results with unordered RIPPER used as the initial rule generator. The performance improvement owing to the use of the proposed algorithm is also significant with 8 wins, 6 ties and 2 losses, albeit having 2 wins fewer than the number achieved by FURIA. Different behaviours of FURIA in fuzzifying two different types of crisp rule bases (returned by C4.5 and UR, respectively) can be observed. This is because UR works by searching for fuzzified outcomes for one antecedent variable at a time in a brute-force way, thereby meeting the underlying strategy taken by FURIA, whilst C4.5 works over all individual attributes by one go. Nevertheless, the proposed approach is shown to be able to work with both strategies, leading to significant performance improvements.

As each of the original crisp rules points to different places where potentially desirable fuzzy rules may exist, the quality of preliminary crisp rules has an obvious impact upon the final generated fuzzy rules, as illustrated above. Thus, any direct attempt to compare the performances between the two fuzzy rule bases produced by C45-IFRC and UR-IFRC makes little sense, given their very different starting points. What is important is that they both achieve improved performances using only predefined fuzzy sets, producing models of inherent interpretability.

Table 5.3: Rule base comparison with C4.5 as initial rule generator using 10×10 cross-validation with respect to pair-wise t-test results ($p = 0.05$), where v, -, and * indicate statistically better, same, and worse classification performance against generated fuzzy rule base

Data Set	C4.5	C45-IFRC	C45-FURIA	C45-IFRC v.s. C45-FURIA
appendicitis	82.79 ± 1.78	**84.34 ± 2.63** (*)	73.47 ± 13.93 (v)	(v)
banknote	97.96 ± 0.45	**98.63 ± 0.34** (*)	98.16 ± 0.37 (*)	(v)
blood	76.89 ± 0.82	**77.53 ± 0.48** (*)	59.59 ± 12.28 (v)	(v)
breast-cancer	94.13 ± 0.65	**95.15 ± 0.68** (*)	94.81 ± 0.55 (*)	(=)
column-2C	79.52 ± 1.74	**80.16 ± 2.16** (=)	79.70 ± 1.77 (=)	(=)
column-3C	79.81 ± 2.05	77.51 ± 1.90 (v)	**79.93 ± 2.17** (=)	(*)
ionosphere	**87.00 ± 1.19**	86.80 ± 0.72 (=)	86.62 ± 1.26 (v)	(=)
iris	93.33 ± 1.30	**95.32 ± 0.54** (*)	93.33 ± 1.17 (=)	(v)
liver-disorders	63.08 ± 2.45	**64.83 ± 1.64** (*)	63.16 ± 2.13 (=)	(v)
mammographic	**82.03 ± 0.66**	79.13 ± 0.79 (v)	81.49 ± 1.04 (=)	(*)
new-thyroid	91.35 ± 1.52	**91.88 ± 1.20** (=)	91.54 ± 1.39 (=)	(=)
parkinsons	**84.48 ± 2.26**	84.33 ± 1.11 (=)	84.42 ± 2.24 (=)	(=)
pima-diabetes	73.89 ± 0.77	**75.05 ± 0.89** (*)	74.22 ± 0.81 (*)	(v)
seeds	90.38 ± 1.10	**91.37 ± 1.25** (*)	90.61 ± 0.95 (=)	(=)
sonar	70.23 ± 3.36	**72.59 ± 4.21** (*)	70.67 ± 3.44 (=)	(v)
wdbc	93.76 ± 0.64	**94.30 ± 0.53** (*)	93.87 ± 0.64 (=)	(=)
Summary (*/-/v)	83.789	**84.308** (10/4/2)	82.224 (3/10/3)	(2/7/7)

5.5.3 Comparison with Alternative Interpretable Fuzzy Rule-based Learning Classifiers

Performance of both classifiers implemented using the two resultant fuzzy rule bases (by C45-IFRC and UR-IFRC) is compared against 7 alternative fuzzy learning classifiers which also induce interpretable fuzzy rules with only fixed and uniformly divided quantity space, including: PTTD [136, 135], GP-COACH [14], SLAVE2 [54, 57], FH-GBML [68, 78], SGERD [105], MOGUL [34] and QSBA [142, 128], which have been reviewed in Chapter 2.3. The results on classification accuracy are summarised in Table 5.5, and the corresponding t-test outcomes are shown in Table 5.6.

Table 5.4: Rule base comparison with UR as initial rule generator using 10×10 cross-validation with respect to pair-wise t-test results ($p = 0.05$), where v, -, and * indicate statistically better, same, and worse classification performance against generated fuzzy rule base

Data Set	**UR**	**UR-IFRC**	**UR-FURIA**	**UR-IFRC v.s. UR-FURIA**
appendicitis	85.79 ± 1.80	**86.83 ± 1.70** (*)	85.80 ± 1.92 (=)	(*)
banknote	98.40 ± 0.22	98.75 ± 0.22 (*)	**99.12 ± 0.22** (*)	(v)
blood	**78.02 ± 0.55**	77.82 ± 1.11 (=)	**78.02 ± 0.55** (=)	(=)
breast-cancer	94.16 ± 0.47	**95.90 ± 0.39** (*)	94.96 ± 0.41 (*)	(*)
column-2C	81.90 ± 1.90	81.00 ± 2.07 (=)	**82.39 ± 1.67** (*)	(v)
column-3C	75.54 ± 0.88	**78.76 ± 1.68** (*)	77.52 ± 1.57 (*)	(*)
ionosphere	86.76 ± 1.07	85.58 ± 1.84 (=)	**87.35 ± 1.37** (=)	(v)
iris	92.59 ± 1.23	**95.59 ± 0.56** (*)	94.33 ± 0.72 (*)	(*)
liver-disorders	66.97 ± 2.18	64.86 ± 2.09 (v)	**68.79 ± 2.00** (*)	(v)
mammographic	82.31 ± 0.34	78.46 ± 1.09 (v)	**82.53 ± 0.49** (=)	(v)
new-thyroid	94.28 ± 0.72	94.29 ± 0.85 (=)	**94.84 ± 0.86** (*)	(=)
parkinsons	88.30 ± 1.98	87.02 ± 1.34 (=)	**89.87 ± 1.32** (*)	(v)
pima-diabetes	74.82 ± 0.87	**75.27 ± 0.69** (=)	74.93 ± 1.03 (=)	(=)
seeds	90.48 ± 0.78	**92.66 ± 1.52** (*)	92.05 ± 0.68 (*)	(=)
sonar	74.82 ± 2.26	**77.39 ± 1.98** (*)	75.49 ± 2.09 (=)	(*)
wdbc	94.44 ± 0.55	94.98 ± 0.73 (*)	**94.99 ± 0.45** (*)	(=)
Summary (*/-/v)	84.974	85.323 (8/6/2)	**85.811** (10/6/0)	(6/5/5)

5.5.3.1 C45-IFRC vs. Alternatives

Although regarding individual data sets C45-IFRC may not be a top performer, its average performance across all tested datasets is higher than that achieved by any of the seven alternatives. In terms of statistical t-test, it ties with the two (GP-COACH and SLAVE2) and significantly beats the other five (e.g., C45-IFRC has 15 wins, 1 tie and no losses as compared to SGERD). Yet, GP-COACH and SLAVE2 learn fuzzy rules involving the use of disjunctive norm of fuzzy sets, i.e., they allow multiple fuzzy sets to be compounded to describe a single domain variable. This not only greatly expands the solution search space, but also causes the learned rules to become more complicated and hence less comprehensible.

5.5.3.2 UR-IFRC vs. Alternatives

The performance of UR-IFRC is even more superior than C45-IFRC in terms of their relative performance against the seven alternatives. Again, it has achieved the best average accuracy amongst all, and this is further supported with statistically significant better results throughout, even beating GP-COACH and SLAVE2, the two best performers amongst the seven, with substantially more wins than losses.

5.5.4 Model Complexity

Table 5.7 presents an empirical analysis of the complexity of learned interpretable fuzzy rule bases, in terms of average number of antecedent conditions (*Cond*) per fuzzy rule, and average number of rules (*Rul*) per rule base.

For *Cond*, PTTD and SGERD return the most compact rules, with both learning fuzzy rules involving fewer than 2 antecedent conditions. Following these two, C45-IFRC also enjoys high structural interpretability, being able to learn rules of the third shortest on average in length. UR-IFRC also learns short fuzzy rules employing only fewer than 4 antecedent variables on average. In contrast, MOGUL and QSBA have a fixed length of any fuzzy rule as it is set according to the problem dimensionality. Note that for GP-COACH and SLAVE2, *Cond* only counts the number of the antecedent variables appearing in the rule, not the additional complexity incurred due to their use of compounded fuzzy terms in describing the variables.

For *Rul*, PTTD and QSBA return rule bases with the smallest size, due to their imposed heuristic nature of setting the number of rules to the number of the classes. However, the interpretability of QSBA model is poor since the rules it returns are very complicated, involving all variables for each rule. In general, both PTTD and SGERD tend to generate most compact rule bases with not only very small rule sizes but also short rules. Yet, their classification performances are poor compared with that of the proposed approach. C45-IFRC is able to learn rule bases of a small cardinality (returning fewer than 12 rules required on average across the 16 data sets), simpler than those returned by GP-COACH, MOGUL, and FH-GBML. One possible reason that UR-IFRC learns rule bases with a bigger size may be due to the fact that UR is set to generate rules with more antecedent variables using a bigger maxDepth parameter value as indicated in Table 5.2.

Table 5.5: Comparison against interpretable fuzzy rule-based classifiers using 10×10 cross-validation with respect to classification accuracy (%), where bold figures signify overall top results for each data set

Data Set	C45-IFRC	UR-IFRC	PTTD	GP-COACH	SLAVE2	MOGUL	FH-GBML	SGERD	QSBA
appendicitis	84.34 ± 2.63	**86.83 ± 1.70**	86.66 ± 0.89	86.27 ± 1.81	84.36 ± 1.52	76.97 ± 2.47	84.33 ± 2.06	85.04 ± 1.01	86.48 ± 0.95
banknote	98.63 ± 0.34	98.75 ± 0.22	84.52 ± 0.15	91.59 ± 0.65	91.63 ± 0.17	**98.99 ± 0.22**	98.36 ± 0.41	84.20 ± 0.36	82.19 ± 0.31
blood	77.53 ± 0.48	77.82 ± 1.11	77.42 ± 0.13	76.17 ± 0.24	76.51 ± 0.14	**78.18 ± 0.56**	77.04 ± 0.30	76.22 ± 0.18	66.58 ± 1.32
breast-cancer	95.15 ± 0.68	95.90 ± 0.39	95.35 ± 0.23	95.78 ± 0.45	96.15 ± 0.35	75.46 ± 0.81	**96.21 ± 0.56**	93.49 ± 0.36	95.65 ± 0.16
column-2c	80.16 ± 2.16	81.00 ± 2.07	74.65 ± 1.48	75.52 ± 1.12	79.00 ± 1.04	77.03 ± 1.34	**81.35 ± 1.84**	70.00 ± 1.22	69.42 ± 0.34
column-3c	77.51 ± 1.90	**78.76 ± 1.68**	75.16 ± 0.79	74.65 ± 2.16	74.94 ± 0.71	75.45 ± 1.78	77.87 ± 1.31	70.77 ± 1.25	70.94 ± 0.42
ionosphere	86.80 ± 0.72	85.58 ± 1.84	74.04 ± 1.16	**90.09 ± 0.96**	89.61 ± 1.35	31.13 ± 1.11	48.13 ± 2.35	73.65 ± 1.77	82.96 ± 0.94
iris	95.32 ± 0.54	95.59 ± 0.56	95.60 ± 0.34	**97.67 ± 0.35**	96.60 ± 0.38	93.33 ± 1.40	94.20 ± 1.18	94.27 ± 1.00	91.67 ± 0.35
liver-disorders	64.83 ± 1.64	64.86 ± 2.09	**67.75 ± 1.52**	58.99 ± 0.71	60.76 ± 0.71	58.77 ± 2.75	65.89 ± 1.77	59.01 ± 1.10	57.69 ± 1.02
mammographic	79.13 ± 0.79	78.46 ± 1.09	76.23 ± 0.44	78.95 ± 0.42	78.58 ± 0.62	78.85 ± 0.73	**80.87 ± 0.64**	77.39 ± 0.20	80.58 ± 0.26
new-thyroid	91.88 ± 1.20	**94.29 ± 0.85**	88.75 ± 0.53	91.78 ± 0.65	91.56 ± 0.50	93.50 ± 1.18	92.63 ± 0.91	87.23 ± 0.58	93.12 ± 0.61
parkinsons	84.33 ± 1.11	87.02 ± 1.34	85.03 ± 0.71	**87.27 ± 1.03**	86.82 ± 1.25	62.38 ± 2.71	81.26 ± 1.10	82.28 ± 1.53	81.07 ± 1.22
pima-diabetes	75.05 ± 0.89	75.27 ± 0.69	74.13 ± 0.36	75.13 ± 0.87	**75.38 ± 0.71**	71.26 ± 0.77	73.72 ± 1.03	70.17 ± 0.69	73.45 ± 0.65
seeds	91.37 ± 1.25	**92.66 ± 1.52**	89.43 ± 1.00	91.67 ± 1.19	90.00 ± 1.31	91.65 ± 1.23	90.76 ± 1.71	86.52 ± 0.87	81.57 ± 0.64
sonar	72.59 ± 4.21	77.39 ± 1.98	67.77 ± 1.57	**78.86 ± 3.01**	78.07 ± 1.83	5.91 ± 1.34	45.06 ± 3.17	70.09 ± 2.38	74.44 ± 0.82
wdbc	94.30 ± 0.53	**94.98 ± 0.73**	93.25 ± 0.54	94.41 ± 0.50	94.66 ± 0.44	81.62 ± 0.92	90.47 ± 0.96	91.86 ± 0.67	91.35 ± 0.30
Summary	84.308	**85.323**	81.609	84.049	84.039	71.905	79.885	79.512	79.946

Table 5.6: Comparison against fuzzy rule-based classifiers using 10 × 10 cross-validation with respect to pair-wise t-test results (p = 0.05), where v, -, and * indicate statistically better, same, and worse classification performance against the proposed approach

	C45-IFRC							UR-IFRC						
Data Sets	**PTTD**	**GP-COACH**	**SLAVE2**	**MOGUL**	**FH-GBML**	**SGERD**	**QSBA**	**PTTD**	**GP-COACH**	**SLAVE2**	**MOGUL**	**FH-GBML**	**SGERD**	**QSBA**
appendicitis	v	=	=	*	=	=	v	=	=	*	*	*	*	=
banknote-authentication	*	*	*	v	*	*	*	*	*	*	v	*	*	*
blood	=	*	*	v	*	*	*	=	*	*	=	*	*	*
breast-cancer-wisconsin	=	v	=	*	v	*	v	*	=	v	*	=	*	=
column-2C	*	*	*	*	=	*	*	*	*	*	*	=	*	*
column-3C	*	*	v	*	=	*	*	*	*	*	*	=	*	*
ionosphere	*	v	v	*	*	*	*	*	v	v	*	*	*	*
iris	=	v	*	*	*	*	*	=	v	v	*	*	*	*
liver-disorders	v	*	=	*	=	*	*	v	*	*	*	=	*	*
mammographic	*	=	=	=	v	*	v	*	=	=	=	v	*	v
new-thyroid	*	=	v	v	=	*	v	*	*	*	*	*	*	*
parkinsons	=	v	=	*	*	*	*	*	=	=	*	*	*	*
pima-diabetes	*	=	*	*	*	*	*	*	=	=	*	*	*	*
seeds	*	=	=	=	=	*	*	*	=	*	*	*	*	*
sonar	*	v	v	*	*	*	=	*	=	=	*	*	*	*
wdbc	*	=	v	*	*	*	*	*	*	=	*	*	*	*
Summary (*/-/v)	(10/4/2)	(5/6/5)	(5/6/5)	(11/2/3)	(8/6/2)	(15/1/0)	(11/1/4)	(12/3/1)	(7/7/2)	(8/5/3)	(13/2/1)	(11/4/1)	(16/0/0)	(13/2/1)

Table 5.7: Comparison against fuzzy rule-based classifiers using 10 × 10 cross-validation with respect to average number of rules per rule base and average number of antecedent attributes per rule

	C45-IFRC		UR-IFRC		PTTD		GP-COACH		SLAVE2		MOGUL		FH-GBML		SGERD		QSBA	
Data Sets	**Rul**	**Cond**	**Rul**	**Cond**	**Rul**	**Cond**	**Rul**	**Cond**	**Rul**	**Cond**	**Rul**	**Cond**	**Rul**	**Cond**	**Rul**	**Cond**	**Rul**	**Cond**
appendicitis	2.76	1.97	5.17	2.00	2.00	1.50	8.71	7.26	5.56	2.94	47.20	7.00	18.61	4.86	2.48	2.00	2.00	35.00
banknote-authentication	28.59	2.83	35.42	2.81	2.00	0.57	8.27	4.41	6.80	2.60	117.86	4.00	25.50	3.46	3.30	2.00	2.00	20.00
blood	7.51	2.67	7.35	2.25	2.00	1.50	7.38	3.92	4.19	1.95	129.11	4.00	17.55	3.41	2.69	2.00	2.00	20.00
breast-cancer-wisconsin	18.42	3.63	76.15	4.32	2.00	1.50	11.66	4.75	14.87	4.04	375.10	9.00	22.64	5.62	2.18	1.53	2.00	45.00
column-2C	8.12	3.36	14.8	3.14	2.00	1.50	4.70	4.99	5.65	2.96	157.75	6.00	18.88	4.74	2.98	2.00	2.00	30.00
column-3C	7.6	3.21	15.98	3.68	3.00	3.00	8.56	5.28	6.66	3.98	202.97	6.00	17.55	4.71	3.63	2.00	3.00	30.00
ionosphere	7.05	3.54	24.01	6.77	2.00	1.50	11.45	12.38	14.85	3.96	122.00	33.00	20.06	16.27	4.14	1.39	2.00	165.00
iris	4.35	1.45	8.73	2.12	3.00	3.00	3.04	2.14	5.86	1.72	68.70	4.00	23.18	3.24	3.63	1.99	3.00	20.00
liver-disorders	14.19	3.52	27.84	3.68	2.00	1.50	16.51	7.61	9.13	4.34	221.66	6.00	21.27	4.85	2.84	2.00	2.00	30.00
mammographic	10.9	3.15	5.46	2.79	2.00	1.50	13.73	3.78	8.58	2.95	639.79	5.00	23.75	3.75	2.02	2.00	2.00	25.00
new-thyroid	8.77	2.6	14.73	2.6	3.00	3.00	7.02	2.87	7.57	2.26	105.37	5.00	17.76	1.48	3.26	2.00	3.00	25.00
parkinsons	12.48	4.93	20.35	5.07	2.00	1.50	12.41	8.99	9.24	5.02	105.00	22.00	14.80	11.81	2.47	2.00	2.00	110.00
pima-diabetes	14.66	3.19	24.01	4.08	2.00	1.50	62.17	11.19	15.16	4.84	417.60	8.00	22.88	5.49	3.71	2.00	2.00	40.00
seeds	9.74	2.55	24.04	2.77	3.00	3.00	12.75	5.14	11.35	3.45	137.45	7.00	19.93	5.03	3.91	2.00	3.00	35.00
sonar	16.99	7.68	38.73	4.94	2.00	1.50	33.15	15.56	21.76	6.23	103.80	60.00	13.73	31.12	3.07	2.00	2.00	300.00
wdbc	13.4	4.98	43.83	8.55	2.00	1.50	9.16	4.74	9.16	5.60	281.90	30.00	16.53	15.22	3.57	2.00	2.00	150.00
Average	11.596	3.454	24.163	3.848	2.250	1.817	14.417	6.562	9.774	3.678	202.079	13.500	19.664	7.817	3.118	1.932	2.250	67.500

5.5.5 Comparison with Non-Fuzzy-Rule-Based Classifiers

In addition to comparing against alternative fuzzy learning classifiers, the performance of the proposed approach is further compared with another 6 popular learning classifiers which are non-fuzzy-rule-based. Table 5.8 summarises the classification accuracy and Table 5.9 shows the t-test results. The six compared methods are: SMO [137], a sequential optimisation algorithm for building support vector machines with polynomial kernel function; IBk [153], the classical *k*-nearest neighbour approach, where an instance is classified by a majority vote of its neighbours; FRNN [81], a fuzzy-rough set-based nearest neighbour classification algorithm, which classifies instances based on their membership to lower and upper approximations of the decision classes; NB [112], a probabilistic learning classifier, based on direct application of Bayesian theorem with strong independence assumptions; and RIPPER [31], the classical rule induction algorithm, with rule pruning and optimisation process performed to fine-tune the learned rules including a default rule added for the most frequent class; and NFC [21], an optimised neuro-fuzzy classifier.

Compared with NB and RIPPER, UR-IFRC performs significantly better, in terms of both the average accuracy and the t-test results. Such clear wins are also achieved by C45-IFRC, compared to NB and RIPPER. The performance gap between C45-IFRC and the well-designed and robust SVM classifier is only fewer than 0.1%, with statistical results close to those of SVM and nearest neighbour-based learning classifiers. Whereas UR-IFRC does not outperform the rest for any data set, it achieves better average accuracy than SMO, IBK and FRNN, supported with better statistical results, and a statistically equal performance with NFC. Collectively, the resultant fuzzy rule bases have demonstrated a promising performance that is at least comparable to the popular, well-established non-fuzzy-rule-based classifiers. Importantly, such an excellent performance is achieved using only fixed quantity space with interpretable inference results, forming a sharp contrast with SVM and nearest neighbour-based learning classifiers.

5.5.6 Effect of Local Rule Selection

The above experimental results have demonstrated the promising performance of the proposed approach, in terms of both classification accuracy and model interpretability

Table 5.8: Comparison against non-fuzzy-rule-based classifiers using 10 × 10 cross-validation with respect to classification accuracy (%), where bold figures signify overall top results for each data set

Data Set	C45-IFRC	UR-IFRC	RIPPER	NB	SMO	IBk	FRNN	NFC
appendicitis	84.34 ± 2.63	86.83 ± 1.70	79.51 ± 2.61	85.21 ± 0.52	**87.42 ± 0.77**	80.96 ± 1.22	83.72 ± 1.42	85.56 ± 1.75
banknote	98.63 ± 0.34	98.75 ± 0.22	98.41 ± 0.32	84.01 ± 0.14	97.97 ± 0.07	99.85 ± 0.00	99.85 ± 0.00	**99.94 ± 0.05**
blood	77.53 ± 0.48	77.82 ± 1.11	69.60 ± 0.97	75.28 ± 0.37	76.18 ± 0.09	71.06 ± 0.76	71.50 ± 0.85	**78.46 ± 0.48**
breast-cancer	95.15 ± 0.68	95.90 ± 0.39	93.71 ± 1.08	96.12 ± 0.08	**96.77 ± 0.18**	95.35 ± 0.24	96.45 ± 0.19	95.68 ± 0.20
column-2c	80.16 ± 2.16	81.00 ± 2.07	76.71 ± 0.81	77.87 ± 0.27	78.90 ± 0.81	81.06 ± 1.19	78.68 ± 1.26	**85.00 ± 0.57**
column-3c	77.51 ± 1.90	78.76 ± 1.68	76.81 ± 2.43	82.58 ± 0.59	76.10 ± 0.83	76.74 ± 0.93	75.68 ± 0.84	**84.23 ± 0.94**
ionosphere	86.80 ± 0.72	85.58 ± 1.84	84.04 ± 2.13	83.78 ± 0.21	82.96 ± 0.76	85.17 ± 0.78	**89.22 ± 0.45**	85.62 ± 1.78
iris	95.32 ± 0.54	95.59 ± 0.56	94.40 ± 1.61	95.53 ± 0.45	**96.27 ± 0.47**	95.40 ± 0.38	94.07 ± 0.38	93.80 ± 1.81
liver-disorders	64.83 ± 1.64	64.86 ± 2.09	62.35 ± 2.86	54.89 ± 1.14	57.98 ± 0.24	62.22 ± 1.15	62.81 ± 0.97	**70.35 ± 1.17**
mammographic	79.13 ± 0.79	78.46 ± 1.09	78.96 ± 1.05	77.64 ± 0.53	79.39 ± 0.34	74.91 ± 0.60	74.11 ± 0.56	**81.56 ± 0.37**
new-thyroid	91.88 ± 1.20	94.29 ± 0.85	88.99 ± 1.66	96.92 ± 0.25	89.30 ± 0.51	96.93 ± 0.57	**97.39 ± 0.79**	95.27 ± 2.25
parkinsons	84.33 ± 1.11	87.02 ± 1.34	88.24 ± 1.99	70.14 ± 0.59	87.00 ± 0.61	**95.90 ± 0.41**	93.96 ± 0.46	83.23 ± 0.45
pima-diabetes	75.05 ± 0.89	75.27 ± 0.69	66.88 ± 1.62	75.76 ± 0.44	**76.80 ± 0.24**	70.62 ± 0.84	69.07 ± 0.89	75.68 ± 0.86
seeds	91.37 ± 1.25	92.66 ± 1.52	87.52 ± 1.25	90.53 ± 0.47	93.57 ± 0.34	**93.86 ± 0.76**	93.09 ± 0.82	91.14 ± 1.17
sonar	72.59 ± 4.21	77.39 ± 1.98	73.92 ± 2.39	67.71 ± 1.08	76.59 ± 1.94	**86.17 ± 0.84**	85.25 ± 0.62	76.00 ± 1.73
wdbc	94.30 ± 0.53	94.98 ± 0.73	93.82 ± 0.80	93.31 ± 0.17	**97.54 ± 0.22**	95.64 ± 0.28	95.29 ± 0.39	94.52 ± 0.57
Summary	84.308	85.323	82.117	81.704	84.421	85.116	85.009	**86.003**

Table 5.9: Comparison against non-fuzzy-rule-based classifiers using 10 × 10 cross-validation with respect to pair-wise t-test results (p = 0.05), where v, -, and * indicate statistically better, same, and worse classification performance against the proposed approach

	C45-IFRC						UR-IFRC					
Data Sets	**RIPPER**	**NB**	**SMO**	**IBk**	**FRNN**	**NFC**	**RIPPER**	**NB**	**SMO**	**IBk**	**FRNN**	**NFC**
appendicitis	*	=	v	*	=	=	*	*	=	*	*	*
banknote-authentication	=	*	*	v	v	v	*	*	*	v	v	v
blood	*	*	*	*	*	v	*	*	*	*	*	=
breast-cancer-wisconsin	*	v	v	=	v	v	*	=	v	*	v	=
column-2C	*	*	=	=	*	v	*	*	*	=	*	v
column-3C	=	v	*	=	*	v	*	v	*	*	*	v
ionosphere	*	*	*	*	v	*	*	*	*	=	v	=
iris	*	=	v	=	*	*	*	=	v	=	*	*
liver-disorders	*	*	*	*	*	v	*	*	*	*	*	v
mammographic	=	*	=	*	*	v	=	*	v	*	*	v
new-thyroid	*	v	*	v	v	v	*	v	*	v	v	=
parkinsons	v	*	v	v	v	*	=	*	=	v	v	*
pima-diabetes	*	v	v	*	*	=	*	=	v	*	*	=
seeds	*	*	v	v	v	=	*	*	v	v	=	*
sonar	=	*	v	v	v	v	*	*	=	v	v	=
wdbc	*	*	v	v	v	=	*	*	v	v	=	*
Summary (*/-/v)	(11/4/1)	(10/2/4)	(6/2/8)	(6/4/6)	(7/1/8)	(3/4/9)	(14/2/0)	(11/3/2)	(7/3/6)	(7/3/6)	(8/2/6)	(5/6/5)

(thanks to the use of only predefined fuzzy sets and the induction of compact rules and rulesets). The high comprehensibility is achieved without embedding any sophisticated criterion in the final GA-based tuning step (by setting $w_i = 0$ as indicated in Table 5.2). However, such compact and transparent rule bases cannot be obtained without the stage of local rule selection through functional generalisation. This is confirmed with the further experimental investigations as reported below.

In conducting this purposefully devised experimentation, 3 different assignments for the interpretability weight w_i, as given in Eqn. (5.20) are used, namely: 0.0, 0.1 and 1.0. A single 10-CV run is performed for 5 data sets with C45-IFRC. Results are averaged, and analysed, in terms of: training accuracy (*Trn*), testing accuracy (*Tst*), average number of rules (R_1) after Stage 1 (i.e., the average number of potential fuzzy rules after heuristic mapping procedure), average number of rules (R_2) after Stage 2 (i.e., the number of all returned fuzzy rules with the local rule selection procedure), and average number of rules (R_3) after Stage 3 (i.e., the size of the final ruleset). The average number of antecedent variables, or the conditions (*Cond*), per resultant rule is also recorded together with the execution time (*Time*) for each complete 10-CV run.

As shown in Table 5.10, the reduction in the number of rules obtained after local rule selection is significant, R_2 is at least 10 times smaller than R_1 (for the data set column-3C, it is over 20 times smaller). Such reduction still results in a highly compact rule base, even when the interpretability weight is not included in the subsequent genetic tuning. The sizes of the resultant rule bases after running Stage 2 are generally over 2 times smaller than those without when $w_i = 0.1$, and much less when $w_i = 0$ (more than 10 times).

Recall that such substantial reduction is designed subject to functional generalisation, without loss in the performance of selected rules. However, if Stage 2 is not run, when $w_i = 1$, although a very small rule base with short rules may be returned, the classification performance is significantly decreased. Better performances are generally achieved when $w_i = 0$ or $w_i = 0.1$ in terms of accuracy, yet all of which are still worse than those with Stage 2 running. In particular, regarding the data sets column-2C and column-3C, the resultant classification accuracies are far worse than those achievable with local selection procedure being on. As an example, Figure 5.5 shows a single GA run (with the compared settings as illustrated in the figure), regarding both training and testing accuracy. When running with Stage 2 it only

takes a few generations to converge. In situations where Stage 2 is not implemented, the plot on the testing accuracy oscillates before it settles down around 20th generation when $w_i = 0$; it takes more than 100 generations to converge in case of $w_i = 0.1$; whereas when interpretability is weighted significantly higher, GA fails to find solutions with good performance.

Running the local rule section procedure requires additional computation in search, where GA needs to run multiple times (with the number depending on that of the given crisp rules) as overheads. However, in real applications of the proposed approach, such multiple search attempts can be realised in parallel in order to reduce the otherwise required time for series implementation. Despite the time measured in this experiment is obtained by running multiple GAs sequentially, the result is very promising as such additional cost helps reduce the overall run time that the final genetic tuning will spend. As shown by the results, the overall run time cost is generally much smaller than that is required without running local rule selection (when $w_i = 0$ or $w_i = 0.1$).

5.6 Summary

Owing to the necessity of incorporating consistent domain expertise by the use of predefined fuzzy sets, this chapter has proposed a novel approach to generating interpretable fuzzy classification rules. For a given classification problem, simple crisp rules are utilised for initialisation, with each of them pointing to the model sub-spaces where desirable fuzzy rules potentially exist. This is followed by a heuristic mapping procedure that converts each preliminary crisp rule into a set of interpretable fuzzy rules involving only the predefined fuzzy sets, ensuring semantic interpretability. A local rule selection procedure is then performed to obtain a compact subset of initially mapped fuzzy rules that jointly generalise the capability of the underlying crisp rule. A fine grain tuning of all selected subsets of fuzzy rules is finally carried out with a conventional GA, resulting in an accurate and interpretable fuzzy rule-based classifier with a simplified structure.

Systematic experimental examinations of the proposed approach have been carried out, involving the use of two different crisp rule generation mechanisms for initialisation, over 16 benchmark datasets, in comparison with 7 alternative

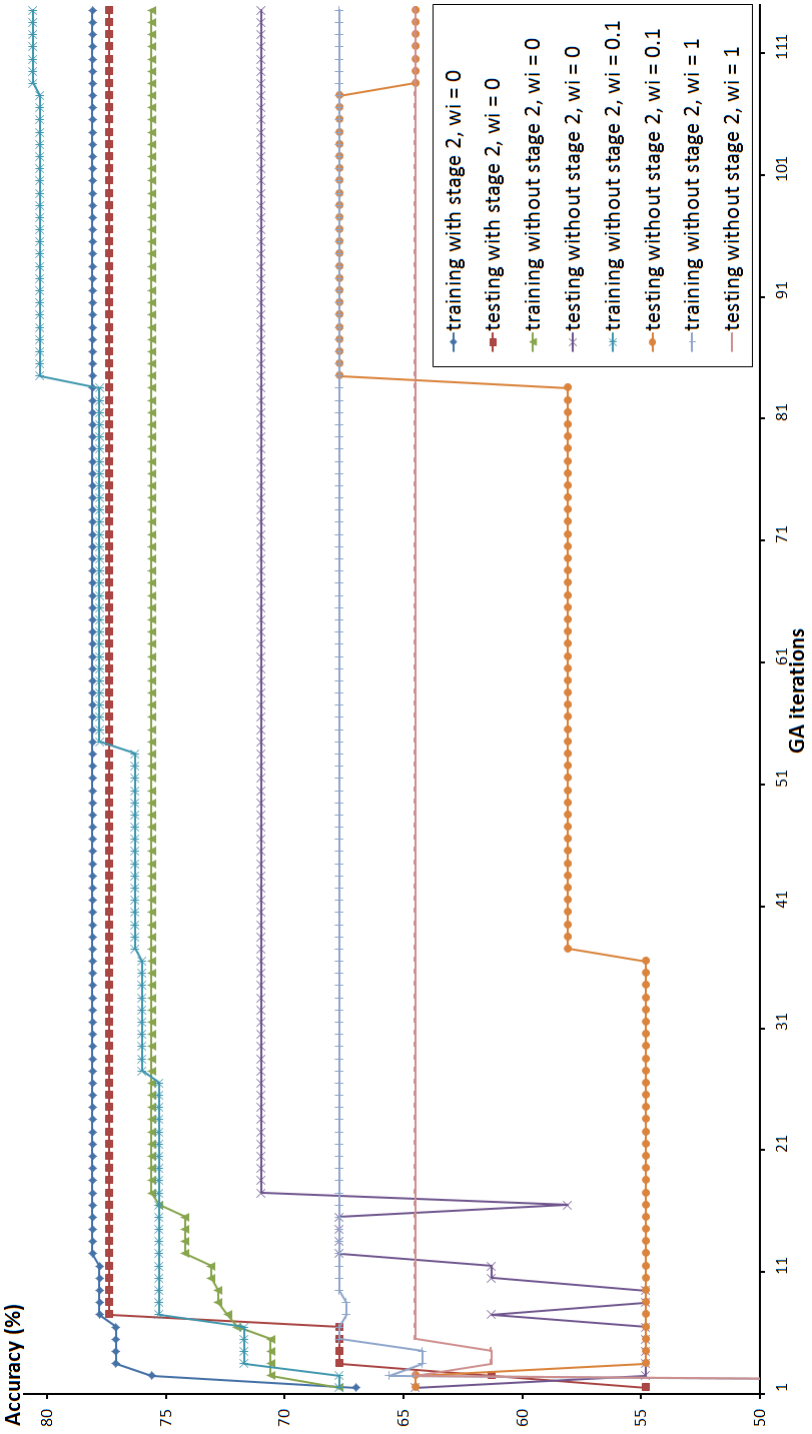

Figure 5.5: Example genetic tuning runs (on the data set column-2C)

Table 5.10: Analysis of local rule selection

Data sets	Setup	Trn	Tst	R_1	R_2	R_3	Cond	Time
column-2C	with stage 2, $w_i = 0$	81.5	77.1	101.1	7.5	5.8	2.2	29.4
	without stage 2, $w_i = 0$	75.9	71.6	101.1	101.1	48.4	2.7	109.1
	without stage 2, $w_i = 0.1$	74.4	70.3	101.1	101.1	16.6	2.7	35.7
	without stage 2, $w_i = 1$	64.8	61.6	101.1	101.1	1.6	1.0	3.4
column-3C	with stage 2, $w_i = 0$	78.5	76.5	141.6	6.0	4.2	2.0	34.1
	without stage 2, $w_i = 0$	54.0	50.7	141.6	141.6	30.2	2.5	52.4
	without stage 2, $w_i = 0.1$	55.6	56.8	141.6	141.6	11.7	2.2	24.3
	without stage 2, $w_i = 1$	46.8	50.3	141.6	141.6	3.8	1.4	7.0
ionosphere	with stage 2, $w_i = 0$	89.5	85.2	93.0	8.5	8.1	2.3	15.4
	without stage 2, $w_i = 0$	88.8	82.6	93.0	93.0	79.1	2.8	93.1
	without stage 2, $w_i = 0.1$	89.8	84.4	93.0	93.0	32.4	2.6	62.6
	without stage 2, $w_i = 1$	72.6	73.5	93.0	93.0	6.8	1.5	18.5
seeds	with stage 2, $w_i = 0$	93.5	91.0	85.8	8.0	7.3	2.1	18.0
	without stage 2, $w_i = 0$	91.5	89.0	85.8	85.8	56.1	2.6	60.3
	without stage 2, $w_i = 0.1$	92.0	90.0	85.8	85.8	21.9	2.2	27.7
	without stage 2, $w_i = 1$	57.5	55.2	85.8	85.8	5.3	1.7	5.7
wdbc	with stage 2, $w_i = 0$	95.3	94.2	133.2	12.8	9.7	2.5	78.1
	without stage 2, $w_i = 0$	95.1	93.3	133.2	133.2	100.6	2.8	258.1
	without stage 2, $w_i = 0.1$	95.0	93.0	133.2	133.2	23.2	2.5	110.9
	without stage 2, $w_i = 1$	77.7	78.8	133.2	133.2	5.6	1.6	24.7

fuzzy learning classifiers and 6 popular non-fuzzy-rule-based classifiers. The results have revealed the overall superiority of the proposed approach over the rest. In particular, the introduced functional generalisation method has proven effective in the production of the fuzzy rule bases, which are of high interpretability, being compact with short rules and exhibiting semantic comprehensibility.

Chapter 6

Case Study: Journal Ranking with Induced Fuzzy Rules

Further to te systematic investigation into the efficacy of the proposed approach in Chapter 5, for dealing with benchmark datasets, this chapter presents a study of applying the work to a real-world problem – academic journal ranking [144]. This is inspired by the fact that academic journal ranking has recently drawn much attention in support of research quality assessment [12]. The rank of a journal typically implies its prestige, impact and even difficulty level of having a paper accepted for publication. Therefore, this chapter presents a case study, concentrating on the exhibition of learned fuzzy rule models being indeed comprehensible, in terms of both semantics and structure.

6.1 JCR Indicators and Expert-provided Journal Ranking

The assessment of research output quality is a serious issue which relates to many educational and financial problems such as evaluation of research projects and distribution of research funding. Recently, many countries have implemented their own national projects for academic output assessment. Examples include the Research Excellence Framework (REF) in the UK [152] and the Excellence in Research for

Australia (ERA) [151]. One significant aspect of research quality assessment may involve academic journal ranking, though the efficacy of using such information is not universally agreed upon [41]. However, the rank of a journal typically implies its prestige, impact, and even difficulty of having a paper accepted for publication in it. Nevertheless, the general concept of academic journal quality is a multi-faceted notion. Conventionally, assessing the quality of research publications is done through peer-review that is carried out by experts in the relevant research areas. It is almost inevitable that such expertise-based assessment is financially intensive and time consuming. For example, in the ERA, over 700 experts were employed to make a journal ranking list. Although the sophisticated results judged by the experts can be very useful in, for instance, directing government research funding and reflecting appropriate use of public funds, the running costs involved make it impracticable to implement such approaches frequently.

The Journal Citation Report (JCR) [93] has a long history of applications for researchers and librarians in choosing their reading lists. All impact indicator score calculations in JCR are based on the same set of journals, namely journals which are indexed by Web of Science. Six indicators that are reported in JCR (2010) are selected as the indicators to construct journal ranking data sets. These are [13]:

- Total Cites (TC): number of times the journal was cited in a year;
- Impact Factor (IF): ratio of cites to recent articles to the number of recent articles, with the recency being defined within a 2-year window;
- 5-year (5IF): the same as IF, but covering articles within a 5-year window;
- Immediacy Index (II): ratio of cites to the current articles over the number of those articles;
- Eigenfactor (Ei): similar to IF, but eliminating self-referencing and weighting journals by the amount of time elapsed before being cited;
- Article Influence (AI): ratio of the Eigenfactor score to the total number of articles considered.

Generally, all these six indicators assign greater scores to journals with more citations. Apart from the indicators included in JCR, many other indicators are

available from various of academic publication databases. Note that these indicators have their own characteristics. As briefly defined above, Eigenfactor is developed to eliminate the effect of self-citation while IF and 5-IF include self-citation. AI is developed to offset the size effect of journals while TC does not take the size of a journal into consideration. However complex the interactions between these indicators may appear, they are more likely to be complementary to one another than to cause contradictions between each other. For example, an excellent journal can have high scores both in Eigenfactor and IF, and a journal which performs badly in TC may also perform badly in IF. In other cases, a journal could have higher scores in several indicators than in others. Due to the fact that they are proposed to measure journal quality with different focuses, direct comparison of the individual scores owing to their use can be difficult.

The values of these objective impact indicators are given in the form of precise numerical values. However, when ranking is done through human peer evaluation, the values of these indicators, and also, the rankings themselves are commonly referred to using linguistic terms, with subjective underlying semantics. To develop a system that may imitate subjective human peer evaluation, the proposed approach is herein utilised to generate a set of meaningful linguistic ranking rules, using only expert-provided fuzzy term sets. To focus this justified experimentation, only journals from the Computer Science subject category indexed by the Web of Science (including Artificial Intelligence, Cybernetics, Hardware & Architecture, Information Systems, Interdisciplinary Applications, Software Engineering, Theories & Methods) are selected for this case study.

6.2 Fuzzy Set Partition Using Fuzzy *c*-means

In the absence of expert's knowledge, uniformly divided fuzzy sets are often adopted due to the common practice and being most interpretable from the shape point of view [108] in the literature. However, uniform partition may not reflect the true distribution of underlying data, therefore affecting the performance of the resulting fuzzy rules. Despite distribution of underlying data may not necessarily reflect overall domain expertise, it is still worth investigating the effect of predefined fuzzy sets obtained by way of data-driven partition. This is carried out in comparison with the uniform partitions previously employed before presenting rules that employ

such data-driven defined fuzzy sets. In particular, fuzzy c-means (FCM) [149] is herein employed to implement the fuzzification process, such that each of generated clusters is treated as a predefined fuzzy set which may be artificially associated with a linguistic label.

FCM is a method of clustering which allows a single data point to belong to multiple clusters simultaneously, smoothing the abrupt boolean boundaries that are often not natural or even counterintuitive. It is based on the minimisation of the objective function:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} \mu_{ij}^m ||x_i - c_j||^2, 1 \leqslant m < \infty \tag{6.1}$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the i-th of d-dimensional measured data, c_j is the d-dimension center of the cluster, and $||*||$ is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimisation of the objective function, with the update of membership u_{ij} and the cluster centers c_j by:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{||x_i - c_j||}{||x_i - c_k||})^{\frac{2}{m-1}}} \tag{6.2}$$

$$c_j = \frac{\sum_{i=1}^{N} \mu_{ij}^m \cdot x_i}{\sum_{i=1}^{N} \mu_{ij}^m} \tag{6.3}$$

The iteration will shop when $\max_{ij}\{|\mu_{ij}^{(k+1)} - \mu_{ij}^{(k)}|\} < \epsilon$, where ϵ is a termination criterion between 0 and 1, whereas k are the iteration steps. The procedure converges to a local minimum of the cost function.

Note that a possible drawback of employing FCM to implement fuzzification is that a data point's membership to a cluster is not monotonically decreasing along with its distance to the cluster center. Therefore, a modification precess is applied, ensuring that the membership does become monotonically deceasing with regard to its distance to the cluster center. Such a modification process can be implemented using the following two steps:

1. Given a set of clusters $D^l_{ji}, j = 1, \cdots, l$, generated for the variable $x_i, i \in 1, \cdots, n$, where l specifies the partition granularity and is set to $l = 2, \cdots, 5$, assign the membership $\mu_{D^l_{ji}}(x^p_i) = 0$ for each instance $\bar{x}^p$ if x^p_i is smaller than the centre of $D^l_{(j-1)i}, j = 2, \cdots, l$; assign the membership $\mu_{D^l_{ji}}(x^p_i) = 0$ if x^p_i is greater than the centre of $D^l_{(j+1)i}, j = 1, \cdots, l-1$.

2. For instance x^p, update its memberships to all the clusters within granularity l by normalisation:

$$\mu_{D^l_{ji}}(x^p_i) = \frac{\mu_{D^l_{ji}}(x^p_i)}{\sum_{j'=1}^{l} \mu_{D^l_{j'i}}(x^p_i)} \tag{6.4}$$

6.2.1 Performance Comparison between Grid Partitioning and Partitioning by FCM

To continue experiments from the preceding Chapter where Unordered Ripper (UR) is used as the initial rule generator for 16 benchmark datasets. The results in Table 5.4 indicates that 8 data sets out of the 16 benchmarks have not been able to significantly improve the performance while using the uniform partition. Therefore these 8 data sets are employed here for further exploitation, including *blood*, *column2C*, *ionosphere*, *mammographic*, *thyroid*, *parkinsons*, *diabetes*. This choice is deliberately made so as to illustrate the power of employing fine-tuned MFs in performing classification. Of course, this experimentation has on purpose ignored the issue of model interpretability.

Exact same experimental settings are used here as that used for uniform partition. UR-IFRC(FCM) shows results employing FCM to implement the fuzzification process as shown in Table. 6.1, where UR-IFRC presents the same results using uniform partition as before. In terms of performance of resulting rule bases, UR-IFRC(FCM) achieves 4 significantly statistical better results by employing FCM partitioned fuzzy sets, with average accuracy improved over 1.1% overall. This is expected, as clustered fuzzy sets better reflect underlying data distribution compared to heuristic uniformly divided partition. This could potentially provide higher matching degrees when mapping data-driven generated crisp intervals into predefined fuzzy sets, lessening subsequent tuning steps with GA. This is also reflected in the complexity of generated rule bases using FCM partitioned fuzzy sets, where average number of conditions

needed for each rule across all data sets is significantly decreased from 3.8 to 2.53, and rule base structure also has shrunk using over 2 rules fewer, generating more concise rule bases, and hence, more compact knowledge. That is, by using data-driven partitioned fuzzy sets that more reflect the distribution of underlying data, it is likely to generate more accurate and compact rule bases.

Table 6.1: Performance comparison with FCM-partitioned fuzzy sets against that of uniform partition

	UR-IFRC			**UR-IFRC(FCM)**		
Data Sets	**Accu**	**Rul**	**Cond**	**Accu**	**Rul**	**Cond**
blood	77.82 ± 1.11	7.35	2.25	**78.81 ± 0.86** (*)	7.31	1.84
column2C	81.00 ± 2.07	14.80	3.14	**83.57 ± 1.40** (*)	14.58	2.42
ionosphere	**85.58 ± 1.84**	24.01	6.77	84.93 ± 2.46 (=)	16.47	2.88
bupa	**64.86 ± 2.09**	27.84	3.68	64.53 ± 2.77 (=)	28.92	3.30
mammographic	78.46 ± 1.09	5.46	2.79	**82.45 ± 0.89** (*)	4.21	1.72
thyroid	94.29 ± 0.85	14.73	2.60	**94.58 ± 1.12** (=)	11.39	2.29
parkinsons	87.02 ± 1.34	20.35	5.07	**89.18 ± 2.19** (*)	17.01	2.56
diabetes	**75.27 ± 0.69**	24.01	4.08	75.11 ± 1.05 (=)	20.09	3.26
average	80.538	17.32	3.80	**81.645**	15.00	2.53

6.3 Journal Ranking with Interpretable Fuzzy Rules

Although much debate has surrounded the issue of subjective ranking of academic journals, to verify the results of this experiment, the professional report on Ranked Journal List (RJL) provided by ERA 2010 [151] from human experts is employed as the ground truth. Each journal in RJL has a rank in the (ordered) domain *Ranks* = {C, B, A, A*}, where rank A* indicates the top category of journals in a certain research area. When combining the selected indictor scores from JCR and the ranked result from RJL, only those journals that are both indexed by JCR and ranked in RJL are considered as valid experimental data for fair comparison. The resultant data contains 320 journals in total including 44 ranked as A*, 101 as A, 108 as B, and 67 as C.

Given of no direct expertise accessible in this work, FCM-partitioned fuzzy sets that have been proven effective as demonstrated in Section 6.2 are employed here to represent domain knowledge. This is clearly more intuitive than equally divided

uniform partition in that impact indicators values are normally highly skewed. In particular, each impact indicator is heuristically partitioned into 5 linguistically labelled fuzzy sets. Figure 6.1 shows the 5 linguistically labelled fuzzy sets that domain experts use to generate the FCM result on a selective set of journals in Computer Science which are evaluated by the JCR of 2010. Figure 6.2 shows the adjusted result by applying the filter process to ensure that the membership of a data point to a cluster is monotonically decreasing with its distance to the cluster centre. It is important to recall that though being data-driven, FCM-partitioned fuzzy sets are obtained homogeneously for each impact indicator and remain fixed throughout the modelling and inference processes. Note that, in practice, the required labelling of generated fuzzy sets may be accomplished by consulting human experts in the field, but in this work, they are assigned on the basis of common sense, due to the unavailability of such direct expertise.

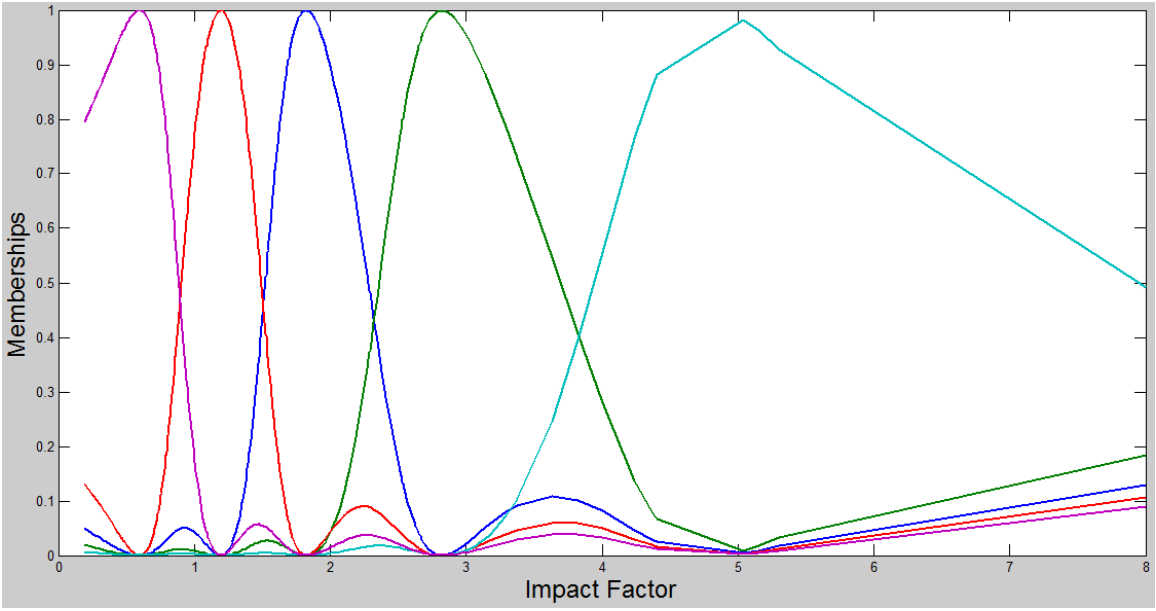

Figure 6.1: FCM-partitioned fuzzy sets on journal impact factors

To run the proposed approach without losing generality, suppose that C4.5 is used to generate the required initial set of basic crisp rules, with the resultant crisp rules given in Figure 6.3. Note that the exact same variables may be utilised multiple times within a single crisp rule (e.g., AI is used twice in rule C_1). Given these crisp rules, a set of descriptive fuzzy rules is generated using the proposed approach, as shown in Figure 6.4. Instead of using numerical intervals (which are hard to interpret) to describe the impact indicators, this set of fuzzy rules use the linguistic labels and are

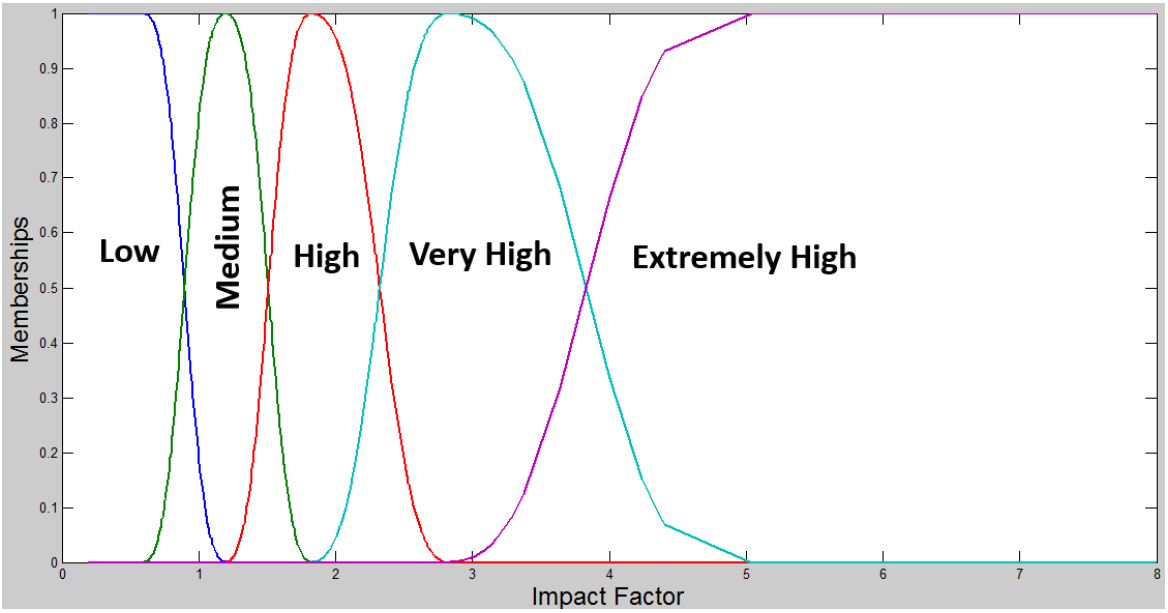

Figure 6.2: Fuzzy sets used in journal ranking rules

readily readable. Interestingly, the number of resultant interpretable fuzzy rules is fewer than that of the original crisp rules, with individual fuzzy rules being generally much more compact also. This further enhances the structural interpretability of the learned classifier, while reducing run-time computational cost. Furthermore, such interpretability is not obtained by sacrificing the performance of the original crisp rules as systematically demonstrated with the experiments in Chapter 5.

By further examining the generated fuzzy rule base it can be seen that 5IF and AI are very heavily used in the resultant rules. However, this is not surprising since both of these indicators are selected by C4.5 as the ones with most discriminating capability, as shown in the first two conditions in all eight original crisp rules. Importantly, these two indicators are also the most highly correlated features to the ranking in RJL, which is further confirmed by the findings of [144]. Specifically, when both 5IF and AI are low, the covered journal is likely to be ranked in the bottom category. Whereas when AI is very high, the resultant journal ranks tend to be in the top category. For journals with a medium IF, but high in terms of 5IF, they are still likely to be ranked in category A. Whereas for journals with a medium 5IF, it is possible for them to be classified into category B. Generally speaking, despite the learned fuzzy model being compact and concise, the ranking outcomes based on the given impact indicators seem to be consistent with the RJL ranks.

Figure 6.3: Crisp rule base generated by C4.5

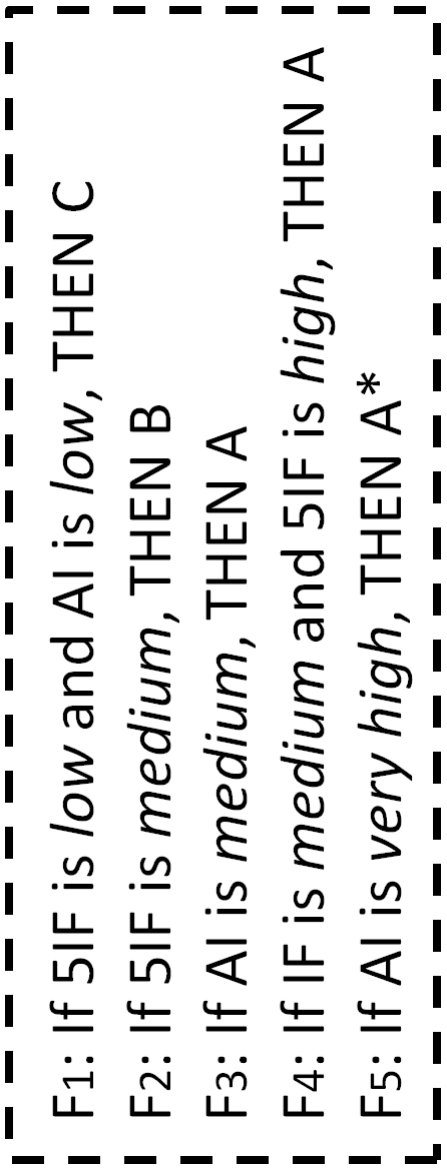

Figure 6.4: Generated fuzzy rule base

Running C4.5 on fixed intervals generated by homogeneously discretising numerical attributes beforehand can help regain such interpretability, but this easily results in performance loss. However, this works for fuzzy rules, which consists of fuzzy sets that permit gradual assessment of the membership of elements in the set, thereby resulting in more flexible decision boundaries (compared to those also using fixed crisp intervals). Note that semantics-based interpretability is not automatically obtained by just using fuzzy rules. Consider the following fuzzy rule generated by fuzzifying C_1 with the popular benchmark fuzzy classifier FURIA [66]:

$$\begin{aligned} &\text{If 5IF is } [2.467, 2.475, +\infty, +\infty] \text{ and AI is} \\ &\qquad [-\infty, -\infty, 1.247, 1.258], \text{ Then A with CF} = 0.43 \end{aligned} \tag{6.5}$$

where both generated fuzzy sets are of trapezoid MFs open to one side. The trapezoid MF obtained for each attribute is obtained by searching for the support bound with best purity, where the existing crisp interval fits the lower or upper bound of the core. Each generated fuzzy set that is purely searched with regard to rule performance may vary greatly, without fixed and uniformly consistent knowledge to refer to. No consistent linguistic labels can therefore be attached to these generated fuzzy sets, making sense only within individual fuzzy rules. Hence, the semantic interpretability may be largely lost without global semantics. Needless to say, the rule base transparency is even further deteriorated by utilising certainty factors [77] as rule weights with weighted majority vote as inference methods.

6.4 Summary

Due to the significance and popularity of journal ranking in research assessment, this chapter has first given a brief introduction to journal impact indicators and the potential problems of journal ranking typically done by human experts (mainly being financial and time consuming). As an initial attempt to have a computer-based solution, this chapter has collected statistics of Computer Science journals from Web of Science and generated a set of interpretable fuzzy rules with approach from Chapter 5. Empirical partitioning method via the use of FCM which has shown to outperform equal partition via the use of generating fuzzy sets that form fixed quantity space for this modelling task. The generated fuzzy rules are highly readable and can help users understand the relationship between journal impact indicators and their ranks.

Chapter 7

Reliability-guided Fuzzy Classifier Ensemble

In human society, when there are important decisions to make, having a committee of experts with different perspectives to vote against a certain motion offers an effective way of decision-making, reducing if not completely avoiding any bias which may otherwise be caused by a single expert. The development of classifier ensembles has been motivated by this observation. The main idea is to weight several individual classifiers, and combine them in order to obtain a classifier that outperforms every one of them [130]. Different classifiers usually make different predictions on certain samples, caused by their diverse internal modelling structures and parameters. Combining such classifiers has become the natural way of trying to increase the overall classification accuracy and hence, a focus of attention in current research [39].

A typical approach to building classifier ensembles involves constructing a group of classifiers with diverse training backgrounds [18], [64], before their decisions get integrated to produce the final classification. Instead of adopting a simple majority voting-based aggregation [91], ensemble stacking [44] has also been developed that employ meta-level learners to combine the outputs of the base classifiers. As each ensemble member may be trained using a subset of training samples, this may also reduce the computational complexity that arises when a single classification algorithm is applied to a very large dataset, while supporting potential parallel implementation.

Classifier ensemble selection (CES), i.e., an intermediate step between ensemble construction and decision aggregation has drawn significant attention [154]. It selects ensemble members from a pre-constructed pool of base classifiers, to form a reduced subset of classifiers that can still deliver the same classification results as the original full set of potential ensemble members [39]. Efficiency is one of the obvious gains from CES. Having a reduced number of base classifiers helps to reduce run-time overheads; having fewer models also implies relaxed memory and storage requirements. In addition, removing unreliable ensemble members decreases the adverse effect of a false or biased judgment within the emerging ensemble, while increasing potential ensemble classification performance. Existing approaches include techniques that employ clustering [55] to discover groups of models that share similar predictions and subsequently prune each cluster separately, and those that use reinforcement learning [118] and fuzzy-rough-based feature selection [85, 39] to achieve removal of redundant base classifiers.

The intuitive idea of data reliability [16] has recently been incorporated into the main stream of research on ordered weighted averaging (OWA) operators [166, 143, 148, 147]. In the process of combining multiple arguments, a precaution worth noting is that unduly high/low or abnormal aggregated values may result from a false or biased judgement. In such cases, a typical OWA operator may suffer significantly from assigning the highest priority to just either the highest or the lowest value. To address this problem, the reliability-oriented approach models the aggregation behaviour in accordance with the underlying characteristics of the data being aggregated. Different from the original dependent OWA (DOWA) operator [165], where a normal distribution of argument values is presumed in order to determine their reliability degrees, this approach assesses the significance of possible trends that may emerge from a local structure involving a set of nearest neighbours which are tightly clustered together [16].

This chapter further develops the idea of data reliability with application to classifier ensemble. In particular, opinions from ensemble members of low reliabilities should be naturally awarded low weights on their potential contribution to the final decision making process. Instead of simply projecting decision labels of each classifier member onto the training instances [39], an M-nary representation is proposed to retain complete decision information from the ensemble member [39]. This is of particular significance for building classifier ensembles using base classifiers that work

on fuzzy rules [66, 26], where instances match against rules with different classes to various degrees. Reliability measure guided by nearest-neighbour-based assessment is then carried out for each ensemble member, such that ensemble members of a low reliability are removed. The reliabilities of the remaining ensemble members are perceived as a stress function, from which argument-dependent weights can be generated, leading to the final aggregated classification decision.

The reminder of this chapter is organised as follows. Section 7.1 introduces the background of OWA aggregation and the nearest-neighbour-based reliability measure. Section 7.2 describes the proposed classifier ensemble selection by incorporating this novel reliability measure. Section 7.3 presents and discusses experimental results, and Section 7.4 concludes the paper and outlines ideas for further development.

7.1 Preliminaries

7.1.1 OWA Aggregation

When dealing with real-world problems, the opinions of different experts are usually aggregated in order to provide more robust solutions. Similarly, numeric measures of certain properties are also typically aggregated when addressing a given problem [139, 40]. Apart from the classical aggregation operators (such as average, maximum and minimum), another interesting and more general type of aggregation operator is the family of OWA operators [166]. OWA is a parameterised operator based on the ordering of extraneous variables to which it is applied. The fundamental aspect of this family of operators is the reordering step in which the extraneous variables are rearranged in descending order, with their values subsequently integrated into a single aggregated one.

Formally, a mapping $A_{\text{owa}} : \mathbb{R}^v \to \mathbb{R}$ is called an OWA operator if

$$A_{\text{owa}}(a_1, \cdots, a_v) = \sum_{i=1}^{v} w_i a_{\pi(i)} \tag{7.1}$$

where $a_{\pi(i)}$ is a permutation of the values of a_i, which satisfies that $a_{\pi(i)}$ is the i-th largest value of a_i, and $w_i \in [0,1]$ is a collection of weights that jointly satisfy $\sum_i w_i = 1$, $i = 1, \cdots, v, v > 1$. For simplicity, let $W = (w_1, \cdots, w_v)^T$.

Different specifications of the weighting vector W lead to different aggregation results. The ordering of extraneous variables gives OWA a nonlinear feature. Three special cases of the OWA operator are the classical *mean*, *max* and *min*. The *mean* operator results by setting $w_i = 1/v$, the *max* by $w_1 = 1$ and $w_i = 0$ for $i \neq 1$, and the *min* by $w_v = 1$ and $w_i = 0$ for $i \neq v$. These weighting vectors are denoted as W_{mean}, W_{max} and W_{min} respectively in the remainder of this paper. Obviously, an important feature of the OWA operator is that it is a weighted average operator which satisfies

$$\min\{a_1, \cdots, a_v\} \leq \sum_{i=1}^{v} w_i a_{\pi(i)} \leq \max\{a_1, \cdots, a_v\} \tag{7.2}$$

Such an operator provides aggregation between the maximum and the minimum of the arguments. This boundedness implies that it is idempotent; that is, if all $a_i = a$ then $A(a_1, \cdots, a_v) = a$.

A measure which is commonly employed to reflect the overall behaviour of an OWA operator is orness [43]. It captures the design intention of whether an aggregation operator behaves similarly to the interpretation of logical conjunction (influenced by smaller inputs) or that of disjunction (influenced by larger inputs). In particular, an orness measure of an OWA operator with the weighting vector W is defined by [166]

$$\text{orness}(W) = \frac{1}{v-1} \sum_{i=1}^{v} ((v-i)w_i). \tag{7.3}$$

The higher the orness value, the more similar the aggregated result is to that of disjunction. Also, it can be calculated that $\text{orness}(W_{mean}) = 0.5$, $\text{orness}(W_{max}) = 1$ and $\text{orness}(W_{min}) = 0$.

7.1.2 Nearest Neighbour (NN) Based Reliability Measure

In combining multiple arguments using pre-defined weighting vectors in OWA, the weight vector W is normally assumed to be argument-independent as the weights are not necessarily related to the extraneous variables to which they are applied. Therefore, the use of unduly high or low weights should be avoided. Otherwise, a typical OWA operator may suffer from giving the highest priority to outlier variable

values [17], leading to the generation of false or biased judgments when the operator is in action.

To achieve more reliable outcomes, data-oriented operators such as the DOWA [165] utilise centralised data structures to generate reliable weights for aggregating information. An efficient nearest-neighbour-based method for the assessment of data reliability (or sometimes referred to as relevance) has been proposed [16] in which the local data structure that represents a strong agreement of consensus on information can be explored. This reliability measure is effective to discriminate the weights of different input arguments, with the previously adopted closest cluster replaced by a set of K nearest neighbours.

More formally, given a collection of data arguments $A = \{a_1, ..., a_v\}$, let $N_{a_i}^K$ denote a set of K nearest neighbours of the argument a_i, where $N_{a_i}^K \subset A$ and $\forall n_j \in N_{a_i}^K, n_j \neq a_i, j = 1, ..., K$. The reliability measure $R_{a_i}^K \in [0, 1], i = 1, ..., v$ can be computed such that:

$$R_{a_i}^K = 1 - \frac{D_{a_i}^K}{D_{\max}} \tag{7.4}$$

$$D_{a_i}^K = \frac{1}{K} \sum_{\forall n_j \in N_{a_i}^K} |a_i - n_j| \tag{7.5}$$

where $D_{max} = \max_{a_p, a_q \in A, a_p \neq a_q} |a_p - a_q|$

The nearest-neighbour-based method has two main advantages over conventional techniques. First, the otherwise required high computational cost for cluster-based measuring of data reliability is reduced, decreasing both time and space complexity from $O(L^3)$ to $O(L^2)$ and $O(L^2)$ to $O(L)$, respectively. Second, the nature of the distributed approach inherent in clustering is reinforced so that arguments being very far away from the global centre can be considered reliable if they are close to members of their local neighbour sets. Figure 7.1 illustrates the nearest-neighbour-based approach, in which arguments (a_1 and a_2) that are far away from the global centre are considered reliable if they are close to members of their local neighbour sets (N_{a_1} and N_{a_2}, respectively).

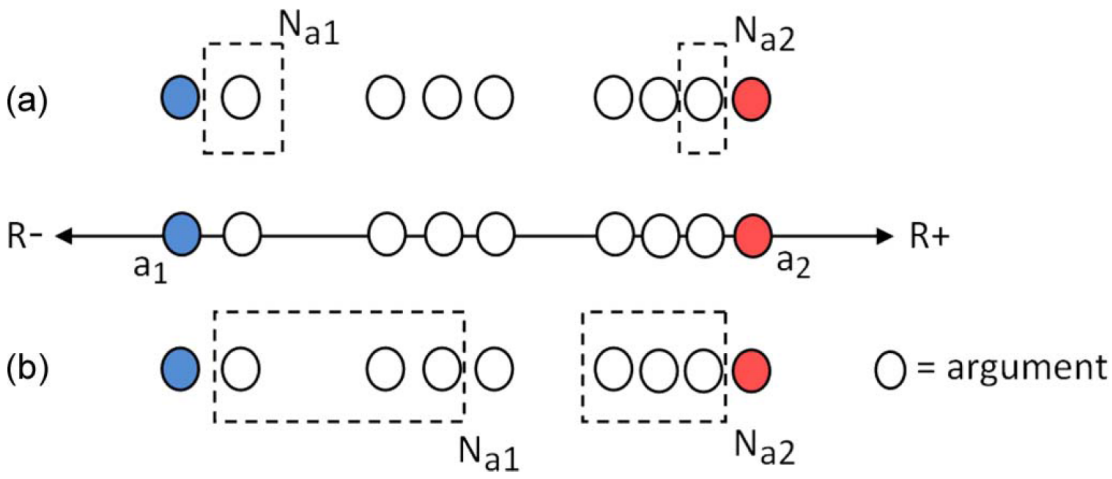

Figure 7.1: Sets of local neighbours N_{a_1} and N_{a_2}, (a) $K = 1$ and (b) $K = 3$

7.2 Reliability-guided Fuzzy Classifier Ensemble

7.2.1 Overview

The overall process of reliability-guided classifier ensemble is outlined in the flow chart as shown in Figure 7.2, with each of the four main components described in the subsequent subsections.

7.2.2 Base Classifier Pool Generation

Forming a set of diverse base classifiers is the first step in producing a working classifier ensemble. Any preferred model-building strategies may be used to build the base classifiers. As an initial implementation to test the proposed approach, only the bagging strategy [18] is adopted here. Bagging randomly selects different subsets of training samples in order to build diverse classifiers. Differences in the training data present extra or missing information for different classifiers, thereby resulting in models with different classification borders.

The bagging strategy is capable of introducing quality diversities in classification model generation, even if just one single base classification method is employed. Having taken notice of this, only the state of the art fuzzy rule-based classifier FURIA

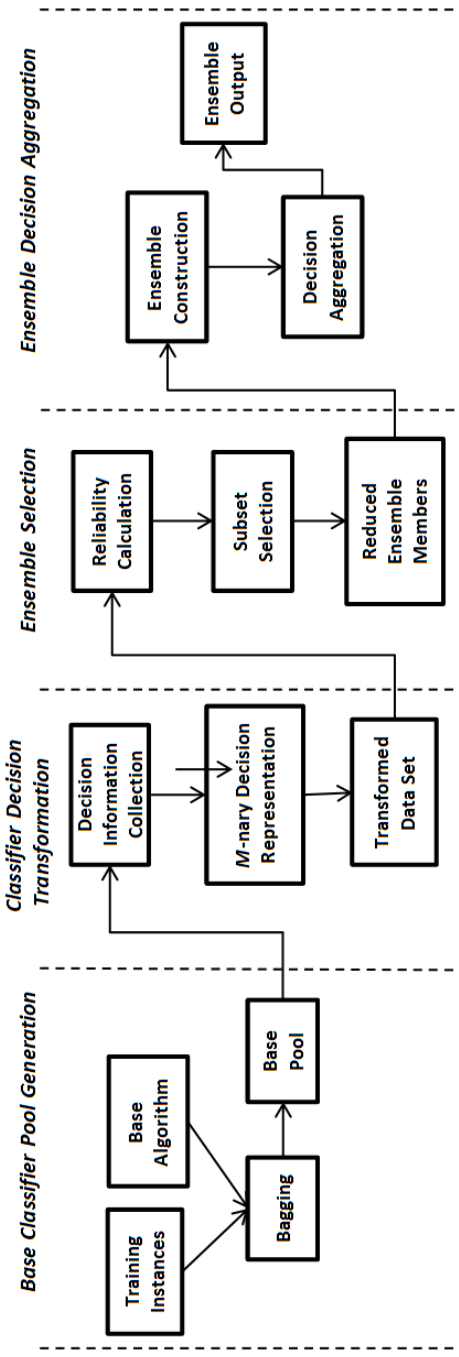

Figure 7.2: Flow chart of reliability-guided classifier ensemble

[66] is used here to implement the individual base classifiers. FURIA is an extension of the well-known crisp rule-based learner RIPPER [31]. Its working process can be summarised as follows: Unordered crisp rule sets are first obtained by learning rules initialised by RIPPER. Each generated crisp rule is then fuzzified by keeping the same structure, but with crisp intervals replaced by fuzzy intervals. The optimal bounds over the classification are greedily learned, with the learning process guided by rule purity. In order to tackle instances that cannot be covered by any existing rule, a rule stretching technique is subsequently applied.

7.2.3 Classifier Decision Transformation

Once the base classifiers are built, their decisions on the training instances can be gathered. A new artificially transformed dataset can be constructed, with each column representing a single base classifier and each row corresponding to a certain training instance [39]. Thus, each cell of the transformed dataset stores the value D_{ij}, representing the decision of the base classifier $C_j, j = 1, 2, .., N_C$ with regard to the instance $I_i, i = 1, 2, ..., N_I$, where N_C denotes the total number of base classifiers generated, and N_I is the total number of the given training instances. Such an artificial dataset can be regarded as a decision matrix as shown in Table 7.1.

Table 7.1: Transformed decision matrix

	C_1	...	C_j	...	C_{N_C}
I_1	D_{11}	...	D_{1j}	...	D_{1N_C}
⋮	⋮	⋮	⋮	⋮	⋮
I_i	D_{i1}	...	D_{ij}	...	D_{iN_C}
⋮	⋮	⋮	⋮	⋮	⋮
I_{N_I}	D_{N_I1}	...	D_{N_Ij}	...	$D_{N_IN_C}$

Following the above approach, two issues may arise. The first is that it may lead to information loss, especially for cases where fuzzy rule-based classifiers are used as a certain given instance is likely to match multiple rules involving different consequents, albeit to a different degree. Simply adopting the one with the maximum degree obviously removes potential decision information regarding the other classes. Similar situations may also occur in traditional classifiers (where an instance is classified as the class with the maximum likelihood). The second issue is that ideally,

the class labels representing different concepts should be completely independent of each other. Namely, the distance between any pair of distinct classes should be the same. Projecting class labels onto sequential numbers may lead to an unreasonable situation, where the distances between different classes may be different, potentially affecting the eventual classification results when ordered aggregation is utilised.

In order to tackle these issues, an M-ary representation for class labels is proposed, where M represents the number of classes in a given classification problem. The idea is to exploit an M-dimensional coordinate system with M planes perpendicular with one another, such that each class label can be projected onto a coordinate axis with full decision membership being 1.0. As each coordinate axis is perpendicular to each other, the distance between any pair of class labels is obviously the same.

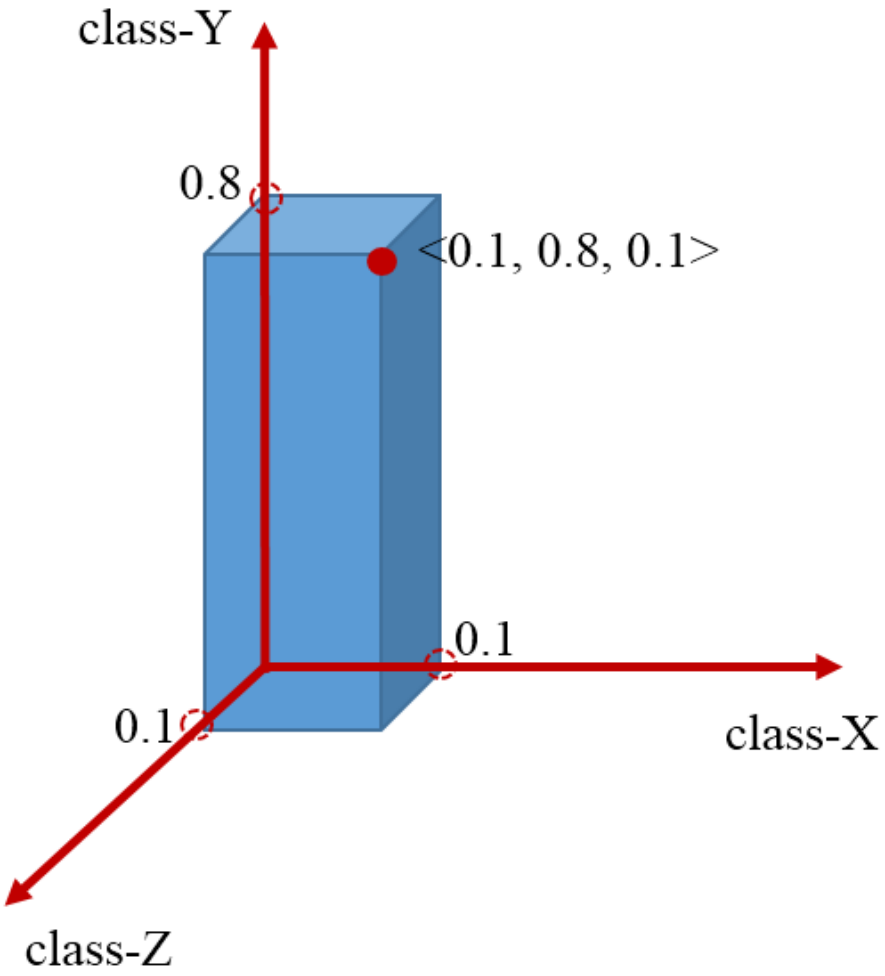

Figure 7.3: Example for M-ary representation

Let $t_M^{ij}, i = 1,2,...,N_I, j = 1,2,..,N_C$ be an M-ary tuple to represent complete decision information of classifier C_j with regard to instance I_i, for an M-class classification problem, such that $t_M^{ij} = \langle \mu_1^{ij}, ..., \mu_m^{ij}, ..., \mu_M^{ij} \rangle$, where μ_m^{ij} is the matching degree or probability density with regard to class m. Consider an example, where a fuzzy classifier that matches an instance against class-X with a degree of 0.1, class-Y with 0.8, and class-Z with 0.1. This can be represented using the M-ary representation as $\langle 0.1, 0.8, 0.1 \rangle$ without information loss as shown in Figure 7.3. A transformed M-nary-based decision matrix can therefore be constructed as shown in Table 7.2.

Table 7.2: Transformed decision matrix using M-ary representation

	C_1	...	C_j	...	C_{N_C}
I_1	$\langle \mu_1^{11}, ..., \mu_m^{11}, ..., \mu_M^{11} \rangle$	...	$\langle \mu_1^{1j}, ..., \mu_m^{1j}, ..., \mu_M^{1j} \rangle$	...	$\langle \mu_1^{1N_C}, ..., \mu_m^{1N_C}, ..., \mu_M^{1N_C} \rangle$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I_i	$\langle \mu_1^{i1}, ..., \mu_m^{i1}, ..., \mu_M^{i1} \rangle$	...	$\langle \mu_1^{ij}, ..., \mu_m^{ij}, ..., \mu_M^{ij} \rangle$	...	$\langle \mu_1^{iN_C}, ..., \mu_m^{iN_C}, ..., \mu_M^{iN_C} \rangle$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I_{N_I}	$\langle \mu_1^{N_I 1}, ..., \mu_m^{N_I 1}, ..., \mu_M^{N_I 1} \rangle$	...	$\langle \mu_1^{N_I j}, ..., \mu_m^{N_I j}, ..., \mu_M^{ij} \rangle$	...	$\langle \mu_1^{N_I N_C}, ..., \mu_m^{N_I N_C}, ..., \mu_M^{N_I N_C} \rangle$

Consider a binary classification problem as an example, where a certain fuzzy classification method is used to construct the base classifiers. Suppose that an instance matches against the ensemble member C_1 with $\langle 0.51, 0.49 \rangle$, C_2 with $\langle 0.49, 0.51 \rangle$, and C_3 with $\langle 1.0, 0.0 \rangle$. Simply projecting decision labels for each ensemble member would lead to a situation that the instance is classified by both C_1 and C_3 into the first class, and the second class by C_2. For the decision information of C_1 or C_2 with regard to this instance, the matching degree for each class does not dominate one or the other. This will lead to the situation where different labels are provided to the same instance. However, the decision information of C_1 appears to be more similar to that of C_2 than to C_3, despite the fact that both C_1 and C_3 provide the same decision labels. Such embedded differentiating information that is captured by the fuzzy classifiers is therefore lost if only decision labels are utilised. This further supports the proposal of employing the M-ary representation.

7.2.4 NN Based Reliability for Ensemble Member Selection

Having obtained the M-ary-based decision matrix, where complete decision information is presented for each instance I_i on classifier C_j, a simple heuristic method can

be used to facilitate classifier ensemble selection, so that unreliable members can be identified and removed. The computation process for this involves the following three steps:

1. Calculate the reliability measure $R^K_{t^{ij}_M}$ of each M-ary tuple t^{ij}_M with regard to its K nearest neighbour, according to Eqns. 7.4 and 7.5. The distance between the tuple t^{ij}_M and its neighbour $t^{ij'}_M$ is computed by

$$d(t^{ij}_M, t^{ij'}_M) = \sqrt{\sum_{m=1}^{M} (\mu^{ij}_m - \mu^{ij'}_m)^2} \tag{7.6}$$

   where μ^{ij}_m is the matching degree with regard to class m using the M-ary representation.

2. Compute the accumulated reliability value CR_j for each classifier ensemble member $C_j, j = 1, 2, .., N_C$, by summing its reliability measures $R^K_{t^{ij}_M}$ with regard to each of the instance $I_i, i = 1, 2, ..., N_I$, such that

$$CR_j = \sum_{i=1}^{N_I} R^K_{t^{ij}_M} \tag{7.7}$$

3. Rank each classifier ensemble member based on their accumulated reliability degrees. Intuitively, the higher the reliability is, the more convincible the ensemble member becomes. Similar to the work of [39, 139], a simple threshold-based selection method is adopted, such that the ensemble member $C_j, j = 1, 2, .., N_C$ is only included in the final ensemble list if its corresponding reliability CR_j exceeds a given threshold. To avoid being subjectively defined, the threshold is empirically set using the average reliability CR_{average} of all ensemble members as:

$$CR_{\text{average}} = \frac{1}{N_C} \sum_{j=1}^{N_C} CR_j \tag{7.8}$$

7.2.5 Ensemble Decision Aggregation

Once reliable ensemble members are obtained by removing unreliable ones from the complete ensemble panel, their decision results can be aggregated to form the final ensemble decision output. Suppose that the reliable ensemble members

$C_j, j = 1, 2, .., N_f$ are retained after filtering, where N_f is the number of remaining ensemble members. Given the decision class $m \in 1, 2, .., M$, with M being the number of decision classes, classifier decisions can be viewed as a matrix of weighted probability distributions $\{\delta_j P_{jm}\}$, such that $\{P_{jm}\}$ indicates the classification from the classifier C_j for decision m and δ_j is a weight associated with C_j, indicating the strength associated with the classification regarding C_j. Here, δ_j is generated by taking the corresponding reliability measure CR_j over the sum of all reliability values from the remaining ensemble members.

Summarising the above, the total weighted probability P_m for decision class m with regard to instance I_i is calculated as follows:

$$P_m = \sum_{j=1}^{N_f} \delta_j \mu_m^{ij} \tag{7.9}$$

where $\delta_j = \frac{CR_j}{\sum_{j=1}^{N_f} CR_j}$, and μ_m^{ij} is the matching degree of classifier C_j with regard to class m. The final aggregated decision assigned is the winning class that has the highest total weighted probability among all classes: $\arg\max_{m=1,2,..,M} P_m$.

7.3 Experimentation and Discussion

7.3.1 Experimental Setup

As indicate previously, the ensemble construction method adopted here is the bagging strategy, and the base classification mechanism is FURIA [66]. Stratified tenfold cross-validation (10-CV) is employed for result validation. In 10-CV, a given dataset is partitioned into ten subsets. Of the ten, nine subsets are used to perform a training fold, where the proposed approach is used to generate a fuzzy rule base, and the remaining single subset is retained as the testing data for assessing the learned classifier's performance. In the experimentation, 10-CV is performed ten times in order to lessen the impact of random factors; these 10×10 sets of evaluations are then averaged to produce each final experimental outcome reported below.

Table 7.3: Comparison against fuzzy rule-based classifiers using 10 × 10 cross-validation with respect to classification accuracy (%), where bold figures signify overall top results per dataset

Dataset	1NN-DOWA	Size	Base	Random	Full
glass	78.12 ± 1.24	26.54	69.15 ± 1.70 (*)	75.88 ± 1.69 (*)	**78.43 ± 1.22** (-)
ionosphere	90.28 ± 0.90	27.62	87.96 ± 1.25 (*)	89.98 ± 0.90 (-)	**90.61 ± 1.07** (-)
leaf	71.03 ± 0.98	25.94	60.06 ± 1.29 (*)	70.11 ± 1.39 (*)	**71.35 ± 0.95** (-)
libras	77.11 ± 1.28	25.78	60.75 ± 1.55 (*)	75.39 ± 1.36 (*)	**77.58 ± 1.52** (-)
olitos	79.25 ± 1.39	26.79	68.50 ± 3.23 (*)	77.92 ± 1.54 (*)	**79.83 ± 0.86** (-)
parkinsons	91.39 ± 1.08	28.09	88.80 ± 1.26 (*)	90.98 ± 0.89 (-)	**91.55 ± 1.04** (-)
sonar	84.26 ± 1.92	26.86	76.44 ± 2.77 (*)	82.94 ± 2.07 (*)	**84.96 ± 1.74** (-)
vehicle	**76.09 ± 0.82**	26.96	66.79 ± 1.06 (*)	76.06 ± 0.71 (-)	75.98 ± 0.77 (-)
yeast	61.51 ± 0.50	27.79	55.73 ± 0.50 (*)	60.93 ± 0.42 (*)	**61.74 ± 0.60** (-)

7.3.2 Results and Discussion

For simplicity, when presenting experimental results, only those with $K = 1$ as the number of nearest neighbours are exploited as shown in Table 7.3, where 1NN-DOWA is the reduced classifier ensemble based on the use of 1NN-DOWA-based reliability measure [16]. It will be shown later that the proposed approach is almost independent of the parameter setting for K. For comparison purpose, results of: a) the base algorithm itself, b) full ensemble classifier pool with size being 50, and c) randomly constructed ensembles are also presented. Pairwise t-tests ($p = 0.05$) are run to gauge results in terms of the significance of statistical differences between different classifiers. Those results that are significantly better, worse or of no difference are marked with "(v)", "$(*)$", or "$(-)$", respectively, in comparison to the achieved accuracy of the 1NN-DOWA-based classifier ensemble.

To demonstrate the proposed approach at work, experiments are performed on 9 real-valued benchmark data sets, the characteristics of which can be found in Appendix B. The results show that the 1NN-DOWA-based system achieves significantly improved performance over the base classifiers statistically, across all employed datasets. Compared to randomly formed ensembles, the proposed approach also achieves better accuracy for all employed datasets, with 6 of which being statistically better. It is interesting to notice that 5 out of 6 datasets with statistically significant improvement using the proposed approach are those involving more than 3 classes,

e.g., leaf (30), libras (15), yeast (10), glass (7), olitos (4). Thus, this work has an empirical appeal to multi-class problems. One possible explanation for this is that the number of classes for a given training dataset may have a direct impact upon the proposed M-ary representation. The more classes there are, the richer the decision information could potentially be represented. The artificially transformed datasets can be more complex than those originated from the datasets with only 2 or 3 classes. Therefore, they successfully help better discriminate the original ensemble members that possess different degrees of reliability.

Comparison is further made with regard to full ensembles. Although the full ensemble system achieves 8 best results out of 9 datasets, the reduced ensembles maintain very similar classification accuracies – the accuracy of reduced ensembles are statistically equivalent to that of the full ensemble for every dataset. Whereas the size of reduced ensembles has significantly shrunk by about half of its original size (i.e., 50), becoming computationally much more manageable. This reduction rate in the size for each of the reduced is of course expected due to the use of the heuristic threshold, where ensemble members are discarded if their reliabilities are below the average. These experimental results have demonstrated that removing potentially unreliable base classifiers with the proposed approach can significantly reduce the size of a classifier ensemble, whilst maintaining classification accuracy, making the ensemble more efficient.

Given that the underlying reliability measure is parameterised by a user-defined K (the number of nearest neighbours), experiments are further conducted to reveal and reflect the relationship between the improvement of nearest neighbour granularities and the performance of the resultant classifier ensembles. Figure 7.4 depicts the performance variation of a group of nearest neighbours with 10 different sizes, where the x-axis describes the changes of K and y-axis shows the corresponding performance in terms of accuracy. In general, the connected curve for each dataset approximates a straight line without much oscillation, regardless of the number of decision classes. This has demonstrated that performance of the selected ensemble using KNN-DOWA is little affected by the number of nearest neighbours, being almost independent of the setting of parameter K. This shows the robustness of the proposed approach.

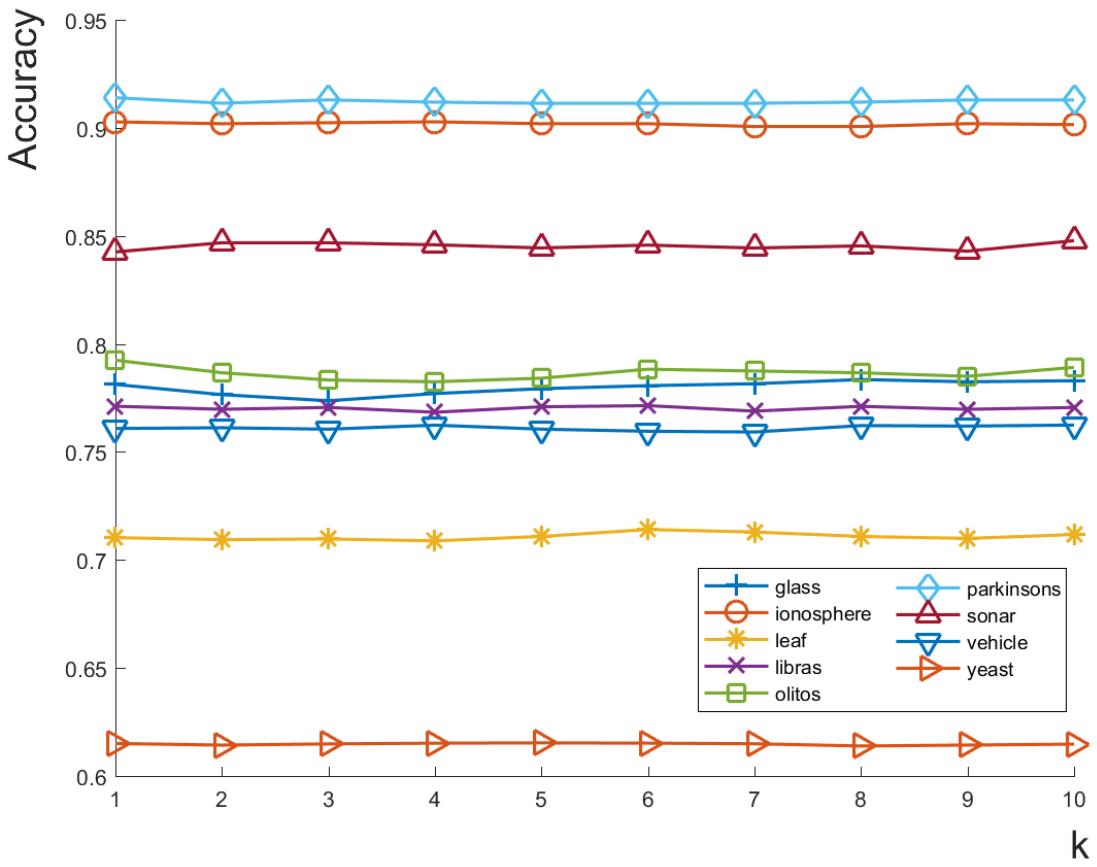

Figure 7.4: Performance variation in relation to parameter K

7.4 Summary

This chapter has proposed a new classifier ensemble selection approach based on the measure of nearest-neighbour-based reliability. To maintain complete decision information from each ensemble member, it has also introduced an M-ary representation that projects decision labels into artificially transformed datasets. In this approach, reliability guided by nearest-neighbour-based assessment is measured for each ensemble member, and ensemble members that are with low reliabilities (below average reliabilities of all ensemble members) are removed. Reliabilities of remaining ensemble members are perceived as a stress function, from which argument-dependent weights are generated for final aggregated classification.

Experimental results have demonstrated that removing potentially unreliable base classifiers can significantly reduce the size of a classifier ensemble, whilst maintaining

classification accuracy, making the ensemble system more efficient. It has also shown that the nearest-neighbour-based reliability measure is robust to the setting of the number of nearest neighbours as the performance of the resultant ensembles is not sensitive to it.

Chapter 8

Discussion and Conclusion

THIS chapter presents a summary of the research as detailed in the preceding chapters. Having introduced the theoretical basis for fuzzy rule induction and reviewed approaches that directly learn interpretable fuzzy rules with fixed and predefined fuzzy sets, as well as evolutionary fuzzy systems that utilise the powerful evolutionary algorithms as problem-independent optimisation methods, this thesis has proposed a number of techniques that have achieved promising results compared to state-of-the-art algorithms. The proposed refinement to rule weights significantly improves the performance of a heuristically initialised rule base with a fixed quantity space. The induction of quantified fuzzy rules is able to learn a set of rules with continuous fuzzy quantifiers. The approach that utilises crisp-based learning classifiers transforms existing crisp rules into fuzzy rules with consulted expertise in terms of predefined fuzzy sets, resulting in highly transparent fuzzy rules, which is verified by applying them in the popular journal ranking problems. The introduced fuzzy classifier ensemble method further improves performance of an existing fuzzy ensemble by removing unreliable members, which releases space storage and speeds up computation. Whilst the work is promising, this chapter also points out some initial thoughts for further research given that there is much that could be improved.

8.1 Summary of Thesis

The theoretical foundation specifies the relationship between fuzzy sets, fuzzy rules, fuzzy logic and fuzzy inference, which are building blocks of a fuzzy rule-based system. As a traditional rule induction technique, fuzzy decision trees induce rules by recursively shrinking the feature space. Different from traditional decision tree techniques, observations simultaneously fire multiple paths, requiring the aid of fuzzy logic for inference. Fuzzy association rule mining induces fuzzy rules based on the fuzzified measures of the support and confidence framework while the mining strategy still follows either Apriori algorithm or FP-Growth. Owing to the powerful search capability of evolutionary algorithms and being problem independent, both GA and PSO are reviewed given that they are intensively utilised as optimisation technique for several methods in the thesis. In a nutshell, the introduction of the knowledge building blocks of fuzzy systems and the review of relevant literature in Chapter 2 lays the foundation for the subsequent theoretical development.

The approach proposed in Chapter 3 follows the first pre-specified research route, i.e., to use a certain weighting scheme to boost performance of an existing fuzzy rule base, where the use of fixed and predefined fuzzy sets is a must for semantic interpretability. In particular, the approach works by optimising weights that are attached at the rule level, such that the significance of existing rules could be adapted to change classification boundary. Systematic experimental results have demonstrated that the performance of a fuzzy rule-based classifier can be significantly improved with rule weight refinement implemented by PSO. The size of an initially built rule base may affect the performance of the proposed method, although optimisation of the initial fuzzy quality space will help reduce such influence. The approach is competitive to typical state-of-the-art learning classifiers even if only expertise in terms of fixed and predefined fuzzy sets is used to create the initial rule base.

An alternative weighting-based approach is proposed in Chapter 4 that can learn a set of rules with continuous fuzzy quantifiers, such that all fuzzy rules can be combined and evaluated simultaneously. The approach works for situations where the information dealt with is not equally important, better capturing the relative importance among antecedent attributes by fuzzy continuous quantifiers. Instead of using crisp weights with fuzzy terms, which may lead to confusion regarding the linguistic interpretation, the use of fuzzy quantifiers to modify the linguistic terms

helps build fuzzy systems in a more natural way and ensure the inferred results remain in consistent fuzzy representation. Experimental results show that the quantified fuzzy rules induced by this method help boost the classification performance compared to those generated without using fuzzy quantifiers. This has enriched the development of first research route by utilising an alternative weighting scheme to enhance performance of an existing fuzzy rule on the basis of fixed quantity space.

Chapter 5 starts with the second research route to generate interpretable fuzzy classification rules by utilising existing crisp rules. Given that each of the crisp rules points to the problem sub-spaces where desirable fuzzy rules potentially exist, a heuristic mapping procedure has been presented that converts each preliminary crisp rule into a set of interpretable fuzzy rules involving only the predefined fuzzy sets, ensuring semantic interpretability. A local rule selection procedure is then performed to obtain a compact subset of initially mapped fuzzy rules that jointly generalise the capability of the underlying crisp rule. A fine grain tuning of all selected subsets of fuzzy rules is finally carried out with a conventional GA, resulting in an accurate and interpretable fuzzy rule-based classifier with a simplified structure. Systematic experimental examinations of the proposed approach have been carried out, involving the use of two different crisp rule generation mechanisms for initialisation, in comparison with both alternative fuzzy learning classifiers and non-fuzzy-rule-based classifiers. The results have revealed the overall superiority of the proposed approach over the rest.

Apart from running proposed methods over benchmark data sets, Chapter 6 applies the proposed work in Chapter 5 into real-world scenario of academic journal ranking, due to its increasing significance and popularity. As an initial attempt, this chapter has collected statistics of Computer Science journals from Web of Science and generated a set of interpretable fuzzy rules with the approach from Chapter 5. Empirical partitioning method via the use of FCM has shown to outperform that via an equal partition in providing the required predefined fuzzy sets that reflect domain expertise. Of course, this is in support of this computer simulation-based analysis of journal ranking. Should there be expert-specified fuzzy sets for use, then they should be adapted to better ensure interpretability. The generated fuzzy rules are highly readable and can help users understand the relationship between journal impact indicators and their ranks.

Chapter 7 has proposed a new classifier ensemble approach based on the measure of nearest-neighbour-based reliability. To maintain complete decision information

for each ensemble member, it has also introduced an M-ary representation that projects decision labels into artificially transformed data sets. In this approach, reliability guided by nearest-neighbour-based assessment is measured for each ensemble member, and ensemble members with low reliabilities are then removed. Reliabilities of remaining ensemble members are perceived as a stress function, from which argument-dependent weights are generated for final aggregated classification. Experimental results have demonstrated that removing potentially unreliable base classifiers can significantly reduce the size of a classifier ensemble, whilst maintaining classification accuracy, making the ensemble system more efficient.

Owing to the use of fixed quantity space that comes from either domain expertise or static fuzzy set definitions, the resulting fuzzy rule base is likely to suffer from performance loss, especially when distribution of the underlying training instances does not follow the pre-specified and fixed fuzzy set definitions. In general, the two proposed approaches in Chapter 3 and 4 with different weighting schemes have achieved significant performance improvement over original rule bases without rule weights. Having been published in two academic conferences and one journal, they have achieved the research goals following the first research route. Different from previous two approaches, the approach in Chapter 5 induces fuzzy rules from a complete different angle by observing the necessity of omitting empty space in the search space to avoid curse of dimensionality. By utilising an alternative data-driven crisp rule-based learning mechanism, it makes possible to focus on only the areas covered by data points, giving a head start to learn a more scalable fuzzy classifier instead of considering the combinations of all input and output variables. Given that it has also been utilised to solve real-world scenario of academic journal ranking, the approach in Chapter 5 has achieved the goal of second and third research route, which is currently under review for journal publication. To conclude, the thesis has not only finished all pre-specified research aims, but also develops an classifier ensemble approach with the concept of data reliability, aiming to combine fuzzy systems with more state-of-the-art techniques, which has been published in an international conference.

8.2 Future Work

Although promising, much can be done to further improve the work presented so far in this thesis. The following addresses a number of interesting issues that may help

strength the current research.

8.2.1 On Induction of Weighted Fuzzy Rules

Currently, only the accuracy of a fuzzy learning classifier is considered as the criterion or fitness measure when evolving rule weights. However, as indicated in Section 3.3.3, the size of rule sets or equivalently the number of rule weights may affect the final result. Thus, the number of rules and hence the partition of the input quantity space need to be considered to possibly become part of the fitness function. The optimisation of PSO parameters also needs to be examined in order to reinforce the ability of the proposed method since the current implementation does not investigate such potential effects. Furthermore, instead of using PSO, it would be interesting to see whether the use of an alternative evolutionary computation mechanism may help develop better fuzzy learning classifiers, regarding both effectiveness and efficiency. It is also worth investing the efficacy of the proposed approach when initial rule base comes from alternative initialisation methods (e.g., clustering algorithms).

8.2.2 On Induction of Quantified Fuzzy Rules

The assumption that one rule is sufficient to adequately describe a class, which is used in the present implementation, may be naive, especially when applied to larger and more complex real-world problems. Further work is required to determine how many rules may be necessary to describe a class for a given type of problem, so as to initialise an appropriate number of PSO particle dimensions for each potential class.

Furthermore, it would be very interesting to combine the two different weighting schemes which have been proposed in Chapter 3 and 4, to produce a more generalised inversion of quantified fuzzy rules with weights at both individual attribute and rule level. This would create more degrees of freedom in fuzzy rule-based system modelling while still only employing fixed and predefined fuzzy sets.

8.2.3 On Induction of Fuzzy Rules with Preliminary Crisp Representation

In the present implementation, multiple modelling objectives are simply converted into a compound single objective using weights. However, it would be interesting

to investigate whether the problem could be directly tackled using multi-objective evolutionary algorithms [47], enabling different tradeoffs between the possibly competing objectives. Also, the optimisation is currently realised with a Genetic Algorithm which is satisfactory, but the underlying approach is more general and can be implemented with other techniques. Another piece of further research would therefore be to explore the possibility of replacing the GA with alternative population-based algorithms such as harmony search [40] and particle swam optimisation [163].

8.2.4 On Journal Ranking with Induced Fuzzy Rules

Current experiment on academic journal ranking using induced fuzzy rules only considers computer science category. It would be interesting to perform analysis on data sets collected across a range of various disciplines. Furthermore, it would also be worthwhile to talk to experts from the panel regarding the rationality of induced fuzzy rules, as well as the definitions of fuzzy sets for individual impact indicators which may commonly be accept by majority of the experts.

8.2.5 On Reliability-guided Fuzzy Classifier Ensemble

Although the heuristic threshold selection approach works well, it would be interesting to investigate the potential of developing relevant techniques so that reliable ensemble members could be selected in a more data-driven way. Furthermore, comparison with state of the art methods on ensemble selection while addressing real world problems remains as future research.

Appendix A

Publications Arising from the Thesis

A number of 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 both papers already published and one article submitted for review.

A.1 Journal Articles

1. **Tianhua Chen**, Changjing Shang, Pan Su and Qiang Shen, Inducing accurate and interpretable fuzzy rules from preliminary crisp representation. Under review for publication.

2. Pan Su, Changjing Shang, **Tianhua Chen** and Qiang Shen, Ordered weighted aggregation of fuzzy similarity relations and its application to detecting water treatment plant malfunction. Engineering Applications of Artificial Intelligence, 2017.

3. Pan Su, Changjing Shang, **Tianhua Chen** and Qiang Shen, Exploiting data reliability and fuzzy clustering for journal ranking. IEEE Transactions on Fuzzy Systems, 25(5):1306-1319, 2017.

4. **Tianhua Chen**, Qiang Shen, Pan Su and Changjing Shang, Fuzzy rule weight modification with particle swarm optimization. Soft Computing, 20.8 (2016): 2923-2937.

A.2 Conference Papers

5. **Tianhua Chen**, Pan Su, Changjing Shang and Qiang Shen, Reliability-guided fuzzy classifier ensemble. Proceedings of the 26th International Conference on Fuzzy Systems, 2017 **(IEEE CIS Outstanding Student Paper Travel Grant)**

6. Pan Su, Changjing Shang, Yitian Zhao, **Tianhua Chen** and Qiang Shen, Fuzzy rough feature selection based on OWA aggregation of fuzzy relations. Proceedings of the 26th International Conference on Fuzzy Systems, 2017

7. **Tianhua Chen**, Qiang Shen, Pan Su and Changjing Shang, Induction of quantified fuzzy rules with particle swarm optimisation. Proceedings of the 24th International Conference on Fuzzy Systems, 2015

8. Pan Su, **Tianhua Chen**, Changjing Shang and Qiang Shen, Nearest neighbour-guided induced OWA and its application to journal ranking. Proceedings of the 23th International Conference on Fuzzy Systems, 2014

9. **Tianhua Chen**, Qiang Shen, Pan Su and Changjing Shang, Refinement of fuzzy rule weights with particle swarm optimization. Proceedings of the 2014 UK Workshop on Computational Intelligence, 2014.

Appendix B

Data Sets Employed in the Thesis

The data sets employed in the thesis are benchmark data that are public available through the UCI machine learning repository [11] which have been drawn from real-world problem scenarios. Table B.1 provides a summary of the properties of these data sets.

Table B.1: Information of data sets used in the thesis

Data set	Attributes	Classes	Instances
appendicitis	7	2	106
banknote	4	2	1372
blood	4	2	748
breast-cancer	9	2	699
column-2C	6	2	310
column-3C	6	3	310
ecoli	7	8	336
glass	9	7	214
haberman	3	2	306
image (training)	19	7	210
ionosphere	33	2	230
iris	4	3	150
leaf	15	30	340
libras	91	15	360
liver-disorders	6	2	345
mammographic	5	2	961
new-thyroid	5	3	215
olitos	25	4	120
parkinsons	22	2	195
pima-diabetes	8	2	768
prnn-synth	2	2	250
seeds	7	3	210
sonar	60	2	208
vehicle	18	4	846
wdbc	30	2	569
yeast	8	10	1484

Appendix C

List of Acronyms

10-FCV 10-fold cross-validation

ARM Association rule mining

C4.5 Decision tree algorithm

C45-IFRC C4.5 initiliased interpretable fuzzy rule-based classifier

C45-FURIA C4.5 initiliased unordered fuzzy rule induction

EA Evolutionary algorithm

EFS Evolutionary-based fuzzy system

FRBCS Fuzzy rule-based classification system

FRBS Fuzzy rule-based system

FST Fuzzy set theory

FURIA An algorithm for unordered fuzzy rule induction

GA Genetic algorithm

IFRC Interpretable fuzzy rule-based classifier

KDD Knowledge discovery in databases

MF Membership function

OWA Ordered Weighted Averaging

PSO Particle Swarm Optimisation

PTTD Top down induction of fuzzy pattern tree

QSBA Fuzzy subsethood-based rule induction algorithm

QuickRules Hybrid fuzzy-rough rule induction and feature selection

UR Unordered Ripper algorithm

UR-IFRC UR initiliased interpretable fuzzy rule-based classifier

UR-FURIA UR initiliased unordered fuzzy rule induction

Bibliography

[1] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, no. 2. ACM, 1993, pp. 207–216.

[2] R. Agrawal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215, 1994, pp. 487–499.

[3] S. Agrawal, K. Panigrahi, and M. K. Tiwari, "Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 5, pp. 529–541, 2008.

[4] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2010.

[5] R. Alcalá, Y. Nojima, F. Herrera, and H. Ishibuchi, "Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions," *Soft Computing*, vol. 15, no. 12, pp. 2303–2318, 2011.

[6] J. Alcala-Fdez, R. Alcala, S. Gonzalez, Y. NOJIMA, and S. Garcia, "Evolutionary fuzzy rule-based methods for monotonic classification," *Fuzzy Systems, IEEE Transactions on*, 2017.

[7] J. Alcala-Fdez, R. Alcala, and F. Herrera, "A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning," *Fuzzy Systems, IEEE Transactions on*, vol. 19, no. 5, pp. 857–872, 2011.

[8] J. M. Alonso, C. Castiello, and C. Mencar, "Interpretability of fuzzy systems: Current research trends and prospects," in *Springer Handbook of Computational Intelligence*. Springer, 2015, pp. 219–237.

[9] J. M. Alonso and L. Magdalena, "Hilk++: an interpretability-guided fuzzy modeling methodology for learning readable and comprehensible fuzzy rule-based classifiers," *Soft Computing*, vol. 15, no. 10, pp. 1959–1980, 2011.

[10] M. Antonelli, P. Ducange, F. Marcelloni, and A. Segatori, "A novel associative classification model based on a fuzzy frequent pattern mining algorithm," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2086–2097, 2015.

[11] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[12] F. Bartolucci, V. Dardanoni, and F. Peracchi, "Ranking scientific journals via latent class models for polytomous item response data," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 178, no. 4, pp. 1025–1049, 2015.

[13] G. Beliakov and S. James, "Citation-based journal ranks: the use of fuzzy measures," *Fuzzy Sets and Systems*, vol. 167, no. 1, pp. 101–119, 2011.

[14] F. J. Berlanga, A. Rivera, M. J. del Jesús, and F. Herrera, "Gp-coach: Genetic programming-based learning of compact and accurate fuzzy rule-based classification systems for high-dimensional problems," *Information Sciences*, vol. 180, no. 8, pp. 1183–1200, 2010.

[15] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[16] T. Boongoen and Q. Shen, "Nearest-neighbor guided evaluation of data reliability and its applications," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 40, no. 6, pp. 1622–1633, 2010.

[17] ——, "Clus-dowa: A new dependent owa operator," in *Proceedings of the 17th International Conference on Fuzzy Systems*. IEEE, 2008, pp. 1057–1063.

[18] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[19] J. Casillas, O. Cordón, F. H. Triguero, and L. Magdalena, *Interpretability issues in fuzzy modeling*. Springer, 2013, vol. 128.

[20] J. Cendrowska, "Prism: An algorithm for inducing modular rules," *International Journal of Man-Machine Studies*, vol. 27, no. 4, pp. 349–370, 1987.

[21] B. Cetişli and A. Barkana, "Speeding up the scaled conjugate gradient algorithm and its application in neuro-fuzzy classifier training," *Soft Computing*, vol. 14, no. 4, pp. 365–378, 2010.

[22] B. Chandra and P. P. Varghese, "Fuzzy sliq decision tree algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 5, pp. 1294–1301, 2008.

[23] ——, "Fuzzifying gini index based decision trees," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8549–8559, 2009.

[24] T. Chen, Q. Shen, P. Su, and C. Shang, "Refinement of fuzzy rule weights with particle swarm optimisation," in *Computational Intelligence, 2014 UK Workshop on*. IEEE, 2014, pp. 1–7.

[25] ——, "Induction of quantified fuzzy rules with particle swarm optimisation," in *Fuzzy Systems, 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–7.

[26] ——, "Fuzzy rule weight modification with particle swarm optimisation," *Soft Computing*, vol. 20, no. 8, pp. 2923–2937, 2016.

[27] T. Chen, P. Su, C. Shang, and Q. Shen, "Reliability-guided fuzzy classifier ensemble," in *Fuzzy Systems, 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[28] Y.-C. Chen, N. R. Pal, and I.-F. Chung, "An integrated mechanism for feature selection and fuzzy rule extraction for classification," *Fuzzy Systems, IEEE Transactions on*, vol. 20, no. 4, pp. 683–698, 2012.

[29] Y.-l. Chen, T. Wang, B.-s. Wang, and Z.-j. Li, "A survey of fuzzy decision tree classifier," *Fuzzy Information and Engineering*, vol. 1, no. 2, pp. 149–159, 2009.

[30] Z. Chen and G. Chen, "Building an associative classifier based on fuzzy association rules," *International Journal of Computational Intelligence Systems*, vol. 1, no. 3, pp. 262–273, 2008.

[31] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the 12th International Conference on Machine Learning*, 1995, pp. 115–123.

[32] O. Cordón, "A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems," *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 894–913, 2011.

[33] O. Cordón, M. J. del Jesus, and F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning*, vol. 20, no. 1, pp. 21–45, 1999.

[34] O. Cordón, M. J. del Jesús, F. Herrera, and M. Lozano, "Mogul: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach," *International Journal of Intelligent Systems*, vol. 14, no. 11, pp. 1123–1153, 1999.

[35] O. Cordón and F. Herrera, "A three-stage evolutionary process for learning descriptive and approximate fuzzy-logic-controller knowledge bases from examples," *International Journal of Approximate Reasoning*, vol. 17, no. 4, pp. 369–407, 1997.

[36] E. Cox, "The fuzzy systems handbook (1994)," *Boston: AP Professional*.

[37] M. Delgado, D. Sánchez, and M. A. Vila, "Fuzzy cardinality based evaluation of quantified sentences," *International Journal of Approximate Reasoning*, vol. 23, no. 1, pp. 23–66, 2000.

[38] Z. Deng, Y. Jiang, K.-S. Choi, F.-L. Chung, and S. Wang, "Knowledge-leverage-based tsk fuzzy system modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1200–1212, 2013.

[39] R. Diao, F. Chao, T. Peng, N. Snooke, and Q. Shen, "Feature selection inspired classifier ensemble reduction," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1259–1268, 2014.

[40] R. Diao and Q. Shen, "Feature selection with harmony search," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 6, pp. 1509–1523, 2012.

[41] I. R. Dobson, "Using data and experts to make the wrong decision," in *Using Data to Improve Higher Education*. Springer, 2014, pp. 229–242.

[42] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[43] J. J. Dujmović, "Properties of local andness/orness," in *Theoretical Advances and Applications of Fuzzy Logic and Soft Computing*. Springer, 2007, pp. 54–63.

[44] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?" *Machine Learning*, vol. 54, no. 3, pp. 255–273, 2004.

[45] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in knowledge discovery and data mining*. AAAI press Menlo Park, 1996, vol. 21.

[46] M. Fazzolari, R. Alcalá, and F. Herrera, "A multi-objective evolutionary method for learning granularities based on fuzzy discretization to improve the accuracy-complexity trade-off of fuzzy rule-based classification systems: D-mofarc algorithm," *Applied Soft Computing*, vol. 24, pp. 470–481, 2014.

[47] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, "A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions," *Fuzzy Systems, IEEE Transactions on*, vol. 21, no. 1, pp. 45–65, 2013.

[48] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *Fuzzy Systems, IEEE Transactions on*, vol. 14, no. 5, pp. 676–697, 2006.

[49] A. Fernández, V. López, M. J. del Jesus, and F. Herrera, "Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges," *Knowledge-Based Systems*, vol. 80, pp. 109–121, 2015.

[50] J. Furnkranz and P. A. Flach, "Roc 'n' rule learning-towards a better understanding of covering algorithms," *Machine Learning*, vol. 58, no. 1, pp. 39–77, 2005.

[51] J. Fürnkranz, D. Gamberger, and N. Lavrač, *Foundations of rule learning*. Springer Science & Business Media, 2012.

[52] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Information Sciences*, vol. 181, no. 20, pp. 4340–4360, 2011.

[53] M. Galea and Q. Shen, "Simultaneous ant colony optimization algorithms for learning linguistic fuzzy rules," in *Swarm Intelligence in Data Mining*. Springer, 2006, pp. 75–99.

[54] D. García, A. González, and R. Pérez, "Overview of the slave learning algorithm: A review of its evolution and prospects," *International Journal of Computational Intelligence Systems*, vol. 7, no. 6, pp. 1194–1221, 2014.

[55] G. Giacinto and F. Roli, "An approach to the automatic design of multiple classifier systems," *Pattern Recognition Letters*, vol. 22, no. 1, pp. 25–33, 2001.

[56] D. E. Goldberg *et al.*, *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley Reading Menlo Park, 1989, vol. 412.

[57] A. González and R. Pérez, "Selection of relevant features in a fuzzy genetic learning algorithm," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 31, no. 3, pp. 417–425, 2001.

[58] A. Gonzalez and R. Perez, "Completeness and consistency conditions for learning fuzzy rules," *Fuzzy Sets and Systems*, vol. 96, no. 1, pp. 37–51, 1998.

[59] A. Gonzblez and R. Pérez, "Slave: A genetic learning system based on an iterative approach," *Fuzzy Systems, IEEE Transactions on*, vol. 7, no. 2, pp. 176–191, 1999.

[60] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM SIGMOD Record*, vol. 29, no. 2. ACM, 2000, pp. 1–12.

[61] I. Hayashi, T. Maeda, A. Bastian, and L. Jain, "Generation of fuzzy decision trees by fuzzy id3 with adjusting mechanism of and/or operators," in *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, vol. 1. IEEE, 1998, pp. 681–685.

[62] H. He and E. A. Garcia, "Learning from imbalanced data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 9, pp. 1263–1284, 2009.

[63] F. Herrera and E. Herrera-viedma, "Aggregation operators for linguistic weighted information," *Systems, Man, and Cybernetics, Part A: Systems, IEEE Transactions on*, pp. 646–656, 1997.

[64] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[65] Z. Huang, T. D. Gedeon, and M. Nikravesh, "Pattern trees induction: a new machine learning method," *Fuzzy Systems, IEEE Transactions on*, vol. 16, no. 4, pp. 958–970, 2008.

[66] J. Hühn and E. Hüllermeier, "Furia: an algorithm for unordered fuzzy rule induction," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[67] E. Hüllermeier, "Fuzzy sets in machine learning and data mining," *Applied Soft Computing*, vol. 11, no. 2, pp. 1493–1505, 2011.

[68] H. Ishibuchi, S. Mihara, and Y. Nojima, "Parallel distributed hybrid fuzzy gbml models with rule set migration and training data rotation," *Fuzzy Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 355–368, 2013.

[69] H. Ishibuchi, T. Murata, and M. Gen, "Performance evaluation of fuzzy rule-based classification systems obtained by multi-objective genetic algorithms," *Computers & Industrial Engineering*, vol. 35, no. 3-4, pp. 575–578, 1998.

[70] H. Ishibuchi, T. Murata, and I. Türkşen, "Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems," *Fuzzy Sets and Systems*, vol. 89, no. 2, pp. 135–150, 1997.

[71] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *Fuzzy Systems, IEEE Transactions on*, vol. 9, no. 4, pp. 506–515, 2001.

[72] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 29, no. 5, pp. 601–618, 1999.

[73] ——, "Three-objective genetics-based machine learning for linguistic rule extraction," *Information Sciences*, vol. 136, no. 1, pp. 109–133, 2001.

[74] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," *Fuzzy Sets and Systems*, vol. 52, no. 1, pp. 21–32, 1992.

[75] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms," *Fuzzy Sets and Systems*, vol. 65, no. 2, pp. 237–253, 1994.

[76] ——, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *Fuzzy Systems, IEEE Transactions on*, vol. 3, no. 3, pp. 260–270, 1995.

[77] H. Ishibuchi and T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," *Fuzzy Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 428–435, 2005.

[78] H. Ishibuchi, T. Yamamoto, and T. Nakashima, "Hybridization of fuzzy gbml approaches for pattern classification problems," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, no. 2, pp. 359–365, 2005.

[79] M. Z. Jahromi and M. Taheri, "A proposed method for learning rule weights in fuzzy rule-based classification systems," *Fuzzy Sets and Systems*, vol. 159, no. 4, pp. 449–459, 2008.

[80] C. Z. Janikow, "Fuzzy decision trees: issues and methods," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 28, no. 1, pp. 1–14, 1998.

[81] R. Jensen and C. Cornelis, "Fuzzy-rough nearest neighbour classification," in *Transactions on Rough Sets XIII*. Springer, 2011, pp. 56–72.

[82] R. Jensen, C. Cornelis, and Q. Shen, "Hybrid fuzzy-rough rule induction and feature selection," in *Proceedings of the 18th International Conference on Fuzzy Systems*. IEEE, 2009, pp. 1151–1156.

[83] R. Jensen and Q. Shen, "Fuzzy–rough attribute reduction with application to web categorization," *Fuzzy sets and systems*, vol. 141, no. 3, pp. 469–485, 2004.

[84] ——, "Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 12, pp. 1457–1471, 2004.

[85] ——, *Computational intelligence and feature selection: rough and fuzzy approaches*. John Wiley & Sons, 2008, vol. 8.

[86] Z. John Lu, "The elements of statistical learning: data mining, inference, and prediction," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 173, no. 3, pp. 693–694, 2010.

[87] J. Kennedy, R. Eberhart *et al.*, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, no. 2. Perth, Australia, 1995, pp. 1942–1948.

[88] J. Kennedy, J. F. Kennedy, and R. C. Eberhart, *Swarm intelligence*. Morgan Kaufmann, 2001.

[89] J. Kerr-Wilson and W. Pedrycz, "Some new qualitative insights into quality of fuzzy rule-based models," *Fuzzy Sets and Systems*, vol. 307, pp. 29–49, 2017.

[90] A. S. Koshiyama, M. M. Vellasco, and R. Tanscheit, "Gpfis-class: A genetic fuzzy system based on genetic programming for classification problems," *Applied Soft Computing*, vol. 37, pp. 561–571, 2015.

[91] L. I. Kuncheva, "Switching between selection and fusion in combining classifiers: an experiment," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 2, pp. 146–156, 2002.

[92] A. Laurent, "Generating fuzzy summaries: a new approach based on fuzzy multidimensional databases," *Intelligent Data Analysis*, vol. 7, no. 2, pp. 155–177, 2003.

[93] L. Leydesdorff, "Can scientific journals be classified in terms of aggregated journal-journal citation relations using the journal citation reports?" *Journal of the Association for Information Science and Technology*, vol. 57, no. 5, pp. 601–613, 2006.

[94] X. Liu, X. Feng, and W. Pedrycz, "Extraction of fuzzy rules from fuzzy decision trees: An axiomatic fuzzy sets (afs) approach," *Data & Knowledge Engineering*, vol. 84, pp. 1–25, 2013.

[95] Y. Liu and E. E. Kerre, "An overview of fuzzy quantifiers.(i). interpretations," *Fuzzy Sets and Systems*, vol. 95, no. 1, pp. 1–21, 1998.

[96] A. Lotfi, H. C. Andersen, and A. C. Tsoi, "Interpretation preservation of adaptive fuzzy inference systems," *International Journal of Approximate Reasoning*, vol. 15, no. 4, pp. 379–394, 1996.

[97] A. Lotfi, C. Langensiepen, S. M. Mahmoud, and M. J. Akhlaghinia, "Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour," *Journal of ambient intelligence and humanized computing*, vol. 3, no. 3, pp. 205–218, 2012.

[98] J. P. Lucas, A. Laurent, M. N. Moreno, and M. Teisseire, "A fuzzy associative classification approach for recommender systems," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 20, no. 04, pp. 579–617, 2012.

[99] B. L. W. H. Y. Ma and B. Liu, "Integrating classification and association rule mining," in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, 1998.

[100] Y. Ma, G. Chen, and Q. Wei, "A novel business analytics approach and case study–fuzzy associative classifier based on information gain and rule-covering," *Journal of Management Analytics*, vol. 1, no. 1, pp. 1–19, 2014.

[101] S. Mabu, C. Chen, N. Lu, K. Shimada, and K. Hirasawa, "An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 1, pp. 130–139, 2011.

[102] S. Mahmoud, A. Lotfi, and C. Langensiepen, "Behavioural pattern identification and prediction in intelligent environments," *Applied Soft Computing*, vol. 13, no. 4, pp. 1813–1822, 2013.

[103] E. H. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," *International Journal of Man-Machine Studies*, vol. 8, no. 6, pp. 669–678, 1976.

[104] E. G. Mansoori, M. J. Zolghadri, and S. D. Katebi, "A weighting function for improving fuzzy classification systems performance," *Fuzzy Sets and Systems*, vol. 158, no. 5, pp. 583–591, 2007.

[105] ——, "Sgerd: A steady-state genetic algorithm for extracting fuzzy classification rules from data," *Fuzzy Systems, IEEE Transactions on*, vol. 16, no. 4, pp. 1061–1071, 2008.

[106] J. G. Marin-Blazquez, Q. Shen, and A. F. Gomez-Skarmeta, "From approximative to descriptive models," in *Proceedings of the 9th International Conference on Fuzzy Systems*, vol. 2. IEEE, 2000, pp. 829–834.

[107] J. G. Marín-Blázquez and Q. Shen, "From approximative to descriptive fuzzy classifiers," *Fuzzy Systems, IEEE Transactions on*, vol. 10, no. 4, pp. 484–497, 2002.

[108] C. Mencar and A. M. Fanelli, "Interpretability constraints for fuzzy information granulation," *Information Sciences*, vol. 178, no. 24, pp. 4585–4618, 2008.

[109] T. M. Mitchell, "Machine learning. 1997," *Burr Ridge, IL: McGraw Hill*, vol. 45, 1997.

[110] S. Mitra, S. K. Pal, and P. Mitra, "Data mining in soft computing framework: a survey," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 3–14, 2002.

[111] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part i," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 4–19, 2014.

[112] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[113] D. Nauck and R. Kruse, "How the learning of rule weights affects the interpretability of fuzzy systems," in *Proceedings of the 7th International Conference on Fuzzy Systems*, vol. 2. IEEE, 1998, pp. 1235–1240.

[114] N. Nithya and K. Duraiswamy, "Correlated gain ratio based fuzzy weighted association rule mining classifier for diagnosis health care data," *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 4, pp. 1453–1464, 2015.

[115] K. Nozaki, H. Ishibuchi, and H. Tanaka, "Adaptive fuzzy rule-based classification systems," *Fuzzy Systems, IEEE Transactions on*, vol. 4, no. 3, pp. 238–250, 1996.

[116] F. P. Pach, A. Gyenesei, and J. Abonyi, "Compact fuzzy association rule-based classifier," *Expert systems with applications*, vol. 34, no. 4, pp. 2406–2416, 2008.

[117] D. P. Pancho, J. M. Alonso, O. Cordón, A. Quirin, and L. Magdalena, "Fingrams: visual representations of fuzzy rule-based inference for expert analysis of comprehensibility," *Fuzzy Systems, IEEE Transactions on*, vol. 21, no. 6, pp. 1133–1149, 2013.

[118] I. Partalas, G. Tsoumakas, and I. Vlahavas, "Pruning an ensemble of classifiers via reinforcement learning," *Neurocomputing*, vol. 72, no. 7, pp. 1900–1909, 2009.

[119] R.-E. Precup and H. Hellendoorn, "A survey on industrial applications of fuzzy control," *Computers in Industry*, vol. 62, no. 3, pp. 213–226, 2011.

[120] P. Pulkkinen and H. Koivisto, "Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms," *International Journal of Approximate Reasoning*, vol. 48, no. 2, pp. 526–543, 2008.

[121] ——, "A dynamically constrained multiobjective genetic fuzzy system for regression problems," *Fuzzy Systems, IEEE Transactions on*, vol. 18, no. 1, pp. 161–177, 2010.

[122] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[123] ——, "Learning logical definitions from relations," *Machine Learning*, vol. 5, no. 3, pp. 239–266, 1990.

[124] ——, *C4. 5: programs for machine learning*. Morgan kaufmann, 1993, vol. 1.

[125] K. Rasmani, J. M. Garibaldi, Q. Shen, and I. O. Ellis, "Linguistic rulesets extracted from a quantifier-based fuzzy classification system," in *Proceedings of the 18th International Conference on Fuzzy Systems*. IEEE, 2009, pp. 1204–1209.

[126] K. A. Rasmani and Q. Shen, "Weighted linguistic modelling based on fuzzy subsethood values," in *Proceedings of the 12th International Conference on Fuzzy Systems*, vol. 1. IEEE, 2003, pp. 714–719.

[127] ——, "Modifying weighted fuzzy subsethood-based rule models with fuzzy quantifiers," in *Proceedings of the 13th International Conference on Fuzzy Systems*, vol. 3. IEEE, 2004, pp. 1679–1684.

[128] ——, "Data-driven fuzzy rule generation and its application for student academic performance evaluation," *Applied Intelligence*, vol. 25, no. 3, pp. 305–319, 2006.

[129] A. Rezaee Jordehi and J. Jasni, "Parameter selection in particle swarm optimisation: a survey," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 527–542, 2013.

[130] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[131] A. Salman, I. Ahmad, and S. Al-Madani, "Particle swarm optimization for task assignment problem," *Microprocessors and Microsystems*, vol. 26, no. 8, pp. 363–371, 2002.

[132] S. L. Salzberg, "C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993," *Machine Learning*, vol. 16, no. 3, pp. 235–240, 1994.

[133] J. A. Sanz, A. Fernandez, H. Bustince, and F. Herrera, "Ivturs: A linguistic fuzzy rule-based classification system based on a new interval-valued fuzzy reasoning method with tuning and rule selection," *Fuzzy Systems, IEEE Transactions on*, vol. 21, no. 3, pp. 399–411, 2013.

[134] J. A. Sanz, M. Galar, A. Jurio, A. Brugos, M. Pagola, and H. Bustince, "Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system," *Applied Soft Computing*, vol. 20, pp. 103–111, 2014.

[135] R. Senge and E. Hullermeier, "Top-down induction of fuzzy pattern trees," *Fuzzy Systems, IEEE Transactions on*, vol. 19, no. 2, pp. 241–252, 2011.

[136] R. Senge and E. Hüllermeier, "Fast fuzzy pattern tree learning for classification," *Fuzzy Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 2024–2033, 2015.

[137] J. Shawe-Taylor and S. Sun, "A review of optimization methodologies in support vector machines," *Neurocomputing*, vol. 74, no. 17, pp. 3609–3618, 2011.

[138] Q. Shen and A. Chouchoulas, "A rough-fuzzy approach for generating classification rules," *Pattern Recognition*, vol. 35, no. 11, pp. 2425–2438, 2002.

[139] Q. Shen, R. Diao, and P. Su, "Feature selection ensemble." *Turing-100*, vol. 10, pp. 289–306, 2012.

[140] Q. Shen and R. Jensen, "Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring," *Pattern Recognition*, vol. 37, no. 7, pp. 1351–1363, 2004.

[141] Q. Shen and R. Leitch, "Fuzzy qualitative simulation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 4, pp. 1038–1061, 1993.

[142] D. Soria, J. M. Garibaldi, A. R. Green, D. G. Powe, C. C. Nolan, C. Lemetre, G. R. Ball, and I. O. Ellis, "A quantifier-based fuzzy classification system for breast cancer patients," *Artificial intelligence in medicine*, vol. 58, no. 3, pp. 175–184, 2013.

[143] P. Su, T. Chen, C. Shang, and Q. Shen, "Nearest neighbour-guided induced owa and its application to journal ranking," in *Fuzzy Systems, 2014 IEEE International Conference on*. IEEE, 2014, pp. 1794–1800.

[144] P. Su, C. Shang, T. Chen, and Q. Shen, "Exploiting data reliability and fuzzy clustering for journal ranking," *Fuzzy Systems, IEEE Transactions on*, vol. 25, no. 5, pp. 1306–1319, 2017.

[145] P. Su, C. Shang, and Q. Shen, "Link-based approach for bibliometric journal ranking," *Soft Computing*, vol. 17, no. 12, pp. 2399–2410, 2013.

[146] ——, "A hierarchical fuzzy cluster ensemble approach and its application to big data clustering," *Journal of Intelligent & Fuzzy Systems*, vol. 28, no. 6, pp. 2409–2421, 2015.

[147] P. Su, C. Shang, Y. Zhao, T. Chen, and Q. Shen, "Fuzzy rough feature selection based on owa aggregation of fuzzy relations," in *Fuzzy Systems, 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[148] P. Su, Q. Shen, T. Chen, and C. Shang, "Ordered weighted aggregation of fuzzy similarity relations and its application to detecting water treatment plant malfunction," *Engineering Applications of Artificial Intelligence*, vol. 66, pp. 17–29, 2017.

[149] R. Suganya and R. Shanthi, "Fuzzy c-means algorithm-a review," *International Journal of Scientific and Research Publications*, vol. 2, no. 11, p. 1, 2012.

[150] M. Sugeno and G. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, no. 1, pp. 15–33, 1988.

[151] The Australian Research Council(ARC), *Excellence in research for australia (era), [online], "Available: http://www.arc.gov.au/era/"*, 2010.

[152] t. D. f. E. the Scottish Funding Council (SFC), the Higher Education Funding Council for Wales (HEFCW) and N. I. D. Learning, *Research excellence framework, [Online], Available: http://www.ref.ac.uk/*.

[153] I. Triguero, J. Derrac, S. Garcia, and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 1, pp. 86–100, 2012.

[154] G. Tsoumakas, I. Partalas, and I. Vlahavas, "A taxonomy and short review of ensemble selection," in *Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications*, 2008.

[155] R. Viertl, *Statistical methods for non-precise data*. CRC Press, 1995.

[156] M.-A. Vila, J.-C. Cubero, J.-M. Medina, and O. Pons, "Using owa operator in flexible query processing," in *The ordered weighted averaging operators*. Springer, 1997, pp. 258–274.

[157] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 6, pp. 1414–1427, 1992.

[158] L.-X. Wang, *A course in fuzzy systems*. Prentice-Hall press, USA, 1999.

[159] N. Wang, M. J. Er, and M. Han, "Dynamic tanker steering control using generalized ellipsoidal-basis-function-based fuzzy neural networks," *Fuzzy Systems, IEEE Transactions on*, vol. 23, no. 5, pp. 1414–1427, 2015.

[160] X. Wang, X. Liu, W. Pedrycz, and L. Zhang, "Fuzzy rule based decision trees," *Pattern Recognition*, vol. 48, no. 1, pp. 50–59, 2015.

[161] X. Wang, B. Chen, G. Qian, and F. Ye, "On the optimization of fuzzy decision trees," *Fuzzy Sets and Systems*, vol. 112, no. 1, pp. 117–125, 2000.

[162] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[163] S.-L. Wu, Y.-T. Liu, T.-Y. Hsieh, Y.-Y. Lin, C.-Y. Chen, C.-H. Chuang, and C.-T. Lin, "Fuzzy integral with particle swarm optimization for a motor-imagery-based brain–computer interface," *Fuzzy Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 21–28, 2017.

[164] J. Wyatt, "Nervous about artificial neural networks?" *The Lancet*, vol. 346, no. 8984, pp. 1175–1177, 1995.

[165] Z. Xu, "Dependent owa operators," in *Modeling Decisions for Artificial Intelligence*. Springer, 2006, pp. 172–178.

[166] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[167] ——, "Using stress functions to obtain owa operators," *Fuzzy Systems, IEEE Transactions on*, vol. 15, no. 6, pp. 1122–1129, 2007.

[168] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and Systems*, vol. 69, no. 2, pp. 125–139, 1995.

[169] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[170] ——, "The concept of a linguistic variable and its application to approximate reasoningâĂŤi," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[171] L. Zhang, L. Gao, X. Shao, L. Wen, and J. Zhi, "A pso-fuzzy group decision-making support system in vehicle performance evaluation," *Mathematical and Computer Modelling*, vol. 52, no. 11, pp. 1921–1931, 2010.

[172] S.-M. Zhou and J. Q. Gan, "Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling," *Fuzzy Sets and Systems*, vol. 159, no. 23, pp. 3091–3131, 2008.

[173] H.-J. Zimmermann, *Fuzzy set theory and its applications*. Springer Science & Business Media, 2011.

[174] M. J. Zolghadri and E. G. Mansoori, "Weighting fuzzy classification rules using receiver operating characteristics (roc) analysis," *Information Sciences*, vol. 177, no. 11, pp. 2296–2307, 2007.