ADVANCES IN FUZZY RULE-BASED SYSTEM FOR PATTERN CLASSIFICATION

CHUA TECK WEE

(B.Eng.(Hons.),UTM)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2010

Acknowledgments

I am grateful to many people for supporting me not only intellectually, but also furnishing me with joy and inspiration in other aspects of life outside of work. This acknowledgement can only give a glimpse on how much I benefited and learnt from all my mentors, colleagues, friends, and family. Thank you so much to all of you.

First of all, I wish to sincerely thank my supervisor Assoc. Prof. Tan Woei Wan, who supplied me with invaluable advice and guidance throughout my time at the university concerning my research, writing, organisation, and life. Her insights in fuzzy logic are always stimulating and many chapters of this thesis were shaped by the numerous discussions we had in the weekly meetings since year 2005.

I am also thankful to Assoc. Prof. Tan Kay Chen and Assoc. Prof. Dipti Srinivasan for building up my fundamentals in Neural Networks and Evolutionary Computation. I would also like to express my gratitude to my colleagues for their inspirational input and my friends for their true friendship. Finally, I am forever grateful to my loving family back in Malaysia. This thesis would not have been possible without their encouragement and love.

Contents

Summary

Pattern classification encompasses a wide range of information processing problems that are of great practical significance, from the classification of handwritten characters, to fault detection in machinery and medical diagnosis. Fuzzy logic system was initially introduced to solve a pattern classification problem because the system has similar reasoning style to human being. One of the main advantages of fuzzy logic is that it enables qualitative domain knowledge about a classification task to be deployed in the algorithmic structure. Despite the popularity of fuzzy logic system in pattern classification, a conventional singleton type-1 fuzzy logic system does not capture uncertainty in all of its manifestations, particularly when it arises from the noisy input and the vagueness in the shape of the membership function. The aim of this study is to seek a better understanding of the properties of extensional fuzzy rule-based classifiers (FRBCs), namely non-singleton FRBC and interval type-2 FRBC. Besides, this research aimed at systemising the learning procedure for fuzzy rule-based classifier.

Non-singleton FRBC was found to have noise suppression capability. Therefore, it can better cope with input that is corrupted with noise. In addition, the analysis demonstrated that non-singleton FRBC is capable of producing variable boundary which may be useful to resolve the overlapping boundary between classes. The significance is that non-singleton FRBC may reduce the complexity of feature extraction by extending the possibility to use the features that are easier to extract but contain more uncertainties. As an extension to type-1 fuzzy classifier, type-2 classifier appears to have better performance and robustness due to its richness of footprint of uncertainty (FOU) in membership function. The proposed FOU design methodology can be useful when one is uncertain about the descriptions for the features (i.e., the membership function). The robustness study and extensive experimental results suggest that the performance of type-2 FRBC is at least comparable, if not better than type-1 counterpart.

Designing and optimising FRBCs are just as important as understanding the properties of different types of fuzzy classifiers. In view of this, an efficient learning algorithm based on support vector machine and fuzzy c-means algorithm was proposed. Not only that the resulting fuzzy classifier has a compact rule base, but it also has good generalisation capability. Besides, the curse of dimensionality which is often faced by FRBCs can be avoided. In the later part of this thesis, it was also shown that the proposed fuzzy rule-based initialisation procedure can enhance the performance of conventional crisp and fuzzy K-Nearest Neighbor (K-NN) when the training data is limited. Moreover, the successful implementation of the FRBC to classify faults in induction motor has provided clear evidence of its practical applicability.

In conclusion, it is foreseeable that FRBCs will continue to play an important role in pattern classification. With the advances in extensional FRBCs, the

uncertainties which the conventional classifiers failed to address for, can now be handled more effectively.

List of Figures

List of Tables

Chapter 1

Introduction

Pattern classification problems emerge constantly in everyday life: reading texts, identifying people, retrieving objects, or finding the way in a city. In order to perceive and react to different situations, individuals must process the sensory information received by the eyes, ears, skin etc. This information contains the features or attributes of the objects. Humans recognise two objects as being similar because they have similarly valued common attributes. Often these are problems which many humans solve in a seemingly effortless fashion. In contrast, their solution using computers has, in many cases, proved to be immensely difficult. In order to have effective solutions, it is important to adopt a principled approach based on sound theoretical concepts. Advances in pattern classification is important for building intelligent machines that emulate humans. Fuzzy logic system is one of the popular machine learning techniques that has been successfully applied to pattern classification.

It is well known that the concept of fuzzy set first originated from the study of problems related to pattern classification [1]. This is not surprising because the process of recognising a pattern, which is an important aspect of human perception, is a fuzzy process in nature. The fuzziness can include the changes in object orientation and size, degree of incompleteness and distortion, amount of background noise, vague descriptions, imprecise measurements, conflicting or ambiguous information, random occurrences and etc. A large amount of literature has been published dealing with fuzzy pattern classification, the search results retrieved from the search engine upon the keyword "fuzzy classifiers" is astonishing. Google search engine returned this statistic at 10 p.m. on August 22, 2009:

"Results 1 - 10 of about 524,000 for fuzzy classifiers. (0.37 seconds)."

It seems that applications of fuzzy pattern classification are far ahead of the theory on the matter. Majority of the works only involved conventional type-1 fuzzy logic systems or used simple notion of fuzzy sets. The advantages and properties of extensional fuzzy logic systems (FLSs) such as non-singleton FLSs and type-2 FLSs are far from being explored. The chapter will provide a brief introduction to pattern classification and fuzzy sets and more attention will be given to the overview of fuzzy pattern classification.

1.1 Fundamental Concepts of Pattern Classification

Classification can be divided into supervised and unsupervised classification. In supervised classification, also termed *discrimination*, a set of data samples consisting of a set of variables is available. All the samples in the data set are labelled; they are thus all assigned to a specific class. With unsupervised classification, sometimes termed *clustering*, the samples in the data set are not labelled.

Class is a core notion in pattern classification. Let Ω be a set of class labels $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ where ω_i is the class label. The term *class* symbolises a group of objects with a common characteristic or common meaning. *Features* (variables) are used to describe the objects numerically. The feature values for a given object are arranged as an n-dimensional vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \Re^n$. The real space $\Re^n$ is called *feature space*, each axis corresponding to a physical feature. A classifier is any function:

$$D : \Re^n \rightarrow \Omega \tag{1.1}$$

The decision functions partition the feature space $\Re^n$ into c (not necessarily compact) *decision regions* or *classification regions* denoted by $R_1, R_2, \ldots, R_c$

$$R_i = \left\{ x | x \in \Re^n, g_i(x) = \max_{k=1,\ldots,c} g_k(x) \right\}, \quad i = 1, \ldots, c \tag{1.2}$$

where $G = \{g_1(x), g_2(x), \ldots, g_c(x)\}$ is a set of decision functions in the canonical model of a classifier. The boundaries of the decision regions are called *classification boundaries.* A point on the boundary can be assigned to any of the bordering classes. If the classes in data set, $\mathbf{Z}$ can be separated completely from each other by a hyperplane (a point in $\Re$, a line in $\Re^2$, a plane in $\Re^3$), they are called linearly separable [2].

There are two methods to develop classifiers. The first one is parametric method, in which a priori knowledge of data distributions is assumed. A classical example of classifier that uses this approach is Bayes classifier. The second one is nonparametric method, in which no a priori knowledge is assumed. Neural

Networks [3], fuzzy systems [4], and Support Vector Machines (SVM) [5] are typical nonparametric classifiers. The classifier acquires its decision function through the training using input-output pairs.

The typical components of a classifier and the design flow of a classifier are shown in Fig. 1.1. The feature extraction step transforms raw data (observation space) into feature vectors (feature space). The resulting feature space is of a much lower dimension than the observation space. The next step is a transformation of the feature space into a decision space, which is defined by a (finite) set of classes. A classifier, which is an algorithm, generates a partitioning of the feature space into a number of decision regions. After the classifier is designed and a desired level of performance is achieved, it can be used to classify new objects. This means that the classifier assigns every feature vector in the feature space to a class in the decision space.

1.2 Fundamental Concepts of Fuzzy Logic System

Fuzzy set theory is not a theory that permits vagueness in our computations, but it is rather a methodology to show how to tackle uncertainty, and to handle imprecise information in a complex situation. Fuzzy sets are the core element in Fuzzy Logic. They are characterised by membership functions which are associated with terms or words used in the antecedent and consequents of rules, and with input and output to the fuzzy logic system.

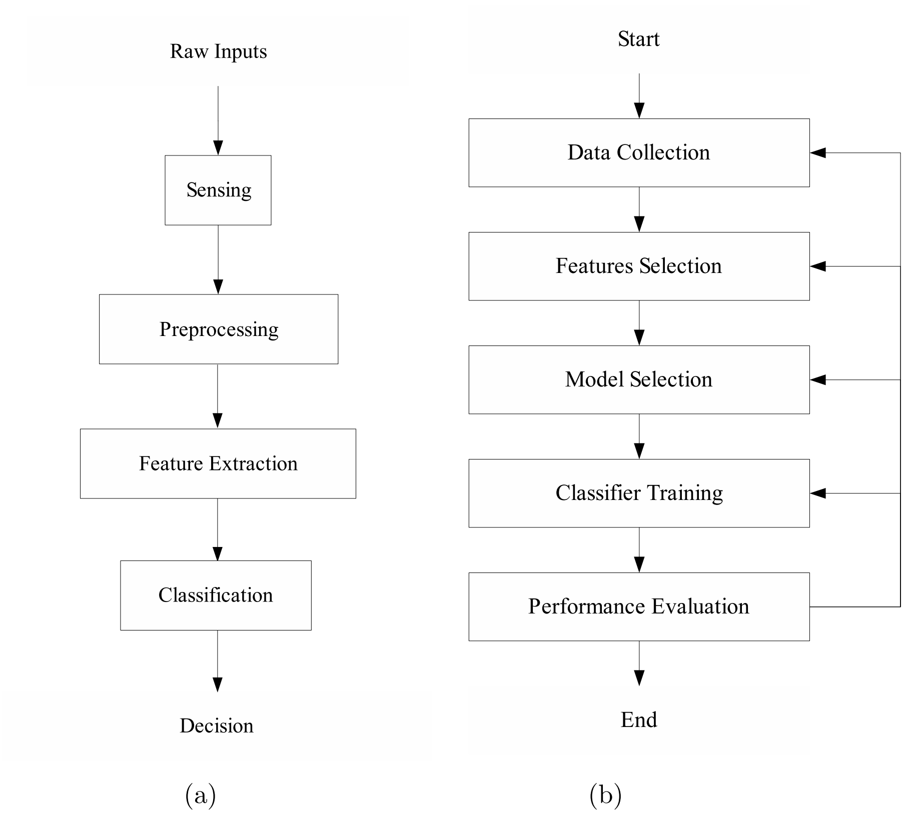


Figure 1.1: (a) The components of a typical classifier and (b) the classifier design flow.

A crisp set A in a universe of discourse X can be defined as

$$A \Rightarrow \mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ \\ 0 & \text{if } x \notin A \end{cases} \tag{1.3}$$

A fuzzy set A is a generalization of a crisp set. It is defined on a universe of discourse X and is characterised by a membership function $\mu_A(x)$ that takes on values in the interval [0,1]. A membership function provides a measure of the degree of similarity of an element in X to the fuzzy set.

$$A = \{(x, \mu_A(x)) \,|\, x \in X\} \tag{1.4}$$

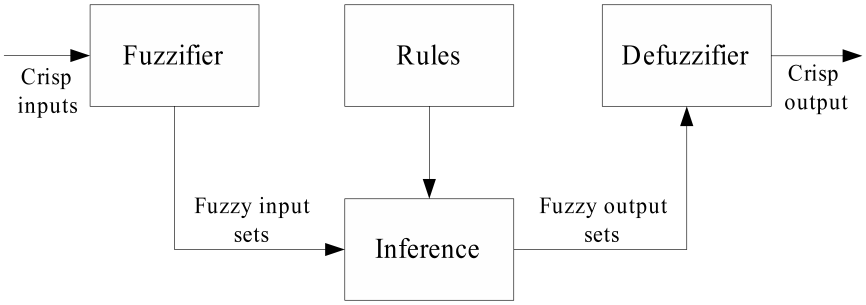


Figure 1.2: Type-1 fuzzy logic system (FLS).

1.2.1 Type-1 Fuzzy Logic System

A type-1 fuzzy set, A, for a single variable, $x \in X$ has already been defined in (1.4). Type-1 membership function, $\mu_A(x)$ is constrained to be between 0 and 1 for all $x \in X$, and is a two-dimensional function.

A fuzzy logic system (FLS) that is described completely in terms of type-1 fuzzy sets is called a type-1 FLS. Fig. 1.2 shows a fuzzy logic system. The system contains four components – fuzzifier, rules, inference engine, and defuzzifier. The fuzzifier maps a crisp point $x = (x_1, \ldots, x_p)^T \in X_1 \times X_2 \times \ldots \times X_p \equiv X$ into a fuzzy set A_x in X. The most widely used fuzzifier is the singleton fuzzifier which is nothing more than a fuzzy singleton, i.e., A_x is a fuzzy singleton with support x' if $\mu_{A_x}(x) = 1$ for $x = x'$ and $\mu_{A_x}(x) = 0$ for all other $x \in X$ with $x \neq x'$. Nonsingleton fuzzifier, however, maps $x_i = x'_i$ into a fuzzy number where a membership function is associated with it. In particular, $\mu_{X_i}(x'_i) = 1$ $(i = 1, \ldots, p)$ and $\mu_{X_i}(x'_i)$ decreases from unity as x_i moves away from x'_i. Rules are the heart of a FLS and they can be expressed as a collection of IF-THEN statements. The IF-part of a rule is its antecedent, and the THEN-part of a rule is its consequent. The terms that appear in the antecedents or consequents of rules are associated with type-1 fuzzy sets. Next, the inference engine maps fuzzy input sets to fuzzy output sets. It handles the way in which rules are activated and combined. Finally, the defuzzifier transforms the output fuzzy sets into crisp outputs.

1.2.2 Type-2 Fuzzy Logic System

Type-1 fuzzy sets are not able to convey the uncertainties about the membership functions. Some typical sources of uncertainties are: (i) the meaning of the words that are used in the antecedents and consequents can be uncertain (words mean different things to different people), (ii) knowledge extracted from a group of experts do not all agree thus the consequents may have a histogram of values associated with them, (iii) inputs or measurements may be noisy [6]. Unlike type-1 membership functions which are two-dimensional, type-2 fuzzy membership functions are three-dimensional. The additional degree of freedom offered by the new third dimension enables type-2 fuzzy sets to model the aforementioned uncertainties. Type-2 fuzzy set is formally denoted as $\tilde{A}$ and is characterised by a type-2 membership function $\mu_{\tilde{A}}(x, u)$ where $x \in X$ and $u \in J_x \subseteq [0, 1]$, i.e.,

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \,|\, \forall x \in X, \qquad \forall u \in J_x \subseteq [0, 1]\} \tag{1.5}$$

in which $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. The domain of a secondary membership function is called the primary membership of x which is J_x (see Fig. 1.3). $\tilde{A}$ can also be expressed as

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u)/(x, u) \qquad J_x \subseteq [0, 1] \tag{1.6}$$

where $\iint$ denotes union over all admissible x and μ.

An interval type-2 (IT2) fuzzy set, $\widetilde{A}$ is characterised as:

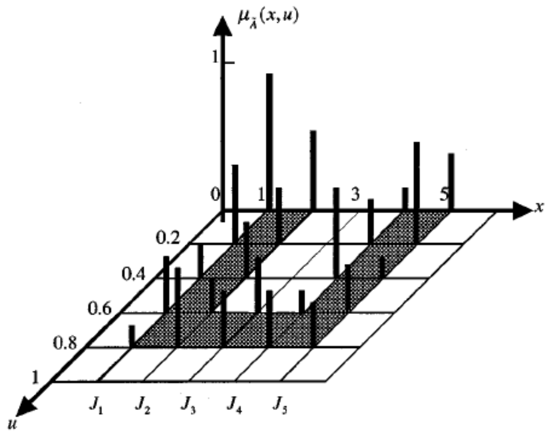


Figure 1.3: Example of a type-2 membership function. J_x, the primary membership of x, is the domain of secondary membership function.

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} 1/(x,u) \qquad J_x \subseteq [0,1] \tag{1.7}$$

where x, the primary variable, has domain X; $u \in U$, the secondary variable, has domain J_x at each $x \in X$. Uncertainty about the shape and position of $\widetilde{A}$ is conveyed by the union of all the primary memberships, which is called the footprint of uncertainty (FOU) of $\widetilde{A}$ (see Fig. 1.4), i.e.

$$FOU(\widetilde{A}) = \bigcup_{\forall x \in X} J_x = \{(x,u) : u \in J_x \subseteq [0,1]\} \tag{1.8}$$

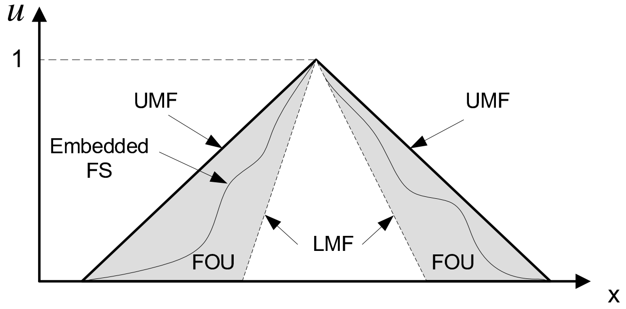


Figure 1.4: FOU (shaded), LMF (dashed), UMF (solid) and an embedded FS (wavy line) for IT2 FS $\widetilde{A}$.

The upper membership function (UMF) and lower membership function (LMF) of $\widetilde{A}$ are two type-1 MFs that bound the FOU (Fig. 1.4). The UMF is associated with the upper bound of FOU and is denoted $\bar{\mu}_{\widetilde{A}}$, $\forall x \in X$, and the LMF is associated with the lower bound of FOU and is denoted $\underline{\mu}_{\widetilde{A}}$, $\forall x \in X$, i.e.

$$\bar{\mu}_{\widetilde{A}}(x) \equiv \overline{FOU(\widetilde{A})} \quad \forall x \in X \tag{1.9}$$

$$\underline{\mu}_{\widetilde{A}}(x) \equiv \underline{FOU(\widetilde{A})} \quad \forall x \in X \tag{1.10}$$

$$J_x = \{(x,u) : u \in [\underline{\mu}_{\widetilde{A}}(x), \bar{\mu}_{\widetilde{A}}(x)]\} \tag{1.11}$$

so that $FOU(\widetilde{A})$ in (1.8) can also be expressed as

$$FOU(\widetilde{A}) = \bigcup_{\forall x \in X} [\underline{\mu}_{\widetilde{A}}(x), \bar{\mu}_{\widetilde{A}}(x)]. \tag{1.12}$$

For interval type-2 fuzzy set, J_x, the primary membership of x is reduced to an interval set which is defined in (1.11); and, the secondary grades of $\widetilde{A}$ all equal 1. Note that (1.7) means: $\widetilde{A} : X \to \{[a,b] : 0 \le a \le b \le 1\}$.

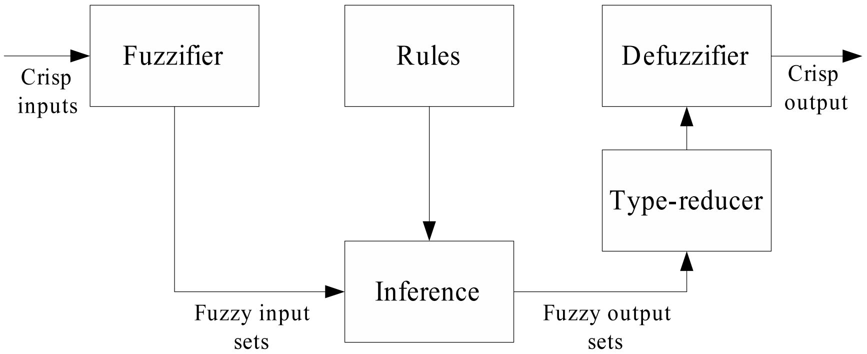
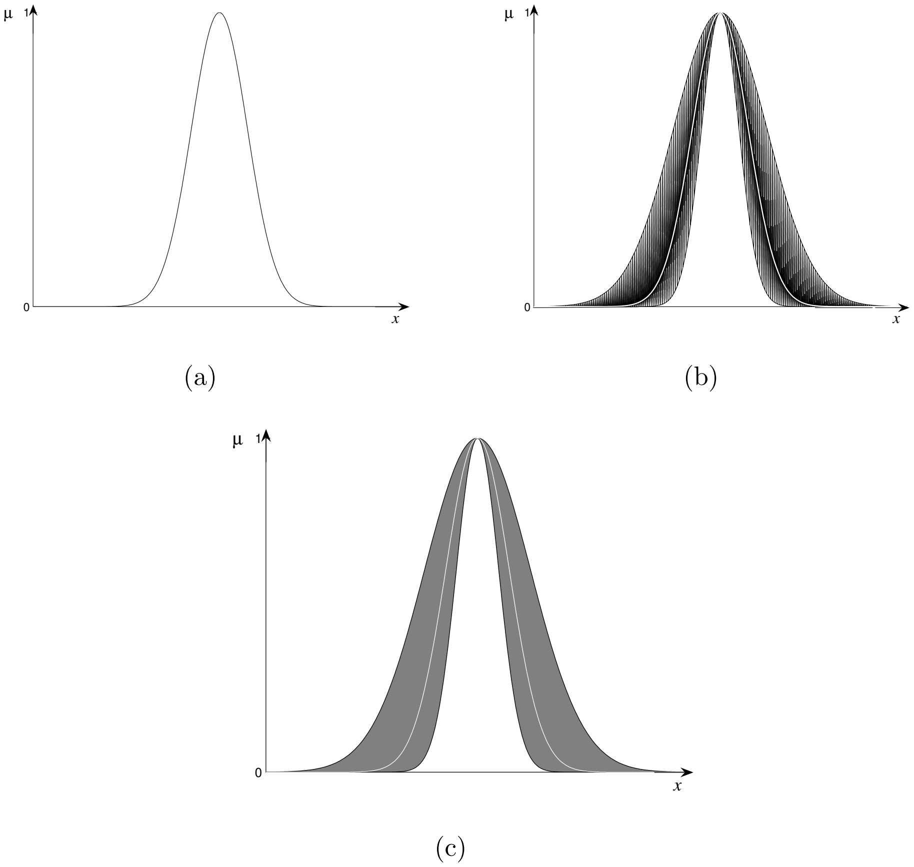


Figure 1.5: Type-2 fuzzy logic system (FLS).

A FLS that is described using at least one type-2 fuzzy set is called a type-2 FLS. A type-2 FLS is depicted in Fig. 1.5. The first observation is that a type-2



(a)

(b)

(c)

Figure 1.6: (a) Type-1 membership function, (b) type-2 membership function (the bounded area is not shaded uniformly to reflect that the secondary membership grades are in [0,1]), and (c) interval type-2 membership function (the bounded area is shaded uniformly to indicate that all the secondary grades are unity).

FLS is very similar to a type-1 FLS. The major structural difference is that there exists a type-reducer block before the defuzzifier block. As the name suggests, type-reducer maps a type-2 set into a type-1 set before the defuzzifier performs defuzzification on the later set. Typical type-reduction methods are: centroid, center-of-sums, height, modified height, and center-of sets.

1.3 Overview of Fuzzy Pattern Classification

Statistical and Neural Networks based approaches are among the most popular pattern classification methods. However, most of these methods produce so-called “crisp” classifiers, those generate decisions without any accompanying confidence measure. The main feature of *crisp* classification is that each pattern only belongs to a single class, in spite of weak correlation between pattern properties with thematic class attributes. On the other hand, fuzzy classification provides a measure of support for the decision (and also alternative decisions) that provides the analyst with greater insight. In other words, each pattern may belong at the same time to each of the existing classes with various grades of membership.

Usually fuzzy pattern classification is associated with fuzzy clustering or with fuzzy rule-based classifiers. In a broader view, fuzzy pattern classification can be any pattern classification paradigm that involves fuzzy sets. While only using simple notion of fuzzy sets, fuzzy clustering appears to be the most successful branch of fuzzy pattern classification so far. The fuzzy c-means algorithm devised by Bezdek [7] has admirable popularity in a great number of fields, both engineering and non-engineering. On the other hand, fuzzy rule-based classifier provides a systematic way to incorporate experts’ knowledge. It has the advantage of interpretability over other non-linear systems such as Neural Networks and Support Vector Machines. Fuzzy rule-based system are easily understood through linguistic interpretation of each fuzzy rule which mimics human reasoning.

1.3.1 Why should we use fuzzy classifier?

In many applications such as medical or fault diagnoses, the users need not only the class label of an object but also some additional information (e.g. how typical the object is, how severe the disease is). Fuzzy classifier is able to provide extra information on the certainty of the decision. Quite often classification is performed with some degree of uncertainty. Either the classification result itself may be in doubt, or the classified pattern may belong to some degree in more than one class. If the certainty grades of the available decisions are close, then the expert is able to verify the classification results by examining the immediate feasible decision next to the decision with highest certainty grade.

In some problems, there is insufficient information to properly implement classical (e.g., statistical) pattern classification methods. Such are the problems where we have difficulty in obtaining training or design sets with sufficient data and which are representative of the classes to be distinguished. For example, in the application of machine fault detection the faulty signal is not accessible during the classifier training stage. It would be expensive and not feasible to damage the machine purposely to collect the faulty data. Therefore, the experts' knowledge can be utilised when designing the fuzzy classifier.

Fuzzy classifiers based on if-then rules might be “transparent” or “interpretable”, i.e., the end user (expert) is able to verify the classification paradigm. This notion of interpretability is crucial in most critical applications where the expert needs to verify the reasoning steps, the plausibility, consistency or completeness of the fuzzy rule-base used in producing an automatic classification task. Nevertheless, this verification is more suitable for small-scale systems, i.e., systems which do not use a large number of input features and big rule bases.

1.3.2 Types of fuzzy classifiers

For classification problems, many approaches based on fuzzy set theory can be found in the literature. The existing fuzzy classification methods may be grouped into the following four categories [8]

1. Methods based on fuzzy relations
2. Methods based on fuzzy pattern matching
3. Methods based on fuzzy clustering
4. Other methods which are more or less generalization of classical approaches.

Pedrycz [9] commented that fuzzy relation methods do not take any information concerning the importance of the features and, in its essence, has an implicit character. The fuzzy relation of the classifier contains all the information conveyed by the training set of the patterns. He suggested that the approximate solution of the fuzzy relational equation can be a problem from a computational point of view. On the other hand, fuzzy pattern matching method is explicit in its character, an additional information dealt with the importance of a feature is required to make the classifier performs effectively.

A more popular way to categorise fuzzy classifier is to identify the existence of fuzzy rule base. Thus, fuzzy classifiers can be divided into two major groups:

1. Fuzzy if-then classifiers

2. Non fuzzy if-then classifiers.

The following sections explain these two groups explicitly.

1.3.2.1 Fuzzy rule-based classifier

The general structure of a fuzzy rule-based classifier is shown in Fig. 1.7. As with any other classification system, a preprocessing unit filters the data, if necessary, and transforms the high dimensional inputs into a subset of desired features in feature space. Next, the fuzzifier transforms the crisp input values into a fuzzy set. The inference engine then combines the fuzzified input with "IF-THEN" rules using fuzzy t-norm to derive the firing strength for each rule. The IF-part of a rule is its antecedent, and the THEN-part of a rule is its consequent. Finally, the decision making unit will select the fuzzy rule with maximum degree of truth (i.e., highest firing strength) and assigns the data to the class associated to the rule. Fuzzy sets are associated with terms that appear in the antecedents or consequents of rules, and possibly with the inputs and outputs. One advantage of fuzzy classifiers based on "IF-THEN" rules is "transparency" or "interpretability", i.e., the end user (expert) is able to verify the classification paradigm by judging the plausibility, consistency or completeness of the rule-base in fuzzy if-then classifiers.

There are a variety of different models of classifiers like Mamdani-Assilian (MA) model that uses fuzzy sets in the consequent part of the rules. MA model

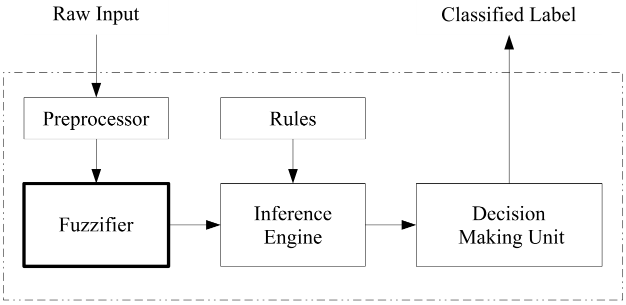


Figure 1.7: The structure of a fuzzy rule-based classifier.

has the following type of rules

$$\begin{aligned} R_k: \quad & \text{IF } x_1 \text{ is } A_{1,i(1,k)} \text{ AND } x_2 \text{ is } A_{2,i(2,k)} \text{ AND } \ldots \text{ AND } x_{\mathrm{n}} \text{ is } A_{n,i(n,k)} \\ & \text{THEN } y_1 \text{ is } B_{o(1,k)} \text{ AND } \ldots \text{ AND } y_c \text{ is } B_{o(c,k)} \quad \text{for } k = 1, \ldots, M \end{aligned}$$

On the other hand, Takagi-Sugeno-Kang (TSK) model allows a (linear) function of the inputs in the consequent part of the rule. Rule in TSK model has the following form

$$\begin{aligned} R_k: \quad & \text{IF } x_1 \text{ is } A_{1,i(1,k)} \text{ AND } x_2 \text{ is } A_{2,i(2,k)} \text{ AND } \ldots \text{ AND } x_{\mathrm{n}} \text{ is } A_{n,i(n,k)} \\ & \text{THEN } y = f_k(x) \quad \text{for } k = 1, \ldots, M \end{aligned}$$

where $f : \Re^n \rightarrow \Re^c$ is a vector function of input x with c components. In fuzzy control, usually the output variables are independent, and a Multiple-Input-Multiple-Output (MIMO) model can be decomposed as a collection of Multiple-Input-Single-Output (MISO) models, which are significantly easier to handle. In pattern classification, the classes (corresponding to the outputs) are not independent-conversely, they are dependent, usually mutually exclusive.

Consider a general fuzzy if-then classifier model:

$$\begin{aligned} R_k: \quad & \text{IF } x_1 \text{ is } A_{1,i(1,k)} \text{ AND } x_2 \text{ is } A_{2,i(2,k)} \text{ AND } \ldots \text{ AND } x_{\mathrm{n}} \text{ is } A_{n,i(n,k)} \\ & \text{THEN } g_{k,1} = z_{k,1} \text{ AND } \ldots \text{ AND } g_{k,c} = z_{k,c} \quad \text{for } k = 1, \ldots, M \end{aligned}$$

The values $z_{k,j} \in \Re$ are interpreted as "support" for class w_j given by rule R_k if the premise part is completely satisfied. There are four types of fuzzy classification systems depending on the consequent as suggested by Cordón [10] and Kuncheva [2]

1. Fuzzy rules with a class consequent, e.g.,

   $R_k : \ldots$ THEN class is $\omega_{o(k)}$

   where $o(k)$ is the output indicator function giving the index of the class associated with rule R_k. For example, this could be translated to a c-dimensional binary output vector with 1 at $o(k)$ and 0, elsewhere.

2. Fuzzy rules with a class and a certainty degree in the consequent, e.g.,

   $R_k : \ldots$ THEN class is $\omega_{o(k)}$ with $z_{k,o(k)}$

   This corresponds to $g_{k,1} = 0$ AND$\ldots$AND $g_{k,o(k)} = z_{k,o(k)}, \ldots,$ AND $g_{k,c} = 0$. In fuzzy terminology, the output is a possibly subnormal singleton over Ω.

3. Fuzzy rules with certainty degrees for all classes in the consequent, i.e., the general model, where $z_{k,i}$ are certainty degrees, typically in the interval [0,1].

4. Fuzzy rules with linguistic labels for the c outputs

   $R_k : \ldots$ THEN $g_{k,1}$ is $B_{o(1,k)}$ AND $\ldots$ AND $g_{k,c}$ is $B_{o(c,k)}$

   where $B_{o(i,k)}$ are linguistic labels defined over a set of certainty values, e.g., the interval [0,1].

The first three groups belong to TSK system model whereas the fourth one belongs to MA system model. More specifically, TSK classifier has been divided into 5 types depending on the types of conjunction (AND connective), A_t , and the calculation of outputs. The firing strength of the rule R_k is as

$$\tau_k(x) = A_t \left\{ \mu_{1,i(1,k)}(x_1), \ldots, \mu_{n,i(n,k)}(x_n) \right\} . \tag{1.13}$$

1. The TSK 1 classifier is derived from the generic TSK model by specifying

   - $z_{k,i} \in \{0,1\}\,,\ k = 1, \ldots M,\ i = 1, \ldots, c,\ \sum_{i=1}^{c} z_{k,i} = 1$; (crisp labels)
   - A_t is minimum
   - The ith output is

$$g_i^{TSK1}(x) = \max_{k=1}^{M} \{ z_{k,i}.\tau_k(x) \} = \max_{k=1}^{M} \left\{ z_{k,i}. \min_{j=1}^{n} \left\{ \mu_{j,i(j,k)}(x_j) \right\} \right\} \tag{1.14}$$

2. The TSK 2 classifier is derived from the generic TSK model by specifying

   - $z_{k,i} \in \Re,\ k = 1, \ldots M,\ i = 1, \ldots, c$
   - A_t is product
   - The ith output is

$$g_i^{TSK2}(x) = \frac{\sum_{k=1}^{M} z_{k,i}.\tau_k(x)}{\sum_{k=1}^{M} \tau_k(x)} = \frac{\sum_{k=1}^{M} z_{k,i}. \prod_{j=1}^{n} \mu_{j,i(j,k)}(x_j)}{\sum_{k=1}^{M} \prod_{j=1}^{n} \mu_{j,i(j,k)}(x_j)} \tag{1.15}$$

3. The TSK 3 classifier is derived from the generic TSK model by specifying

   - $z_{k,i} \in \{0,1\}\,,\ k = 1, \ldots M,\ i = 1, \ldots, c,\ \sum_{i=1}^{c} z_{k,i} = 1$; (crisp labels)
   - A_t is product
   - The ith output is

$$g_i^{TSK3}(x) = \max_{k=1}^{M} \{ z_{k,i}.\tau_k(x) \} = \max_{k=1}^{M} \left\{ z_{k,i}. \prod_{j=1}^{n} \left\{ \mu_{j,i(j,k)}(x_j) \right\} \right\} \tag{1.16}$$

4. The TSK 4 classifier is derived from the generic TSK model by specifying

   - $z_{k,i} \in \{0,1\}\,,\; k = 1,\dots M,\; i = 1,\dots,c,\; \sum_{i=1}^{c} z_{k,i} = 1$; (crisp labels)

   - A_t is product

   - The ith output is

$$g_i^{TSK4}(x) = \frac{\sum_{k=1}^{M} z_{k,i}.\tau_k(x)}{\sum_{k=1}^{M} \tau_k(x)} = \frac{\sum_{k,z_{k,i}=1} \prod_{j=1}^{n} \mu_{j,i(j,k)}(x_j)}{\sum_{k=1}^{M} \prod_{j=1}^{n} \mu_{j,i(j,k)}(x_j)} \tag{1.17}$$

5. The TSK 5 classifier is derived from the generic TSK model by specifying

   - $z_{k,i} \in [0,1]\,,\; k = 1,\dots M,\; i = 1,\dots,c,$ (soft labels)

   - A_t is product

   - The ith output is

$$g_i^{TSK5}(x) = \max_{k=1}^{M} \{z_{k,i}.\tau_k(x)\} = \max_{k=1}^{M} \left\{ z_{k,i}. \prod_{j=1}^{n} \{\mu_{j,i(j,k)}(x_j)\} \right\}. \tag{1.18}$$

1.3.2.2 Non fuzzy rule-based classifier

Fuzzy rule-based and non fuzzy rule-based classifiers complement each other to form the whole framework of fuzzy pattern classification. Since there are numerous types of non fuzzy rule-based classifiers, thus only a few important literature reviews are presented.

Watada et al. [11] suggested a general fuzzy discriminant analysis. The input $\tilde{\mathbf{x}}$ of the classifier is not a point in $\Re^n$ but a set of n fuzzy numbers $\tilde{x}_1, \dots, \tilde{x}_n$, one for each feature. Each element of the data set is also a set of n fuzzy numbers on the feature axes. The discrimination functions are implemented via fuzzy arithmetic. The fundamental idea of this method is that given the input $\tilde{\mathbf{x}}$, each discrimination function $g_i(\tilde{\mathbf{x}})$ is a fuzzy set defined on the interval [0,1], expressing how confident the decision is for the respective class. Each classifier could be defuzzified and the crisp values are compared. Other methods that compare fuzzy sets can also be adopted. While using fuzzy numbers to model the features seems a reasonable choice, the difficulty in computing the fuzzy arithmetic has limited the application of this discriminant model to practical problems. Keller and Hunt [12] introduced the concept of fuzzy perceptron which employs a linear discrimination function $g : \Re^n \longrightarrow \Re \equiv \mathbf{w}^T\mathbf{x}^a$ to distinguish between two classes ω_1 and ω_2 where $\mathbf{w} \in \Re^{n+1}$ is a real-valued vector and $\mathbf{x}^a$ is the augmented vector $[\mathbf{x}^T, 1] \in \Re^n \cup \{1\}$. The training procedure starts with a random weight and updates it iteratively when there is an error in classifying data $\mathbf{z}_j$ via:

$$\mathbf{w} \longleftarrow \mathbf{w} + |l_1(\mathbf{z}_j) - l_2(\mathbf{z}_j)|^m \eta \mathbf{z}_j \tag{1.19}$$

where $l_i(\mathbf{z}_j)$ is the soft label of $\mathbf{z}_j$ in the class ω_i, $i = 1, 2$, η is a constant and m is a parameter, usually $m > 1$. It has been shown that this algorithm can converge for linearly separable classes. The applicability of this model to the case of linearly nonseparable classes is questionable. These branches of fuzzy classifier have not attracted much attention from the researchers probably because fuzzy linear classifiers do not offer a significant benefit over nonfuzzy classifiers and they are not as flexible as fuzzy if-then classifiers.

Fuzzy relational classifier is another type of non if-then classifier which is based on fuzzy relations. This model is useful when the features take a small number of discrete values. As a result, instead of $\Re^n$, a finite feature subspace ς is considered.

More specifically, the fuzzy relation is defined as:

$$R : \varsigma \times \Omega \longrightarrow [0,1] \tag{1.20}$$

where Ω is class label. In the simplest case, R is a look-up table. To calculate the output of the relational classifier, the input $\mathbf{x} \in \Re^n$ is transformed into a input fuzzy set f_ς over ς. The degree of membership participating in $f_\varsigma(\varsigma)$ is calculated by taking the minimum between the n-degrees of membership participating in ς. f_ς can be regarded as a fuzzy relation γ on a universal set with $1 \times |\varsigma|$ elements $f_\varsigma(\varsigma)$. Then the soft class label of $\mathbf{x}$ is obtained by composition:

$$\mu(\mathbf{x}) = \gamma \circ R \tag{1.21}$$

where $\circ$ is usually a max-min composition. Using the maximum membership rules, $\mathbf{x}$ will be assigned a class label ω_i.

Nearest neighbour classification techniques classify an unknown sample by comparing it to its nearest neighbours among a set of known samples. The distance metric used is irrelevant, as long as it applies consistently to all samples in the set. In fuzzy K-nearest neighbours (k-nn) algorithm, an unknown sample's membership in each class is assigned as its K-nearest known neighbours' memberships in those classes, divided by a function of the neighbours' distances from the unknown sample. In essence, $\mathbf{x}$'s membership in class i is given as:

$$\mu_i(x) = \frac{\sum\limits_{j=1}^{K} \mu_{ij} \left(\|x - x_j\|^{\frac{2}{m-1}} \right)^{-1}}{\sum\limits_{j=1}^{K} \left(\|x - x_j\|^{\frac{2}{m-1}} \right)^{-1}} \tag{1.22}$$

Although this algorithm produce more accurate results than crisp k-nn algorithm, it suffers from the drawback where the value of m, used to scale the effect of the distance between $\mathbf{x}$ and its j-th neighbour, $\mathbf{x}_j$, is entirely arbitrary.

The ever popular Fuzzy C-means clustering algorithm [7] also falls into non fuzzy rule-based classifier category. The algorithm is based on minimisation of the following objective function:

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} d_{ij}^{2} \tag{1.23}$$

where $m > 1$ is called the fuzzifier or weighting exponent, u_{ij} is the degree of membership of $\mathbf{x}_j$ in the cluster i, $\mathbf{c}_i$ is the ith cluster center. Fuzzy partitioning is carried out through an iterative optimisation of the objective function shown in Equation (1.23), with the update of membership u_{ij} and the cluster centers $\mathbf{c}_i$ by:

$$u_{ij} = \frac{1}{\sum\limits_{l=1}^{C} \left(\frac{d_{ij}^{2}}{d_{lj}^{2}} \right)^{\frac{1}{m-1}}} \tag{1.24}$$

$$\mathbf{c}_i = \frac{\sum\limits_{j=1}^{N} u_{ij}^{m} \mathbf{x}_j}{\sum\limits_{j=1}^{N} u_{ij}^{m}} \tag{1.25}$$

Other variants of fuzzy clustering algorithms can be found in [13].

1.4 Literature Review on Fuzzy Pattern Classification

This section presents a survey of literature pertinent to the application of fuzzy logic to pattern classification. Since the conventional singleton type-1 fuzzy classifier has been well studied, the review presented here will be focus on the extensional fuzzy classifiers which are still not extensively researched. Specifically, research on non-singleton fuzzy classifiers and type-2 fuzzy classifiers will be reviewed. In addition, the review of the fuzzy classifier learning methods such as rules generation and tuning of rule base parameters are presented. The potential limitations of the existing work are discussed to highlight the rationale for further analysis. Since this thesis consists of multiple topics, a more detailed literature review associated with each topic will be presented in the subsequent chapters.

1.4.1 Non-singleton fuzzy classifiers

A non-singleton fuzzy logic system (NSFLS) is a fuzzy logic system whose inputs are modeled as fuzzy sets, be it type-1 or type-2 fuzzy sets. Mouzouris and Mendel [14] suggested that singleton fuzzy logic system may not adequate to handle uncertainty in the inputs. In the literature, studies on using non-singleton inputs in fuzzy pattern classification are still limited. Hayashi et al. used a vector of type-1 fuzzy sets to train a fuzzy neural network and as inputs during processing [15]. However, the generalization of this method is questionable as it is largely based on heuristic. There is no closed-form expressions for the fuzzy logic systems [14]. Wei and Mendel [16] proposed a non-singleton fuzzy classifier for signal modulation classification by using an addictive fuzzy logic system as a core building block. Each complex data is modeled as a two-dimensional membership function which is characterised by exponential kernel and Hamming distance metric. The simulations show that when ideal conditions hold, the Maximum-Likelihood and fuzzy logic classifiers perform equally but when impulsive noise is present the fuzzy logic classifier performs consistently better. Nevertheless, the authors pointed out that the drawback of their non-singleton fuzzy classifier is that there is no performance analysis for it. Besides, the author also admitted that non-singleton fuzzy classifier reduces to singleton classifier when both the input and antecedent use Gaussian membership function. Wu and Mendel [17] used interval type-2 fuzzy sets to model the uncertain acoustic inputs in ground vehicles classification problem. However, they did not benchmark the performance of non-singleton fuzzy classifier against singleton counterpart. Therefore, the advantage of non-singleton fuzzy classifier over singleton fuzzy classifier is still unclear. In view of the limited previous studies on non-singleton fuzzy classifier, it is highly imperative to find out how a non-singleton fuzzy classifier can handle the noise in inputs more efficiently. In particular, it is important to understand the underlying characteristics of non-singleton fuzzy classifier.

1.4.2 Type-2 fuzzy classifiers

Although type-2 fuzzy sets were originally introduced by Zadeh in 1975, they did not receive much attention from researchers until the field is popularised by Mendel et al. in 1999 through a series of publications. Since type-1 fuzzy logic is most successful in control, to many researchers, it is a natural progress to extend type-1 fuzzy control to type-2 fuzzy control due to the capability of type-2 fuzzy logic system to handle uncertainties. Subsequently, type-2 fuzzy control becomes the most successful application of type-2 fuzzy logic system. On the contrary, type-2 fuzzy pattern classification has not been given adequate attention. The research in this area is still very limited. John et al. [18] represented consultant's interpretation of the radiographic tibia images by type-2 fuzzy sets and classified the images using neuro-fuzzy clustering technique. Liang and Mendel [19] combined type-2 fuzzy sets with type-1 fuzzy logic system to classify video traffic using compressed data. Mitchell [20] investigated the similarity measure for measuring the

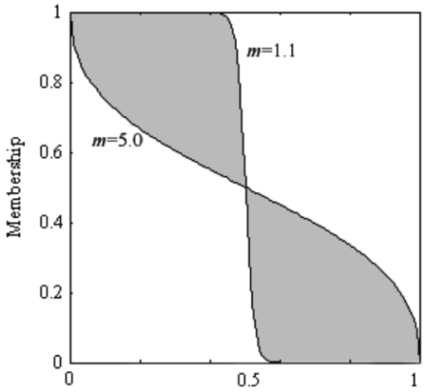


Figure 1.8: Footprint of uncertainty (shaded area) of an interval type-2 fuzzy set FCM.

similarity, or compatibility between two type-2 fuzzy sets. The similarity measure can be used to formulate the classification problems in pattern recognition. Zeng and Liu incorporated type-2 fuzzy sets with Hidden Markov Model and Markov random fields for speech and handwritten Chinese character recognition [21, 22]. Similarly, Zeng et al. extended Gaussian mixtures models by using type-2 fuzzy sets [23]. One major drawback of Zeng et al. methods is that fuzzy set theory is not exploited whereby type-1 membership functions are simply replaced with type-2 membership functions. As a result, the capability of type-2 fuzzy logic system to handle the uncertainties may not be fully utilised. Next, Hwang and Rhee [24] focused on the uncertainty associated with the fuzzifier parameter m that controls the amount of fuzziness of the final partition in the fuzzy C-means (FCM) algorithm. Interval type-2 fuzzy set is used to form two fuzzifiers m_1 and m_2 which creates a footprint of uncertainty (FOU) for the classifier m. This is illustrated in Fig. 1.8. Center-updating and hard partition was modified in the FCM by incorporating interval type-2 fuzzy sets and the results show the effectiveness of this method. Wu and Mendel designed type-2 fuzzy rule-based classifiers for classification of battlefield ground vehicles [17]. Besides the antecedents, the uncertain acoustic features were also modelled as interval type-2 fuzzy sets. Nevertheless, the results show that there is no substantial improvement of type-2 fuzzy classifiers over type-1 counterparts. In particular, it is unclear in what ways (error rate, generalizability, robustness) type-2 fuzzy classifiers can perform better. Besides, it is generally accepted that type-2 fuzzy classifier is much more computationally demanding than type-1 fuzzy classifier. Thus, it is highly commendable to find out if it is worth making use of type-2 fuzzy classifier instead of type-1 fuzzy classifier at the cost of increased computational requirement.

1.4.3 Learning of fuzzy classifiers

Earlier research on fuzzy pattern classification have been focused on the generation of fuzzy if-then rules from numerical data. In general, this problem consists of two phases: (a) fuzzy partitioning of a feature space into fuzzy subspaces and (b) determination of fuzzy if-then rules corresponding to the fuzzy subspaces. For instance, Ishibuchi et al. [25] generated fuzzy if-then rules from the training samples by employing a fuzzy partitioning approach with fuzzy grids. One shortcoming of such an approach is that the number of fuzzy subspaces increases exponentially as the number of features is increased. With the aim of tackling this shortcoming, they proposed another approach subsequently which is based on sequential subdivision of the fuzzy subspaces (of different sizes) [26]. Nevertheless, the dimension of the feature spaces is still very high. Other works in this direction includes references [27],[28], where Mandal et al. decomposes the feature space into some overlapping subspaces using geometric structure of the pattern classes found from the training samples. Pal and Mandal [29] described an approach where a feature space is decomposed into a few (3^N for N features) overlapping regions by considering three primary linguistic properties (small, medium and high) along each of the feature axes. Next, Abe and Lan [30] proposed a method which extracts fuzzy rules with variable fuzzy regions by recursively resolving overlap between two classes. Similar to [25], the number of regions generated becomes very high if the number of features is large. This leads to difficulty in implementing the algorithm. Subsequently, Ishibuchi et al. [31] demonstrated a fuzzy classifier system where the antecedent fuzzy sets of each fuzzy if-then rule are prespecified linguistic values with fixed membership functions and the consequent class and the grade of uncertainty are determined by a simple heuristic procedure. The authors claimed that their fuzzy classifier which is based on simple fuzzy grids is able to work well for problems with more than ten continuous attributes. Although the classifier has high comprehensibility, the number of fuzzy if-then rules is not minimised. Apart from that, Nauck and Kruse [32] showed that the use of rule weights (i.e., certainty grades) can be viewed as the modification of membership function in fuzzy reasoning. Following that, Ishibuchi and Nakashima [33] examined the effect of rule weights in fuzzy rule-based classification systems. The effect of rule weights is illustrated by showing the classification boundary with/without certainty grades. Fig. 1.9 shows the decision area of each fuzzy if-then rule with various combinations of certainty grades. It is shown that the certainty grades play an important role when a fuzzy rule-based classification system is a mixture of general rules and specific rules. Computer simulations showed that high classification performance can be achieved without modifying the membership functions when fuzzy if-then rules with certainty grades are used. The certainty grades have been interpreted

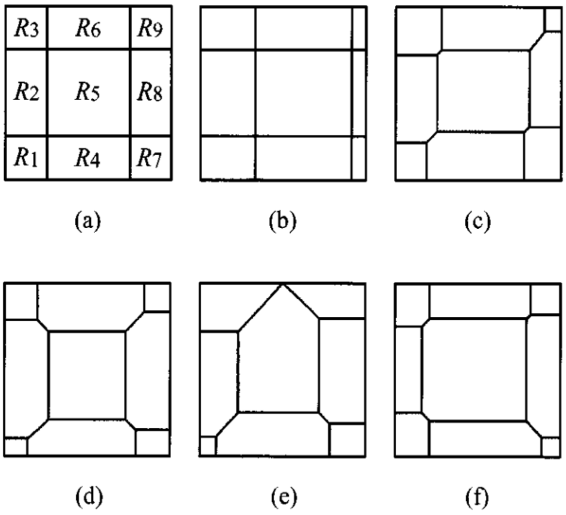


Figure 1.9: Classification area of each fuzzy if-then rule with a different certainty grade (weight).

as the rule strength because they affect the size of the decision area. Although the literature on fuzzy partitioning of feature space has shed some light on how to generate fuzzy rules from the given data, all the aforementioned fuzzy partitioning methodologies are highly heuristic in nature. The rules generated may not be compact enough (redundancy exists) and the parameters of the fuzzy membership functions are not optimum. This leads to the research on fuzzy classifier design.

Many approaches have been adopted to design a fuzzy classifier which can be categorised into three general groups. The first group tunes only the consequents part. In this case, the rule base is fixed and the membership functions of the antecedents are determined a prior. Two popular methods to tune the consequent parts are least-squares [34, 35, 36] and backpropagation algorithms [37]. A serious drawback of least-squares method is that there is no clear procedure on how to choose the parameters of the fuzzy membership functions and the number of rules. Likewise, backpropagation method is sensitive to parameters initialisation and how to choose the number of rules is left as an open issue. Next, rules pruning falls under the second group of design methods. The antecedent membership functions and consequents are fixed and optimisation algorithm such as GA is used to select the best subset of rules and discharges the rests [2]. The last group is a combination of the first and second methodology. The system is built and tuned by simultaneous adaptation of the rule-base and the antecedent membership functions. Again, Genetic Algorithm (GA) has been successfully applied to learn both the antecedent and consequent part with both fixed and varying number of rules [2, 38, 31]. The popularity of GA is attributed to its two great advantages over back-propagation approach. Firstly, the functions that can be used in GAs can be much more general in nature and knowledge of the gradient of the functions is not required. Secondly, GAs are less likely to be trapped in local minima because they explore the solution spaces in multi directions at the same time. While GA is suitable for many optimisation problems, the algorithm is based on empirical risk minimisation, which is to minimise the training error. This could limit the generalisation capability of a classifier when over-fitting issue occurs. Therefore, it would be advantageous to design a hybrid learning algorithm which is also based on structural risk minimisation and inherits the advantage of GA.

1.5 Aims and Scope of the Work

The literature review shows that conventional singleton type-1 fuzzy logic systems were mostly applied to solve pattern classification problems due to their simplicity. However, research has shown that type-1 fuzzy classifier is not sufficient to handle uncertainties. In particular, singleton input is unable to reflect the uncertainty in the data while type-1 fuzzy membership function fails to model the uncertainty in the membership function itself. Based on the literature review, some research gaps in the field of fuzzy logic pattern classification are summarised below:

- There are few studies on non-singleton fuzzy classifiers. Current implementation of the non-singleton fuzzy classifier is reduced to singleton fuzzy classifier when both the input and antecedent use Gaussian membership function. In addition, the advantage of non-singleton fuzzy classifier over singleton fuzzy classifier is still unclear.

- Although type-2 fuzzy logic system has shown promising results in control applications, the amount of research in type-2 fuzzy classifier is very limited. Research has shown that the performance of type-2 fuzzy classifier may not be better than type-1 counterpart although type-2 fuzzy classifier is more computationally demanding.

- The majority of learning methods for fuzzy rule-based classifier are based on heuristics. In particular, how to optimise a fuzzy logic system is still an open question. Choosing rules, membership functions, and operators are in general still done by trial and error. Even with automated learning algorithms such as genetic algorithm and back-propogration, the learning of membership function parameters is still relied on empirical risk minimisation, which is to reduce the training error. The disadvantages are that the generalisation of the classifier is not guaranteed and the algorithm such as back-propagation may get trapped at local minima.

The main aim of this research is to seek a better understanding of the properties of extensional fuzzy rule-based classifiers (FRBCs). These include non-singleton FRBC, interval type-2 FRBC, and fuzzy hybrid classifiers. Moreover, this research aimed at systematising the learning procedure for fuzzy rule-based classifier. The specific objectives were:

1. To investigate if non-singleton type-1 FRBC can better handle the input uncertainties compared to singleton counterpart.

2. To examine the efficacy of interval type-2 FRBC and to study its robustness in three aspects:

    - robustness towards noisy unseen samples;

    - robustness against selected input features; and

    - robustness to randomness in classifier designs.

3. To develop an efficient learning algorithm based on support vector machines and fuzzy C-means algorithms for fuzzy rule-based classifiers.

4. To improve the performance of conventional k-nearest neighbors classifier when the training data is insufficient by introducing a fuzzy rule base to initiate prototypes' membership values.

5. To verify the applicability of fuzzy rule-based classifier in solving practical classification problem such as induction motor fault diagnosis.

The results of this present study may shed light on how to manage the input uncertainties using non-singleton fuzzy rule-based classifier in which the conventional statistical classifiers still fall short in this aspect. Moreover, the comparative analysis between type-1 and type-2 FRBCs may contribute to a better insight of the underlying characteristics and advantages of type-2 FRBCs. More importantly, the findings may help in deciding the choices of classifiers through the understanding of the tradeoffs between the computational speed and classification accuracy, depending on the types of applications. Furthermore, the proposed hybrid learning algorithm could be a valuable tool to design and optimise the fuzzy rule-based system when the conventional methods fail. Finally, the incorporation of fuzzy rule-based initialisation procedure into fuzzy k-nn algorithm should open up new ideas to enhance other existing classification algorithms that lack capability to handle the uncertainties.

The thesis focuses more on the experimental study of fuzzy rule-based classifier to solve practical real-world problems. Theoretical study is not central to this thesis because up to now there is no closed-form mathematical equation to model the ad-hoc architecture of fuzzy rule-based classifiers. In addition, it should be noted that the type-2 FRBC implemented throughout the study belongs to the interval type-2 FRBC. General type-2 FRBC is not considered in this study because it is computationally intensive due to the complexity of type reduction.

1.6 Organisation of the Thesis

The thesis is organised as followed. Chapter 2 aims at analysing a non-singleton fuzzy rule-based classifier (NSFLC) and assessing its ability to cope with uncertainties in pattern classification problems. The analysis demonstrate that the

NSFLC has fuzzy classification boundary and noise suppression capability. These characteristics means that the NSFLC is particulary suitable for problems where the boundaries between classes is non-distinct. To further demonstrate the benefits offered by a NSFLC, a non-singleton fuzzy logic classifier evolved using genetic algorithm (GA) was assessed using a benchmark cardiac arrhythmias classification problem. Chapter 3 introduces an interval type-2 fuzzy rule-based classifier. Three strategies for designing the footprint-of-uncertainty (FOU) were proposed to generate membership functions (MFs) that reflect the underlying data. Similar to Chapter 2, interval type-2 fuzzy classifier was applied to the classification of cardiac arrhythmias. However, different sources of noises have been included to model the uncertainties associated with the vagueness in MFs and the unpredictability of the data. A comparative robustness analysis of type-1 and type-2 fuzzy rule-based classifiers is given in Chapter 4. The experiment were carried out on both the synthetic data set and real data set. In Chapter 5, a hybrid fuzzy rule-based classifier is proposed. Fuzzy c-means clustering and genetic algorithm were used to optimise the number of rules and antecedent parameters. By using the relationship between a support vector machine (SVM) and a Takagi-Sugeno-Kang (TSK) fuzzy logic system, an efficient method for learning the consequent parts of the TSK fuzzy system is introduced. Chapter 6 presents an improved version of fuzzy k-nn classifier with fuzzy rule-based initialisation procedure to address the generalisation problem when there is lack of complete training data set. Practical application of fuzzy rule-based classifier for inverter-fed induction motor fault detection is presented in Chapter 7. Finally, Chapter 8 gives the conclusion remarks of the thesis and suggestions for further work.

Chapter 2

Non-Singleton Fuzzy Rule-Based Classifier: Handling the Input Uncertainty

The choice of feature vector is perhaps the most difficult task in pattern classification. A good feature subset selected from a vast number of possibilities will lead to more accurate classifications, as well as better understanding and interpretation of the data [39]. Inclusion of irrelevant and redundant features generally adversely affect the performance of almost all common machine learning or pattern classification algorithms. Besides classification accuracy, the relative ease with which the features may be extracted is another important factor to consider. Computation costs may be lowered by performing pattern classification using features that are easier to extract. However, these features are likely to contain more uncertainties which could lead to misclassifications. Furthermore, it is common to encounter noise corrupted signals that are caused by electrical interferences within the feature extraction devices or the mishandling of the equipment. Medical diagnosis is an example of a pattern recognition problem where feature selection is a very delicate process since a human life is often at stake. A framework, such as fuzzy logic theory, that systematically represents the uncertainties present in the input signals may reduce the dependence of classification accuracy on the choice of feature vector.

The objective of this chapter is to investigate if a non-singleton fuzzy rule-based classifier (NSFRBC) can better cope with the uncertainties or fuzziness present in the extracted features. This work is motivated by the observation that there is often a trade-off between classification accuracy and the complexity of the algorithm for extracting features. A NSFRBC, designed to capture the uncertainties in the extracted feature, will be especially useful when the fuzziness of the input data is inevitable. The rest of this chapter is organised as follows. In Section 2.1 and 2.2, the non-singleton fuzzy classifier is defined and analysed. Section 2.3 presents the results of applying NSFLC to the practical problem of cardiac arrhythmias classification. This section includes the background information on arrhythmias, feature extraction, fuzzy logic implementation, GA evolution of the fuzzy classifier, experimental results and discussion. Section 2.4 concludes this chapter.

2.1 Non-Singleton Fuzzy Rule-Based Classifier (NSFRBC)

The structure of a fuzzy rule-based classifier is shown in Fig. 1.7. The role of the fuzzifier in a fuzzy system is to map an element, x'_i, in the input vector $\mathbf{x}' = (x'_1, \ldots, x'_m)^T \in X_1 \times X_2 \times \cdots \times X_m \equiv \mathbf{X}$ into the fuzzy set $\widetilde{X}'_i$. This process provides a natural framework for handling uncertain input information. There is a variety of methods for performing fuzzification. The most common approach is singleton fuzzification (Fig. 2.1a), which maps a crisp input into the following membership function :

$$\mu_{\widetilde{X}'_i}(x_i) = \begin{cases} 1 & x_i = x'_i \\ 0 & x_i \neq x'_i \qquad \forall x_i \in \mathbf{X} \end{cases}$$

By mapping the crisp signal into a fuzzy set that has only a single point in its support, the singleton fuzzifier does not model any vagueness in the input. As singleton fuzzification is employed in most existing fuzzy classification schemes, they do not make full use of the modeling capability of the fuzzifier. To better cope with noisy, imprecise or inaccurate input signals [35, 40], the classifier proposed in this chapter comprises a non-singleton fuzzifier. By definition, a non-singleton fuzzifier maps a measurement x'_i into a fuzzy number with an associated membership function such as the Gaussian function:

$$\mu_{\widetilde{X}'_i}(x_i) = \exp\left[-\frac{1}{2}\left(\frac{x_i - x'_i}{\sigma_{\widetilde{X}'_i}}\right)^2\right]$$

where the variance ${\sigma_{\widetilde{X}'_i}}^2$ defines the width (spread) of the Gaussian input fuzzy set, $\widetilde{X}'_i$. As $\mu_{\widetilde{X}'_i}(x'_i) = 1$ and $\mu_{\widetilde{X}'_i}(x_i)$ decreases from unity as x_i moves away from

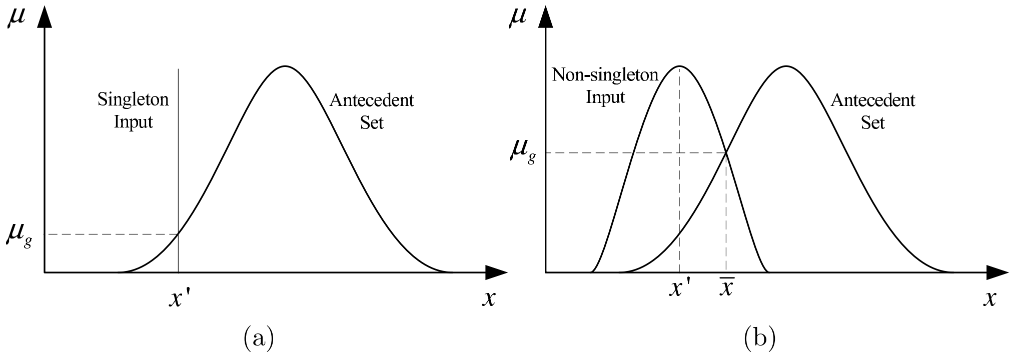


Figure 2.1: Input and antecedent operation for different types of inputs. (a) Singleton and (b) Non-singleton.

x_i' [41], the non-singleton fuzzifier implies that the given input value x_i' is the most likely value to be correct one from all the values in its immediate neighbourhood [14]. When signals are corrupted by noise, adjacent points may also be correct values, but to a lesser degree. A larger spread indicates that more uncertainties are inherent within the data. Fig. 2.1(b) shows how the output of the non-singleton fuzzifier is combined with the antecedent set, $\widetilde{F}_i$, using sup-star composition. The result of the fuzzy inference process is:

$$\mu_g(\bar{x}_i) = \sup\{\mu_{\widetilde{X}_i'}(x_i) \star \mu_{\widetilde{F}_i}(x_i)\} \tag{2.1}$$

where '$\star$' denotes t-norm operators such as minimum or product and $\bar{x}_i$ denotes the value of x_i at which the supremum of (2.1) occurs. μ_g can be interpreted as the degree of consistency of the input fuzzy set, $\widetilde{X}_i'$ with the antecedent fuzzy set, $\widetilde{F}_i$. If the fuzzified input set is defined as in (2.1) and the antecedent set is defined as:

$$\mu_{\widetilde{F}_i}(x_i) = \exp\left[-\frac{1}{2}\left(\frac{x_i - m_{\widetilde{F}_i}}{\sigma_{\widetilde{F}_i}}\right)^2\right]$$

where the variance $m_{\widetilde{F}_i}$ and ${\sigma_{\widetilde{F}_i}}^2$ defines the mean and width of the Gaussian antecedent fuzzy set, $\widetilde{F}_i$. With minimum t-norm, $\bar{x}_i$ is given as:

$$\bar{x}_i = \frac{\sigma_{\widetilde{X}'_i} m_{\widetilde{F}_i} + \sigma_{\widetilde{F}_i} x'_i}{\sigma_{\widetilde{X}'_i} + \sigma_{\widetilde{F}_i}} \tag{2.2}$$

Under product inference, the value of x_i at which the supremum of (2.1) occurs is:

$$\bar{x}_i = \frac{\sigma^2_{\widetilde{X}'_i} m_{\widetilde{F}_i} + \sigma^2_{\widetilde{F}_i} x'_i}{\sigma^2_{\widetilde{X}'_i} + \sigma^2_{\widetilde{F}_i}}$$

The ‘pre-filtering’ effect of NSFRBC is the key to handle input measurement uncertainty [37]. This effect is the result of the sup-star composition within the NSFRBC framework. The noise suppression capability may be demonstrated in the following example. Assume that x'_i is an input corrupted by noise, that is,

$$x'_i = {x_0}'_i + n_i \tag{2.3}$$

where ${x_0}'_i$ is the useful signal and n_i is the noise. Substituting x'_i in (2.2) with (2.3) yields

$$\bar{x}_i = \frac{\sigma_{\widetilde{X}'_i} m_{\widetilde{F}_i} + \sigma_{\widetilde{F}_i} {x_0}'_i}{\sigma_{\widetilde{X}'_i} + \sigma_{\widetilde{F}_i}} + \frac{\sigma_{\widetilde{F}_i} n_i}{\sigma_{\widetilde{X}'_i} + \sigma_{\widetilde{F}_i}} \tag{2.4}$$

(2.4) shows that the noise is suppressed by the factor $\sigma_{\widetilde{F}_i}/(\sigma_{\widetilde{X}'_i} + \sigma_{\widetilde{F}_i})$. Similarly, it can be proved that the triangular membership function also has this kind of noise suppressing capability.

2.2 Characteristics of Non-Singleton Fuzzy Rule-Based Classifier

Consider a NSFRBC that has R rules of the following structure:

$$\text{IF } x_1 \text{ is } \widetilde{F}^r_{1,j_1} \text{ and } \ldots \text{ and } x_m \text{ is } \widetilde{F}^r_{m,j_m}, \text{ THEN } y^r = \text{Class}^r$$

where $r = 1, \ldots, R$, $i = 1, \ldots, m$, and $j_i = \{1, \ldots, p_i\}$. p_i is the number of fuzzy partitions in X_i. The firing strength of the r^{th} rule is:

$$f_r = T^m_{i=1}\mu_{g_r}(\bar{x}_i)$$

where 'T' is the conjunction type aggregation operator or t-norm operator. The minimum and product are the two most widely used operations. Adopting the winner-takes-all approach, the overall decision is

$$Class\ Label = Class^w; \quad w = \arg\max_r(f_r)$$

The ability of the non-singleton fuzzifier in the NSFRBC to handle vague input signals will be illustrated using a two-feature problem. The antecedent sets of the NSFLC is shown in Fig. 2.2 and the fuzzy rule base is as follows :

IF x_1 is *small* and x_2 is *small*, THEN C_1

IF x_1 is *small* and x_2 is *large*, THEN C_2

IF x_1 is *large* and x_2 is *small*, THEN C_3

IF x_1 is *large* and x_2 is *large*, THEN C_4

where C_i represents the class i. Suppose two input feature vectors with the same crisp values (x_1, x_2) have differing level of uncertainties. In particular, vector 1 is

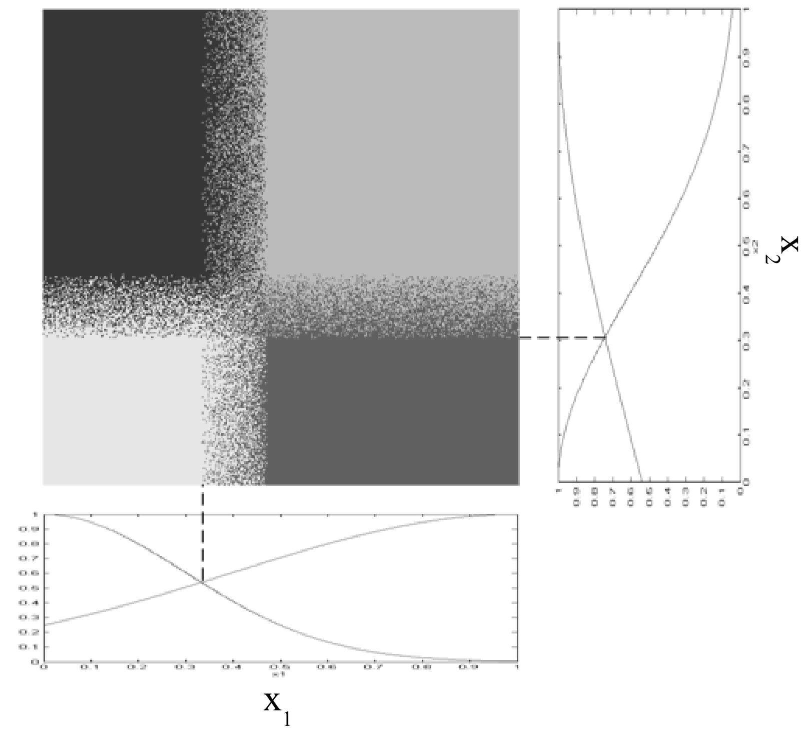
Table 2.1: Firing Strengths of The Example in Section 2.1

	f_{R_1}	f_{R_2}	f_{R_3}	f_{R_4}	Class
Pair 1	0.7408	0.5626	0.6952	0.5280	1
Pair 2	0.4876	0.3699	0.5837	0.4428	3

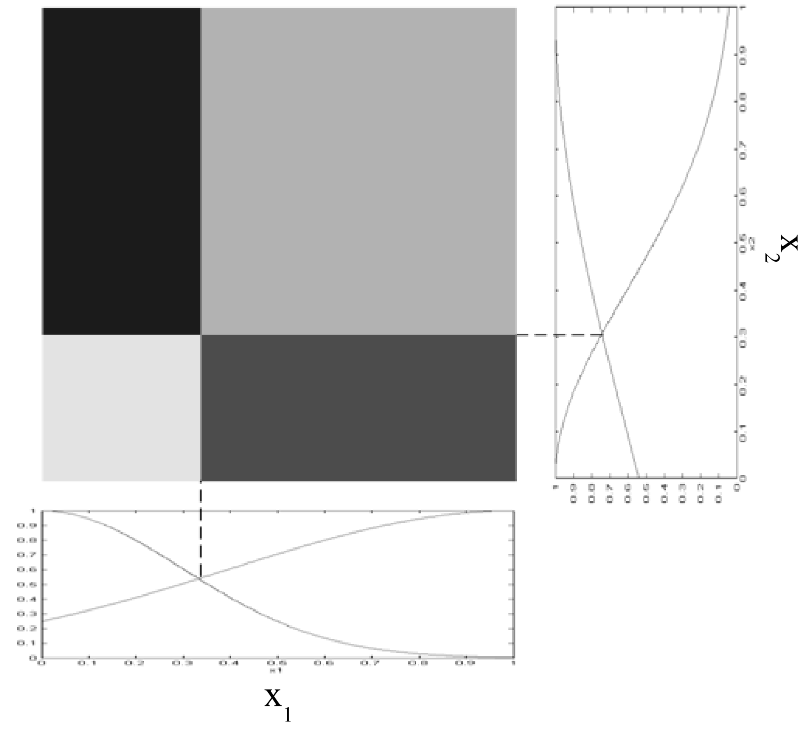
more uncertain than vector 2. Subsequently, the two inputs are fuzzified into the following Gaussian fuzzy membership functions that have the same mean values but different standard deviations :

Vector 1:	Vector 2:
$m_{x_1} = 0.40, \quad \sigma_{x_1} = 0.52$	$m_{x_1} = 0.40, \quad \sigma_{x_1} = 0.19$
$m_{x_2} = 0.31, \quad \sigma_{x_2} = 0.31$	$m_{x_2} = 0.31, \quad \sigma_{x_2} = 0.28$

Table 2.1 tabulates the firing strengths and classification results for two cases. Regardless of whether fuzzy inference is implemented using the product t-norm or the minimum t-norm, input vector 1 is assigned to C_1 while input vector 2 is assigned to C_3. The result demonstrates that the additional information about the variations in the input signal enables the NSFRBC to distinguish between data that have the same crisp input vector. The implication is that NSFRBC is able to produce “fuzzy” boundary, as illustrated by the decision space plot in Fig. 2.2(a). In contrast, a singleton fuzzy rule-based classifier (SFRBC) is only able to produce crisp boundary as shown in Fig. 2.2(b). The fuzzy boundary characteristic of a NSFRBC would be very useful for problems where the various classes do not have clear boundaries.



(a)



(b)

Figure 2.2: Comparison of the classification boundaries produced by (a) non-singleton fuzzy rule-based classifier (NSFRBC) and (b) singleton fuzzy logic classifier (SFRBC). NSFRBC produces fuzzy decision boundary while SFRBC produces crisp decision boundary.

2.3 Application to ECG Arrhythmias Classification

2.3.1 Background Information

An electrocardiogram (ECG) is the representation of the electrical activity of the heart (cardiac) muscle as it is recorded from the body surface. Fig. 2.3 shows the various components of a typical ECG signal. The P wave represents depolarisation of the upper part of the heart, the atria whereas the QRS complex represents ventricular depolarization and T wave represents ventricular repolarisation.

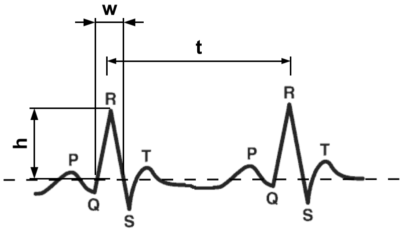


Figure 2.3: ECG components: P wave, QRS complex, and T wave.

Ventricular fibrillation (VF) and ventricular tachycardia (VT) are both life-threatening cardiac arrhythmias. In particular, VF requires immediate defibrillation whereas VT must be distinguished from Normal Sinus Rhythm (NSR) and VF to receive cardioversion, by delivering a shock of somewhat lower energy in synchronization with the heart beat. Critical cardiac incidents occur most often out of hospitals, therefore automatic external defibrillators (AED) were introduced for increasing the survival rate [42]. Since the successful termination of VF and VT requires fast response and application of high-energy shocks in the heart region, the accuracy of the built-in algorithm for VF detection is of paramount importance.

Therefore, the automated diagnosis must match the accuracy of specialists.

In the past, many algorithms based on time or frequency domain, or a combination of time and frequency techniques have been proposed for classifying cardiac arrhythmias. In general, time domain approaches have the advantage over frequency domain approaches because of their computational simplicity while the latter has the advantage of being more reliable in classifying VF and VT. The time domain features are usually known as the "qualitative" features. From the perspective of a medical practitioner, the "qualitative" features such as R-R interval (period) and amplitude of the QRS pulse are generally easier to understand rather than the frequency domain features. Unfortunately, the dynamics of the ECG signal is inherently noisy and therefore the extracted time domain features are uncertain. This is because the recorded ECG signal (especially surface ECG) is very sensitive to cable movement and muscle activity. In addition, the interference from electrical network can further degrade the recording process. Although these artifacts can be reduced by filtering techniques, eliminating them is impossible. Compounding the problem, the feature extraction algorithm may be unreliable or imperfect. In the context of ECG QRS detection, the false detection of a peak is common when the noise amplitude approaches the R wave amplitude. Likewise, the chances that a peak is not detected is high when the peak level is relatively lower than the others. Both problems are common for VF and polymorphic VT signals. Due to the noisy ECG signals and the difficulties in extracting features, the problem of classifying NSR, VF and VT waveforms using time-domain features is suitable for assessing the ability of the NSFLC for handling uncertainties.

2.3.2 Feature Extraction

The ECG data used in this study was obtained from MIT-BIH Malignant Ventricular Arrhythmia Database (VFDB) [43]. All signals were first preprocessed by a 0.05-40Hz bandpass filter and a 60Hz notch filter in order to suppress DC components, baseline drifts and possible electrical interference. The filtered ECG signal were transformed into the binary strings. The transformation algorithm used in this study was an enhanced version of that in the paper [44]. Unlike Zhang's one-pass conversion, a two-pass conversion method was employed. The ECG signal will be transformed into a partial binary string first instead of full binary string directly. This can reduce the false positive peak detection greatly by eliminating low amplitude signal. This step was closely followed by a full binary string conversion for determining a threshold that can maximise the differences between NSR class and VF/VT classes. The feature extraction steps are listed as follows:

1. Select a finite length (i.e., 4s) of ECG. Since the VFDB signals were digitised at 250Hz, then there will be 1000 data points $\{x_i | i = 1, 2, \ldots, n;\ n = 1000\}$ within 4s window length.

2. Mean-center ECG data where the mean data, x_m is subtracted from every data point, i.e., $\{x_i - x_m\}$.

3. Find out the negative peak, V_n and positive peak V_p.

4. Form a partial binary string: if the signal level falls in between the range of $(0 < x_i < 0.2\ V_p)$ or $(0.2\ V_n < x_i < 0)$, then it is assigned "0".

5. Calculate the parameters N_p and N_n. N_p denotes the number of data ($x_i > 0$) while $N_n = n - N_p$.

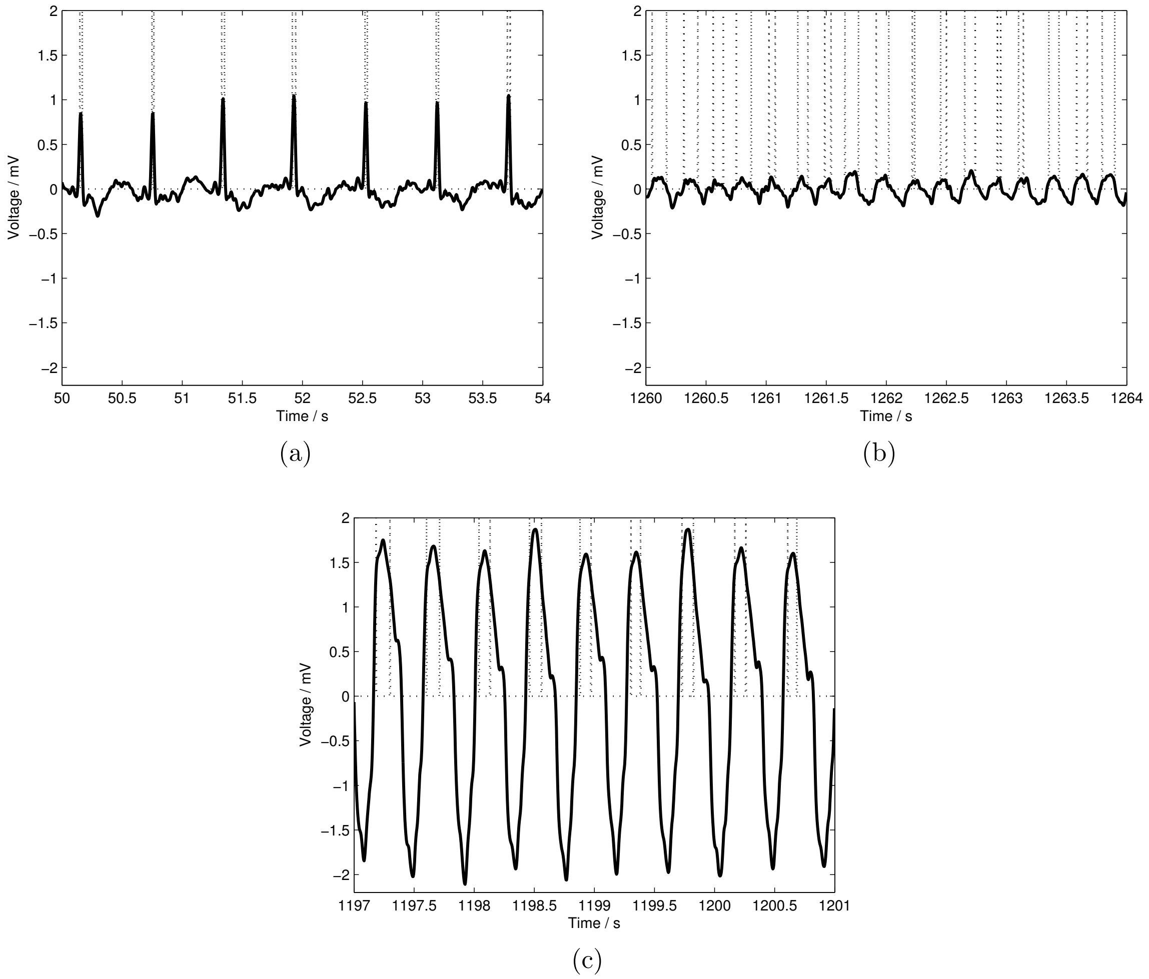
6. Determine a proper threshold, T_r to convert the partial binary string into a complete binary string: if $N_p < 0.15\ n$, then threshold is assigned as $T_r = 0.7\ V_p$, otherwise $T_r = 0$. This step is crucial to separate NSR signals from VF and VT signals.

7. Compare x_i to T_r to turn the partial binary string into a complete binary string, that is if $x_i \leq T_r$, then x_i is assigned as "0" or otherwise "1".

The graphs in Fig. 2.4 show the examples of three different types of ECG signals with their corresponding binary sequences. Three features commonly used for ECG classification were extracted from the binary sequences: *pulse width*, *pulse period* and *peak amplitude.* All parameters were averaged within the 4s window and their standard deviations were calculated accordingly.

The scatter plots of the extracted features are shown in Fig. 2.5. Analysis of the data shows that amplitude feature has less ambiguity, and therefore it is a better input candidate than the period feature. While amplitude information may result a clearer boundary (Fig. 2.5(b)), it is harder to extract as it is sensitive to baseline wanders. On the other hand, period information is easier to be extracted but the classification problem is more challenging because there are overlaps between the VF and VT classes as shown in Fig. 2.5(a). To investigate the ability of the NSFRBC to provide good classification performance using ambiguous data that is easier to extract, a comparative study using the four fuzzy classifiers is performed :

1. SFRBC with *pulse width* and *period* as input features.

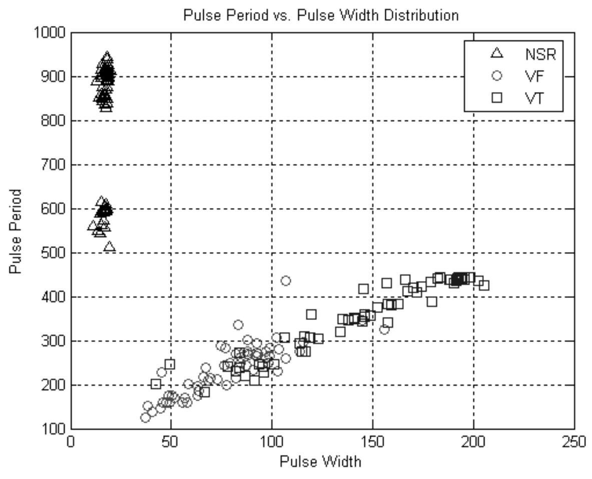


(a)

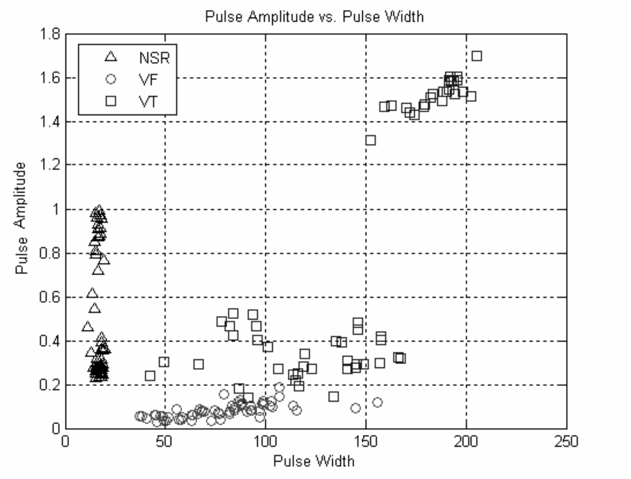
(b)

(c)

Figure 2.4: ECG signals (excerpts from VFDB) and corresponding binary sequences: (a) NSR, record 421 (50-54s), (b) VF, record 424 (1260-1264s), (c) VF, record 611(1197-1201s).



(a)



(b)

Figure 2.5: The scatter plots for inputs. (a) Pulse period vs. width, (b) Pulse amplitude vs. width.

2. SFRBC with *pulse width* and *amplitude* as input features.
3. NSFRBC with *pulse width* and *period* as input features.
4. NSFRBC with *pulse width* and *amplitude* as input features.

2.3.3 Structure of the Fuzzy Classifiers

The fuzzifier transforms the inputs into the fuzzy sets: 'small', 'medium', and 'large'. Each antecedent fuzzy set is characterised by a Gaussian membership function. Gaussian membership function rather than triangular membership function were selected for two reasons. Firstly, the number of parameters to be evolved is smaller since a Gaussian membership function can be represented by two parameters (i.e., mean, m_i and standard deviation, σ_i) only whereas a triangular membership function needs three parameters (i.e., left point, l_i, center point, m_i, and right point, r_i). As a result, the computation time can be reduced. Secondly, Gaussian membership function ensures the fuzzifier is always injective while the injectivity of triangular function fuzzifier is only assured when the sets intersect above a threshold. Injectivity is defined as the discrimination of different values of the input variable, allowing an effective fuzzy processing [45]. In the experiments, the Gaussian membership function used for the non-singleton fuzzifier was symmetrical based on the assumption that the effect of noise is most likely to be equivalent on all points. The universe discourses of inputs were not normalised, but they were set to appropriate ranges according to the input values. Inferring from the data in Fig. 2.5, the pulse period range was set to [100, 1000] ms whereas the peak amplitude range was set to [0, 1.8] mV.

The Takagi-Sugeno-Kang (TSK) [46] classifier was utilised. This was also the structure being adopted by Li *et al.* [47]. A set of nine fuzzy if-then rules were constructed of the form:

$$\text{IF } x_1 \text{ is } \widetilde{F}^r_{1,j_1} \text{ and } x_2 \text{ is } \widetilde{F}^r_{2,j_2}, \text{ THEN Class}^r$$

where $j_1 = j_2 = \{1, 2, 3\}$ (small, medium, large), $r = 1, \ldots, 9$, and $\text{Class}^r = \{1, 2, 3, 4\}$ (CT, NSR, VF, VT). The label CT implies that no decision can be reached for the interval and the patient should be referred for further testing. Since there were nine outputs, the logical products for each rule must be combined or inferred. The "Max-Min" inference engine which takes the firing strengths of each rule and select the highest one was used as explained in Section 2.1.

2.3.4 Classifier Training

This section discusses how the proposed fuzzy based classifier was formulated by using the Genetic Algorithm (GA) approach. The Gaussian membership functions and rule base were initially randomised, and then being tuned simultaneously by GA. Shi et al. [48] suggested that membership functions and rule base should be designed and evolved at the same time since both parameters are said to be co-dependent.

A GA starts off with a population of randomly generated chromosomes, and advances toward better chromosomes by applying genetic operators. The population undergoes evolution in a form of natural selection. During successive iterations, called generations, chromosomes in the population are rated for their adaptation as solutions, and on the basis of these evaluations, a new population of chromosomes is formed using a selection mechanism and specific genetic operators (mutation and crossover). A fitness function is used to return a single numerical fitness of the individual in the population, which is supposed to be proportional to the utility or adaptation of the solution represented by that chromosome.

When designing a fuzzy system using a GA, the first important consideration is the representation strategy, that is how to encode the fuzzy system into the chromosome. In our design, there were two inputs and each of the input fuzzy variables (x_1 and x_2) was partitioned into three membership functions, thus there were 12 parameters. Assume that the mean and standard deviation of the membership function were denoted by $m^j_{x_i}$ and $\sigma^j_{x_i}$, where $i = 1, 2$ and $j = 1, 2, 3$. In addition, there were 9 rules (3×3) in the rule-base, which resulting extra 9 parameters, r_n where $n = 1, 2, \ldots, 9$. Hence, a total of 21 parameters ($3_{\ membership\ functions} \times 2_{\ parameters} \times 2_{\ input\ variables} + 9_{\ rules}$) were needed to be tuned by GA. Each of the membership function parameters was encoded (genotype representation) into a 8-bit binary string whereas each of the rule parameters was encoded into a 2-bit binary string. As a result, the total length of the binary string was 114 bits. The corresponding illustration of the chromosome structure is shown in Fig. 2.6. Subsequently, all parameters must be decoded (phenotype representation) during fitness evaluation. The rule parameters were decoded into integers range from 0 to 4. Likewise, the membership function parameters were decoded into real numbers using linear mapping equation as shown below:

$$g_p = G_q^{min} + (G_q^{max} - G_q^{min}) \times \frac{A_q}{2^N - 1} \tag{2.5}$$

where g_p denotes the actual value of the q^{th} parameter, A_q denotes the integer

Table 2.2: Upper and lower limits of the parameters

Parameters	Gmin	Gmax
Width mean	0	250
Width standard deviation	10	30
Period mean	100	1000
Period standard deviation	30	100
Amplitude mean	0	1.8
Amplitude standard deviation	0.01	1

represented by a N-bit string gene, G_q^{max} and G_q^{min} denote the user defined upper and lower limits of the gene respectively (see Table 2.2).

$m_{x_1}^1$	$\sigma_{x_1}^1$	$\ldots$	$m_{x_2}^3$	$\sigma_{x_2}^3$	r_1	$\ldots$	r_9

Figure 2.6: Chromosome structure.

The next important consideration following the representation is the choice of fitness function. A good fitness function can reflect the objective of the system. Unlike traditional gradient-based methods, GA's can be used to evolve systems with any kind of fitness measurement functions including those that are nondifferentiable, discontinuous, etc. How to define the fitness evaluation function is totally problem dependent. Unlike for prediction and estimation problems which normally use mean-square error or absolute difference error related function, the number of correctly classified classes or misclassified classes was used for classification problems. The fitness function is as follows:

$$fit = \frac{AC_{NSR} + AC_{VF} + AC_{VT}}{3} \tag{2.6}$$

where AC denotes the percentage of correctly classified class.

After each of the chromosomes was evaluated and associated with a fitness, the current population undergoes the reproduction process to create the next generation of population. The "tournament with replacement" selection scheme was used to determine the members of the new generation population. The purpose of the selection mechanism was to focus the search process on the most promising regions of the search space. Variation operators (crossover and mutation) play important roles in GA. They facilitate an efficient search and guide the search into new regions. Crossover facilitates exploration, while mutation facilitates exploitation of the search space. The improvement in performance takes place over iterations in accord with some prescribed stopping criteria.

2.3.5 Results and Discussion

The performance of the genetically evolved NSFLC was tested on VFDB database. The database consists of 22 half-hour surface ECG recordings of subjects who experienced episodes of sustained ventricular tachycardia, ventricular flutter, and ventricular fibrillation. The algorithm applied a signal window of 4 seconds. 60 records of NSR, 60 records of VF and 60 records of VT were extracted as train data. Similarly, the same number of ECG data was extracted as test data. All data was extracted randomly according to the annotation of the database and they covered across multiple individuals.

In the experiment, single point crossover and bit-wise flipping mutation were adopted. For simplicity, the probabilities of crossover and mutation were held constant for the entire run of the GA (200 iterations), which were 0.8 and 0.03 respectively. Since GA is a stochastic search algorithm, the results reported is based on the average of 10 runs. An example of GA convergence trace is shown in Fig. 2.7 when optimising NSFRBC using pulse width and amplitude as inputs.

Three performance measurements (*accuracy*, *sensitivity*, and *specificity*) were

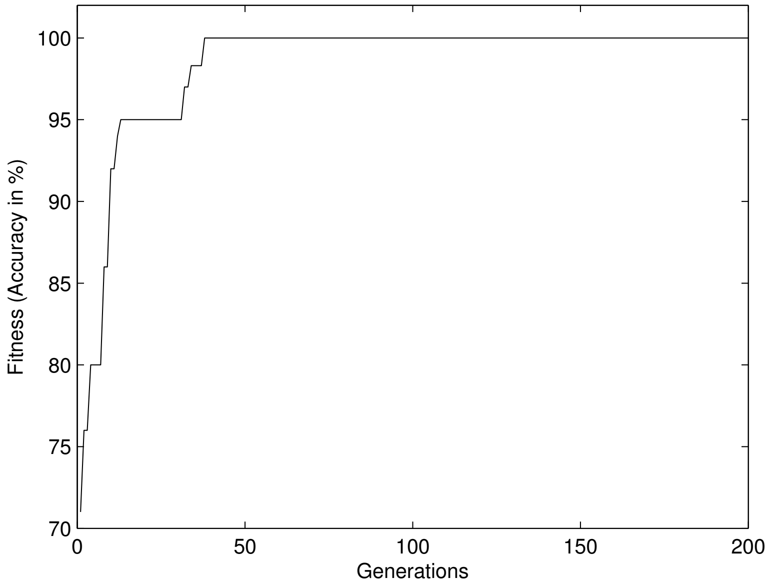


Figure 2.7: GA convergence trace.

used to benchmark the classification systems. Sensitivity is the probability that a test is positive, given that the person has the disease, whereas specificity is the probability that a test is negative, given that the person does not have the disease. The parameters are defined as follows:

$$\text{Sensitivity, } SE = \frac{TP}{TP + FN} \tag{2.7}$$

$$\text{Specificity, } SP = \frac{TN}{TN + FP} \tag{2.8}$$

where TP, TN, FP and FN are defined in Table 2.3.

The results of the test are shown in Table 2.4. Comparing the results for Configurations 1 and 2, it may be concluded that using the amplitude as an input feature produced better accuracy than using the pulse period as a feature. By changing the input signal from period to amplitude, the singleton fuzzy classifier boosts the overall classification results (accuracy +6.66%, sensitivity +6.57%, and specificity +3.31%). Thus, the first hypothesis is proven to be true, that is,

Table 2.3: Notation Used In Sensitivity And Specificity Equations

Predicted/Real	Rhythm A	Rhythm B
Rhythm A	True Positive (TP)	False Negative (FN)
Rhythm B	False Positive (FP)	True Negative (TN)

configuration 2 will outperform configuration 1. Clearly, amplitude is a better alternative to pulse period as the classifier input. In fact, medical practitioners always observe the amplitude information rather than period information when they want to determine the types of ventricular arrhythmias. Period information could be unreliable because shorter pulse period could be resulted from exercise activity or patient's emotion. The structure of Configuration 2 would be easier for medical practitioners to understand as it is more similar to the method they employ. Next, the results from configuration 3 show the superiority of using NSFRBC. Both configurations 1 and 3 adopted pulse period as one of the inputs. Although the results for singleton input case shows that pulse period is inferior to amplitude as the input, NSFRBC manages to handle most of the uncertainties and thus providing improvement over its singleton counterpart. This may suggest that the advantage of using NSFRBC is even more apparent than using a more suitable sets of features. On top of that, NSFRBC still manages to enhance the classification capability when a better input is used (see config. 4 vs. config. 2). Hence, the second hypothesis was also proven to be true. The results imply that singleton classifier has less degrees of freedom or flexibility in classifying a more non-linearly separable set of data. On the other hand, the findings show that a NSFRBC provides more advantages when features with more uncertainties are used as the input signal. Small improvement may still achievable when the input features have less uncertainties. In short, NSFRBC is proven to be a very useful tool in handling the uncertainties exist in pattern classification. The boxplot (Fig. 6.6) shows the classification accuracies of using four different configurations for 10 runs. The boxplot reveals that Configuration 1 yields the most inconsistent results throughout the iterations while others have comparable ranges of variances. In addition, the non-singleton structure also helps GA to converge to global maxima more frequent with the harder-to-classify feature in mind (Configuration 1 versus Configuration 3).

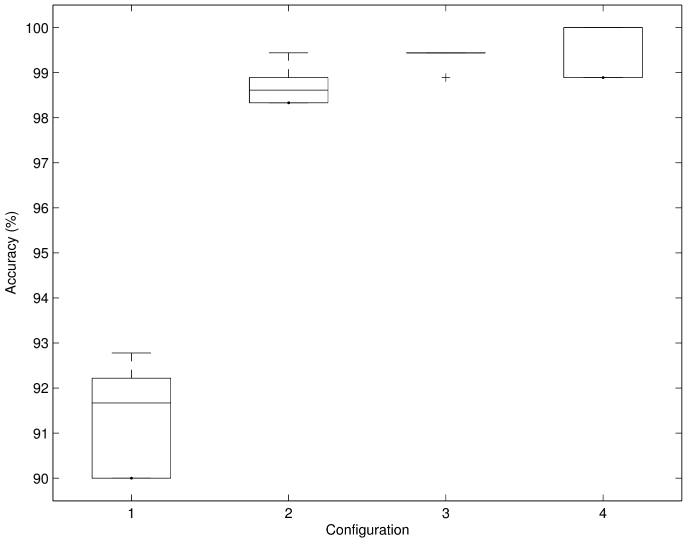


Figure 2.8: The boxplot for each configuration over 10 runs.

Since other algorithms have been proposed for ventricular arrhythmia classification, it would be insightful to compare the obtained results to some existing algorithms. Table 2.5 shows that the proposed classification system is on a par with the existing algorithms. It should be noted that some of the algorithms used

Table 2.4: Classification Results With Different Configurations

Config.	Rhythm	AC (%)	SE(%)	SP (%)
1 [a]	NSR	100.00	100.00	100.00
	VF	83.33	90.91	92.00
	VT	91.67	84.62	95.65
	Average	91.67	91.84	95.88
2 [b]	NSR	100.00	100.00	100.00
	VF	95.00	100.00	97.56
	VT	100.00	95.24	100.00
	Average	98.33	98.41	99.19
3 [c]	NSR	100.00	98.36	100.00
	VF	98.33	100.00	99.17
	VT	100.00	100.00	100.00
	Average	99.44	99.45	99.72
4 [d]	NSR	100.00	100.00	100.00
	VF	100.00	100.00	100.00
	VT	100.00	100.00	100.00
	Average	100.00	100.00	100.00

[a] SFRBC with pulse width and period as input features.

[b] SFRBC with pulse width and amplitude as input features.

[c] NSFRBC with pulse width and period as input features.

[d] NSFRBC with pulse width and amplitude as input features.

own recorded ECG database instead of public accessible ECG database. Even though Zhang [44] and Thakor [49] claimed that they achieved very promising results with Complexity Measure and Sequential Hypothesis algorithms respectively, some researchers like Ayesta [50] and Chen [51] obtained much poorer results when they applied the aforementioned algorithms on the general ECG database. Chen claims that a database derived from defibrillator implantation studies is usually much more stable and rhythm specific in comparison to a general ECG database. Furthermore, MIT-BIH database includes broader range of VT which includes both monomorphic and polymorphic types. Therefore, the classification result is very dependent on database selection. Among all tests using the MIT-BIH (except for Hurst Index algorithm with the same results), the proposed method achieved highest classification rates. Next, the proposed algorithm uses the shortest window length (4.0s) for feature extraction as compared to other algorithms as far as the performance is concerned. Algorithms in [44, 49, 50, 51, 52, 53, 54] have window lengths of 7.0s, 8.0s, 8.0s, 20.0s, 4.0s, 4.8s, and 5.5s respectively to achieve their best results as shown in Table 2.5. Although the fuzzy rule-based classifier introduced by Chowdhury and Ludeman [52] has the same window length with the proposed method, the proposed method's results are far superior. A shorter window length will allow for a shorter detection time, given that the computation time of the classifiers are equivalent. The proposed NSFRBC only spends less than 0.05ms to classify an input, hence the computation time taken by the classifier is negligible compared to the time required for feature extraction.

Table 2.5: Comparative Results of Different Arrhythmia Classification Methods

	NSR			VF			VT			
Algorithm	AC %	SE %	SP %	AC %	SE %	SP %	AC %	SE %	SP %	**Data**
Complexity Measure [44],[50]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	ORD [e]
	NA	23.80	NA	NA	94.60	NA	NA	81.90	NA	VFDB [c], NSRDB [d]
Sequential Hypothesis Testing [49],[44],[51]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	ORD
	NA	NA	NA	90.00	NA	NA	81.00	NA	NA	VFDB
Rate and Irregularity Analysis [44]	100.00	100.00	100.00	87.75	85.88	89.08	87.75	84.71	89.92	VFDB
VF-Filter Leakage [44]	97.55	94.12	98.24	89.22	89.41	89.08	89.71	84.71	93.28	VFDB
Hurst Index [54]	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	MITDB [a]
Cross Correlation [53]	NA	NA	NA	NA	90.20	NA	NA	96.75	NA	MITDB, AHADB [b], ORD
Fuzzy Rule-Based [52]	94.30	NA	NA	78.00	NA	NA	82.00	NA	NA	VFDB
Sample Percentage in the Dynamic Range [50]	NA	100.00	NA	NA	94.60	NA	NA	81.90	NA	VFDB, NSRDB
Sequential Probability Ratio Test [51]	NA	NA	NA	93.00	NA	NA	96.00	NA	NA	VFDB
Proposed Method	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	VFDB

[a] MITDB- MIT-BIH Arrhythmia Database
[b] AHADB- American Heart Association Database
[c] VFDB- MIT-BIH Malignant Ventricular Arrhythmia Database
[d] NSRDB- MIT-BIH Normal Sinus Rhythm Database
[e] ORD- Own Recorded Database

2.4 Conclusion

This chapter presents non-singleton fuzzy rule-based classifier (NSFRBC) for handling the uncertainties in pattern classification problem. The analysis demonstrate that the NSFRBC has fuzzy classification boundary and noise suppression capability. These characteristics means that the NSFRBC is particulary suitable for problems where the boundaries between classes is non-distinct. To further demonstrate the benefits offered by the non-singleton framework, a NSFRBC evolved using genetic algorithm (GA) is assessed using a benchmark cardiac arrhythmias classification problem. Results indicate that a NSFRBC achieved good classification accuracy using features that are easier to extract, but contain more uncertainties.

Chapter 3

Type-2 Fuzzy Rule-Based Classifiers

It has been demonstrated in Chapter 2 that non-singleton type-1 fuzzy rule-based classifier is capable of handling input uncertainties more efficiently. However, non-singleton type-1 fuzzy classifier does not adequately account for the uncertainties associated with antecedent and consequent fuzzy sets as illustrated in Table 3.1. Specifically, the uncertainty about the ambiguities in the meaning of the fuzzy set labels are not considered.

As membership functions (MFs) of type-2 fuzzy sets are fuzzy and contain a footprint of uncertainty (FOU), they can model and handle both linguistic and numerical uncertainties associated with the inputs and outputs of the fuzzy logic system. It has been shown in [55] that the extra degrees of freedom provided by the footprint of uncertainty enables type-2 fuzzy logic system to produce richer outputs that cannot be achieved by type-1 fuzzy logic system with the same number of membership functions. Besides, type-2 fuzzy logic systems have more design parameters, which equips them with the potential to outperform their type-1 counterparts. In reality, problems that contain crisp and precise data do not exist. In Mendel's opinion, any problem that has previously lent itself to type-1 fuzzy sets, in which membership functions are uncertain, is an excellent candidate for re-examination using type-2 fuzzy sets [56]. Therefore, it would be interesting to find out how the FOUs can improve the fuzzy classifier performance.

It should be noted that type-2 FLSs presented throughout this thesis are restricted to interval type-2 FLSs rather than general type-2 FLSs. This is because it is computationally prohibitive to calculate the meet operations for each fired rules and to perform type reduction when general type-2 fuzzy sets are used [37]. The following section introduces interval type-2 fuzzy rule-based classifier. This is followed by the description of the procedures to design the classifier. The proposed Type-2 fuzzy classifiers have been applied to ECG arrhythmic classification problem which has been described in Section 2.3 of Chapter 2. The average period and pulse width of ECG data are extracted as the inputs to the classifier. Different sources of noises have been included to model the uncertainties associated with the vagueness in MFs and the unpredictability of the data. The results show that the proposed strategies to design the FOU are essential to achieve a high performance fuzzy rule-based classifier in face of the uncertainties.

3.1 Interval Type-2 Fuzzy Rule-Based Classifier

This section introduces the interval type-2 FRBC. Fig. 3.1 shows the general structure of the proposed type-2 fuzzy rule-based classifier. There are six com-

Table 3.1: Comparisons of Type-1 and Type-2 Singleton and Non-Singleton FLSs.

	Type-1 FLS	Type-2 FLS
Singleton fuzzification	- No uncertainties about antecedents or consequents	- Uncertainties about antecedents or consequents are accounted for (the FOU)
	- No uncertainties on measurements that activate the FLS	-No uncertainties on measurements that activate the FLS
	- Only a point output is obtained	- Both a type-reduced set and a point output are obtained
Non-singleton fuzzification	- No uncertainties about antecedents or consequents	- Uncertainties about antecedents or consequents are accounted for (the FOU)
	- There are uncertainties on measurements that activate the FLS; they are modeled as type-1 fuzzy numbers	- There are uncertainties on measurements that activate the FLS; they are modeled by treating the measurements as type-1 or type-2 fuzzy numbers
	- Only a point output is obtained	- Both a type-reduced set and a point output are obtained

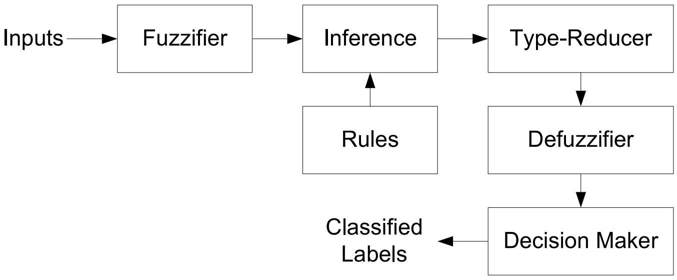


Figure 3.1: Structure of type-2 fuzzy rule-based classifier.

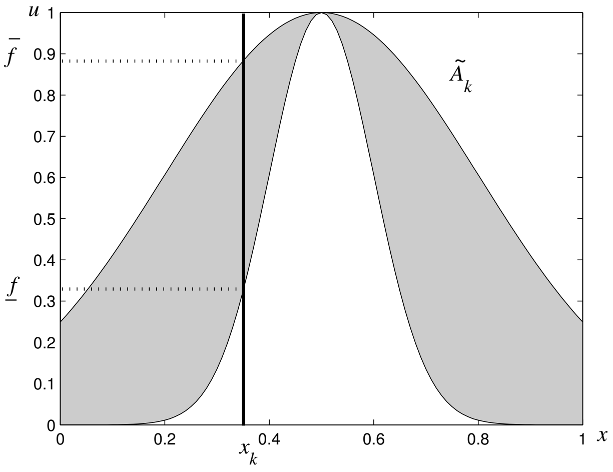


Figure 3.2: Supremum operation between type-2 antecedent fuzzy set, $\widetilde{A}_k$ and singleton input x_k produces firing strengths $[\underline{f},\bar{f}]$.

ponents in the architecture: rule base, fuzzifier, infererence engine, type-reducer, defuzzifier and decision maker. Once the rules have been established, a fuzzy rule-based classifier can be viewed as a mapping from inputs to outputs. Rules are the heart of a fuzzy logic system. They may be provided by experts or extracted from numerical data. The rule-base consists of M rules where each rule relates the domain $X_1 \times \cdots \times X_p \subseteq R^p$ to the range $Y \in R$ and can be expressed as the following intuitive IF-THEN statement:

$$R^j\text{: IF } x_1 \text{ is } \widetilde{A}_1^j \text{ and } \cdots \; x_p \text{ is } \widetilde{A}_p^j, \text{ THEN } y \text{ is } C^j$$

where R^j denotes the jth rule, $\widetilde{A}_k^j$ is an interval type-2 antecedent set associated with the kth input variable x_k $(k = 1, \ldots, p)$, and C^j represents the consequent set associated with the output variable y. The role of the fuzzifier in a fuzzy system is to map each of the element, x'_k, in the input vector $\mathbf{x}' = (x'_1, \ldots, x'_p)^T$ into the fuzzy set $\widetilde{X}'_k$. This process provides a natural framework for handling uncertain input information. There is a variety of methods for performing fuzzification. The most common approach is singleton fuzzification, which maps a crisp input into the following MF:

$$\mu_{\widetilde{X}'}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} = \mathbf{x}' \\ 0 & \mathbf{x} \neq \mathbf{x}' \end{cases}$$

for $\forall \mathbf{x} \in \mathbf{X}$. Next, the inference engine component computes the firing strengths for each rule which expresses how well the fuzzified input $\widetilde{X}'$ match the antecedents $\widetilde{A}'$. For type-2 FRBC, the inference engine produces two firing strengths for each rule (refer to Fig. 3.2), the lower and upper firing strengths of the jth rule, $\underline{f}^j(\mathbf{x}')$

and $\bar{f}^j(\mathbf{x}')$, are computed as:

$$\underline{f}^j(\mathbf{x}') = \prod\nolimits_{k=1}^{p} \sup_{x_k}[\mu_{\widetilde{X}'_k}(x_k), \underline{\mu}_{\widetilde{A}^j_k}(x_k)] \tag{3.1}$$

$$\bar{f}^j(\mathbf{x}') = \prod\nolimits_{k=1}^{p} \sup_{x_k}[\mu_{\widetilde{X}'_k}(x_k), \bar{\mu}_{\widetilde{A}^j_k}(x_k)] \tag{3.2}$$

where sup[.] denotes supremum operation [37]. Assume that the input is an interval type-2 Gaussian primary membership function with uncertain standard deviations:

$$\mu_{\tilde{X}_k}(x_k) = \exp\left[-\frac{1}{2}\left(\frac{x_k - m_{\tilde{X}_k}}{\sigma_{\tilde{X}_\mathrm{k}}}\right)^2\right] \text{ with } \sigma_{\tilde{X}_\mathrm{k}} \in [\sigma_{\tilde{X}_{\mathrm{k}1}}, \sigma_{\tilde{X}_{\mathrm{k}2}}] \tag{3.3}$$

and the antecedents are interval type-2 Gaussian primary membership functions with uncertain means:

$$\mu_{\tilde{F}_k}(x_k) = \exp\left[-\frac{1}{2}\left(\frac{x_k - m_{\tilde{F}_k}}{\sigma_{\tilde{F}_k}}\right)^2\right] \text{ with } m_{\tilde{F}_k} \in [m_{\tilde{F}_{k1}}, m_{\tilde{F}_{k2}}] \tag{3.4}$$

To obtain the value of x_k at which the supremum in (3.2) occurs under minimum t-norm:

$$\bar{x}_{k,\max} = \begin{cases} \frac{\sigma_{\tilde{X}_{k2}} m_{\tilde{F}_{k1}} + \sigma_{\tilde{F}_k} m_{\tilde{X}_k}}{\sigma_{\tilde{X}_{k2}} + \sigma_{\tilde{F}_k}} & , \text{ for } m_{\tilde{X}_k} \leq m_{\tilde{F}_{k1}} \\ m_{\tilde{X}_k} & , \text{ for } m_{\tilde{X}_k} \in [m_{\tilde{F}_{k1}}, m_{\tilde{F}_{k2}}] \\ \frac{\sigma_{\tilde{X}_{k2}} m_{\tilde{F}_{k2}} + \sigma_{\tilde{F}_k} m_{\tilde{X}_k}}{\sigma_{\tilde{X}_{k2}} + \sigma_{\tilde{F}_k}} & , \text{ for } m_{\tilde{X}_k} \geq m_{\tilde{F}_{k2}} \end{cases} \tag{3.5}$$

Likewise, to obtain the value of x_k at which the supremum in (3.2) occurs under minimum t-norm:

$$\underline{x}_{k,\max} = \begin{cases} \frac{\sigma_{\tilde{X}_{k1}} m_{\tilde{F}_{k2}} + \sigma_{\tilde{F}_k} m_{\tilde{X}_k}}{\sigma_{\tilde{X}_{k1}} + \sigma_{\tilde{F}_k}} & , \text{ for } m_{\tilde{X}_k} < \left[\frac{m_{\tilde{F}_{k1}} + m_{\tilde{F}_{k2}}}{2} - \frac{\sigma_{\tilde{X}_{k1}}\left(m_{\tilde{F}_{k2}} - m_{\tilde{F}_{k1}}\right)}{2\sigma_{\tilde{F}_k}}\right] \\ \frac{m_{\tilde{F}_{k1}} + m_{\tilde{F}_{k2}}}{2} & , \text{ for } m_{\tilde{X}_k} \in \left[\frac{m_{\tilde{F}_{k1}} + m_{\tilde{F}_{k2}}}{2} - \frac{\sigma_{\tilde{X}_{k1}}\left(m_{\tilde{F}_{k2}} - m_{\tilde{F}_{k1}}\right)}{2\sigma_{\tilde{F}_k}}\right. \\ & \left. , \frac{m_{\tilde{F}_{k1}} + m_{\tilde{F}_{k2}}}{2} + \frac{\sigma_{\tilde{X}_{k1}}\left(m_{\tilde{F}_{k2}} - m_{\tilde{F}_{k1}}\right)}{2\sigma_{\tilde{F}_k}}\right] \\ \frac{\sigma_{\tilde{X}_{k1}} m_{\tilde{F}_{k1}} + \sigma_{\tilde{F}_k} m_{\tilde{X}_k}}{\sigma_{\tilde{X}_{k1}} + \sigma_{\tilde{F}_k}} & , \text{ for } m_{\tilde{X}_k} > \frac{m_{\tilde{F}_{k1}} + m_{\tilde{F}_{k2}}}{2} - \frac{\sigma_{\tilde{X}_{k1}}\left(m_{\tilde{F}_{k2}} - m_{\tilde{F}_{k1}}\right)}{2\sigma_{\tilde{F}_k}} \end{cases} \tag{3.6}$$

If the input is a type-1 Gaussian membership function, then the equations (3.5) and (3.6) still apply. Next, if the input is singleton, then $\bar{x}_{k,\max} = \underline{x}_{k,\max} = x_k$. Fig. 3.3 gives the pictorial description of the input and antecedent operations (3.2) and (3.2) when minimum t-norm is used.

Before the final crisp output can be obtained, the output of the inference engine and the consequent must be processed. In a more general case where the consequent fuzzy sets $\widetilde{C}^j$ are interval type-2 sets, the center-of-set (COS) type-reduced set Y_{cos} can be computed with center-of-sets type reduction:

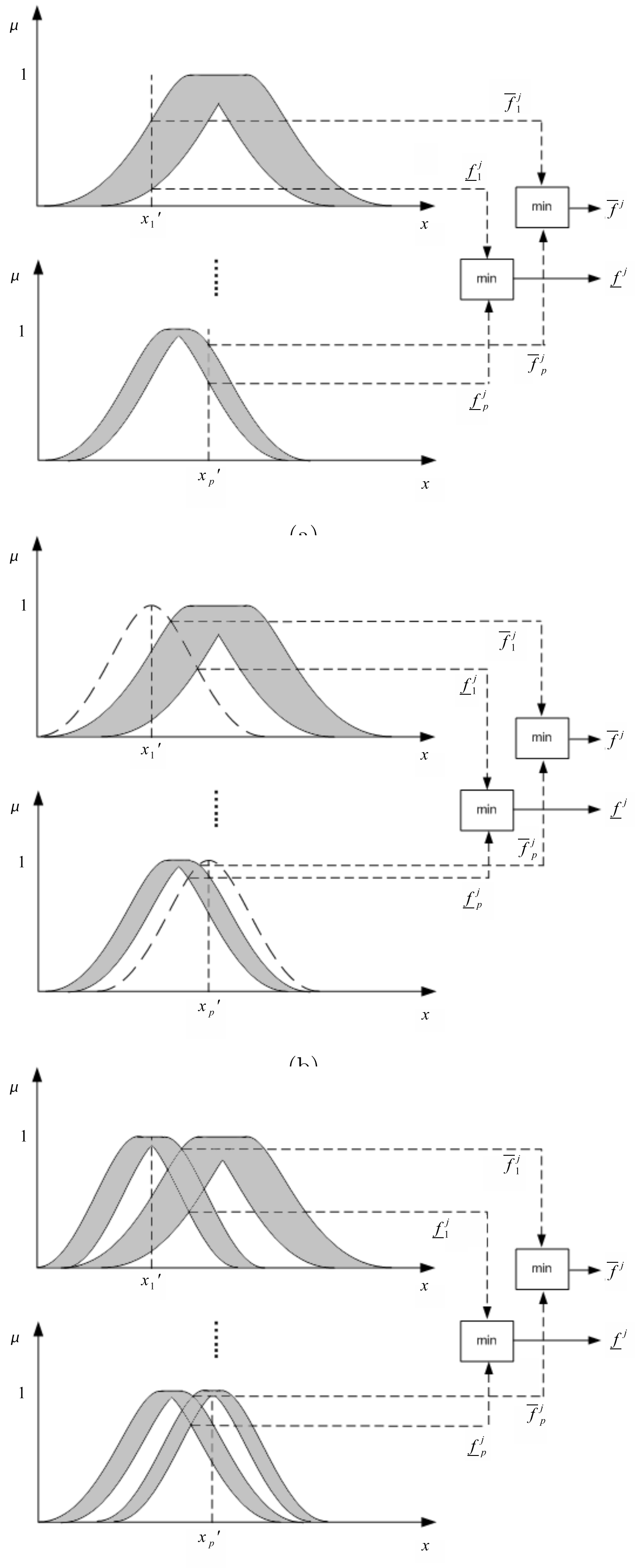
$$\begin{aligned} Y_{cos} = [y_l, y_r] &= \int_{y^1 \in [y_l^1, y_r^1]} \cdots \int_{y^M \in [y_l^M, y_r^M]} \int_{f^1 \in [\underline{f}^1, \bar{f}^1]} \\ &\quad \cdots \int_{f^M \in [\underline{f}^M, \bar{f}^M]} 1 \Bigg/ \frac{\sum_{j=1}^M f^j y^j}{\sum_{j=1}^M f^j} \end{aligned} \tag{3.7}$$

where $[y_l^j, y_r^j]$ denotes to the centroid of the set $\widetilde{C}^j$, which can be obtained from various methods defined in [37]. However, the consequent fuzzy sets in our classification problem correspond to the class labels and are represented by crisp number (singleton), the center-of-sets type-reduction above is simplified to height type-reduction by simply setting $y_l^j = y_r^j$. The type-reduced set which is an interval output, $[y_l(\mathbf{x}'), y_r(\mathbf{x}')]$ can be obtained via Karnik-Mendel iterative algorithm [57]. To compute y_l, the steps are:

1. Without loss of generality, assume that pre-computed y_r^j are arranged in ascending order; i.e., $y_r^1 \le y_r^2 \le \cdots \le y_r^T$;

2. Compute y_r as $y_r = \sum_{j=1}^T f_r^j y_r^j / \sum_{j=1}^T f_r^j$ by initially setting $f_r^j = (\underline{f}^j + \bar{f}^j)/2$ for $j = 1, \ldots, T$ and let $y_r' \equiv y_r$;

3. Find R $(1 \le R \le T-1)$ such that $y_r^R \le y_r' \le y_r^{R+1}$;



(a)

(b)

(c)

Figure 3.3: The operations between interval type-2 antecedent with different types of inputs using minimum t-norm. (a) Singleton input; (b) Non-singleton type-1 input; and (c) Non-singleton interval type-2 input.

4. Compute y_r as $y_r = \sum_{j=1}^{T} f_r^j y_r^j / \sum_{j=1}^{T} f_r^j$ with $f_r^j = \underline{f}^j$ for $i \leq R$ and $f_r^j = \bar{f}^j$ for $i > R$ and let $y_r'' \equiv y_r$;

5. If $y_r'' \neq y_r'$, then go to Step 6. If $y_r'' = y_r'$, then stop and set $y_r'' \equiv y_r$;

6. Set y_r' equal to y_r'', and return to Step 3.

The procedure for computing y_l is very similar to the one for y_r. Just replace y_r^j by y_l^j, and, in Step 3 find $L(1 \leq L \leq T-1)$ such that $y_l^L \leq y_l' \leq y_l^{L+1}$. Additionally, in Step 2 compute y_l as $y_l = \sum_{j=1}^{T} f_l^j y_l^j / \sum_{j=1}^{T} f_l^j$ by initially setting $f_l^j = (\underline{f}^j + \bar{f}^j)/2$ for $j = 1, \ldots, T$ and, in Step 4 compute y_l as $y_l = \sum_{j=1}^{T} f_l^j y_l^j / \sum_{j=1}^{T} f_l^j$ with $f_l^j = \bar{f}^j$ for $i \leq L$ and $f_l^j = \underline{f}^j$ for $i > L$.

The type-reduced set is then defuzzified to the crisp output, y by simply taking the average of y_l and y_r, i.e.:

$$y(\mathbf{x}') = \frac{y_l(\mathbf{x}') + y_r(\mathbf{x}')}{2} \tag{3.8}$$

Finally, the decision maker will determine the class label:

$$Class(\mathbf{x}') = \arg \min_j (y(\mathbf{x}') - \bar{C}^j) \tag{3.9}$$

where $\bar{C}^j$ denotes the singleton at the point having maximum membership in the jth consequent set. For Gaussian MF, this point is equal to the mean of the function.

Since in this chapter type-2 FRBC will be compared against type-1 FRBC, it is appropriate to briefly highlight the differences between both classifiers. The structure of a type-1 FRBC is similar to a type-2 FRBC except for a few aspects. Firstly, the inference engine will produce a firing strength, f^j for jth rule rather than an interval value. Secondly, the type-reducer does not exist since no type-2

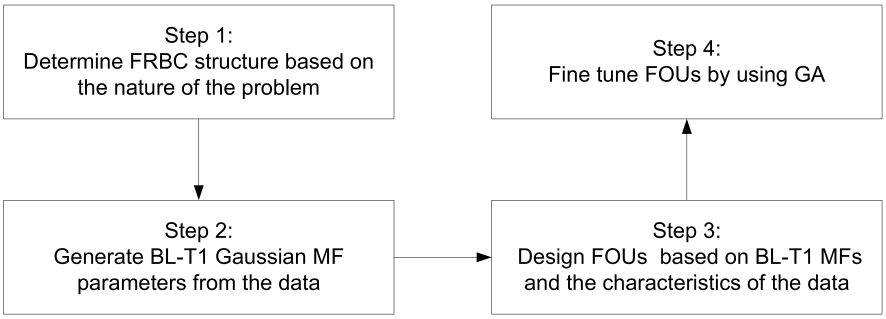

Step 1:
Determine FRBC structure based on the nature of the problem
Step 2:
Generate BL-T1 Gaussian MF parameters from the data
Step 3:
Design FOUs based on BL-T1 MFs and the characteristics of the data
Step 4:
Fine tune FOUs by using GA

Figure 3.4: The design strategy of Type-2 FRBCs.

number is involved. In other words, the output processing of a type-1 FRBC only consists of defuzzification. For height defuzzification, the crisp output, y can be computed as:

$$y(\mathbf{x}') = \frac{\sum_{j=1}^{T} y^j f^j}{\sum_{j=1}^{T} f^j}. \tag{3.10}$$

3.2 Type-2 Fuzzy Rule-Based Classifier Design Methods

In this section, the design strategy of the type-2 classifiers will be explained. A useful trait of the design methodology is most of the antecedent MF parameters can be conveniently derived from data itself. The design strategy, which comprises four steps, is summarised in Fig. 3.4. The first step is to determine the structure of the classifier. This can be achieved by establishing one fuzzy rule for each naturally distinguishable class. Since there are three classes, it would be intuitive to form only three rules (i.e., $T = 3$) in this problem. In addition, the number of antecedents for each rule is determined by the number of features which is two in our case. The next step is to determine the parameters of the MFs. To ensure the designed MF parameters are relevant to the data and to achieve good interpretability and transparency of the rule-base, our second strategy is to design the prototype type-1 MF where the mean M_k^j and standard deviation σ_k^j of the Gaussian MF parameter are computed according to the distribution of the data. For $\forall x \in Class j$:

$$\begin{aligned} M_k^j &= \frac{1}{N_j}\sum_{i=1}^{N_j} x_{i,k} && (3.11) \\ \sigma_k^j &= \sqrt{\sum_{i=1}^{N_j} \frac{1}{N_j}(x_{i,k} - M_k^j)^2} && (3.12) \end{aligned}$$

where N_j denote the total number of samples from class j. Alternatively, the mean of type-1 MF can be computed with other more advanced clustering algorithms like Fuzzy C-Means (FCM), Self-Organizing Map (SOM) etc. We define these prototype MFs as base-line type-1 (BS-T1) MFs.

Based on the BL-T1 MFs, we design three types of FOUs with the aim to account for different sources of uncertainties such as randomness of the data and the ambiguity in determining the exact membership functions. The first type-2 MF is shown in Fig. 3.5(a) where the upper membership function (UMF) is characterised by BL-T1 MF. The lower membership function (LMF) has the same mean as the UMF but with two different standard deviations, $(\underline{\sigma}_{L,k}^j, \underline{\sigma}_{R,k}^j)$. The idea to incorporate two different standard deviations is motivated by the fact that most classification problems have uneven data distribution with respect to the mean. For example, the ECG data distributions in Fig. 2.5 have different densities. The initial values of both parameters are then computed as:

$$\underline{\sigma}^j_{L,k} = \sqrt{\sum_{x \in Classj} \frac{1}{N'_j}(x_{i,k} - M^j_k)^2} \quad \text{for } \forall x_{i,k} \leq M^j_k \tag{3.13}$$

$$\underline{\sigma}^j_{R,k} = \sqrt{\sum_{x \in Classj} \frac{1}{N''_j}(x_{i,k} - M^j_k)^2} \quad \text{for } \forall x_{i,k} > M^j_k \tag{3.14}$$

where N'_j and N''_j represent the total number of samples from class j which satisfy the condition parts of (3.13) and (3.14) respectively. Since the asymmetrical FOUs are created by varying the standard deviations, this classifier is named as the type-2 uncertain standard deviations (T2-US) classifier.

The second one is known as type-2 uncertain means (T2-UM) classifier. As the name suggests, the UMF has two mean values, $[\bar{M}^j_{L,k}, \bar{M}^j_{R,k}]$. For $\forall x \in Classj$:

$$\bar{M}^j_{L,k} = \frac{1}{N'_j}\sum_{i=1}^{N'_j} x_{i,k} \quad \text{for } \forall x_{i,k} \leq M^j_k \tag{3.15}$$

$$\bar{M}^j_{R,k} = \frac{1}{N''_j}\sum_{i=1}^{N''_j} x_{i,k} \quad \text{for } \forall x_{i,k} > M^j_k. \tag{3.16}$$

This strategy is motivated by the limitations of clustering algorithms to locate the true mean of the MF. For example, the FCM algorithm is known to work well with evenly distributed data that are spherical in shape [58] but may not work well in the case of elliptical distribution such as the ECG classification problem in previous chapter where Fig. 2.5 clearly shows that NSR data comprise of two sub-clusters with high densities compared to VF and VT classes with sparse densities. Furthermore, all three classes have elliptical data distribution. The BL-T1 MF is used as the LMF for T2-UM classifier (see Fig. 3.5(b)).

Finally, the third classifier- type-2 uncertain standard deviations and means (T2-USUM), is the combination of T2-US and T2-UM classifiers. We see that BL-T1 MF automatically served as the principal MF, as shown in Fig. 3.5(c). For all three types of type-2 classifiers, the steps above intend to capture as much uncertainty as possible from the data through the FOUs. The initial parameters such as LMF of T2-US classifier, UMF of T2-UM classifier and both LMF and UMF of T2-USUM classifier served as the good initial search points in the later stage of optimisation by Genetic Algorithm (GA).

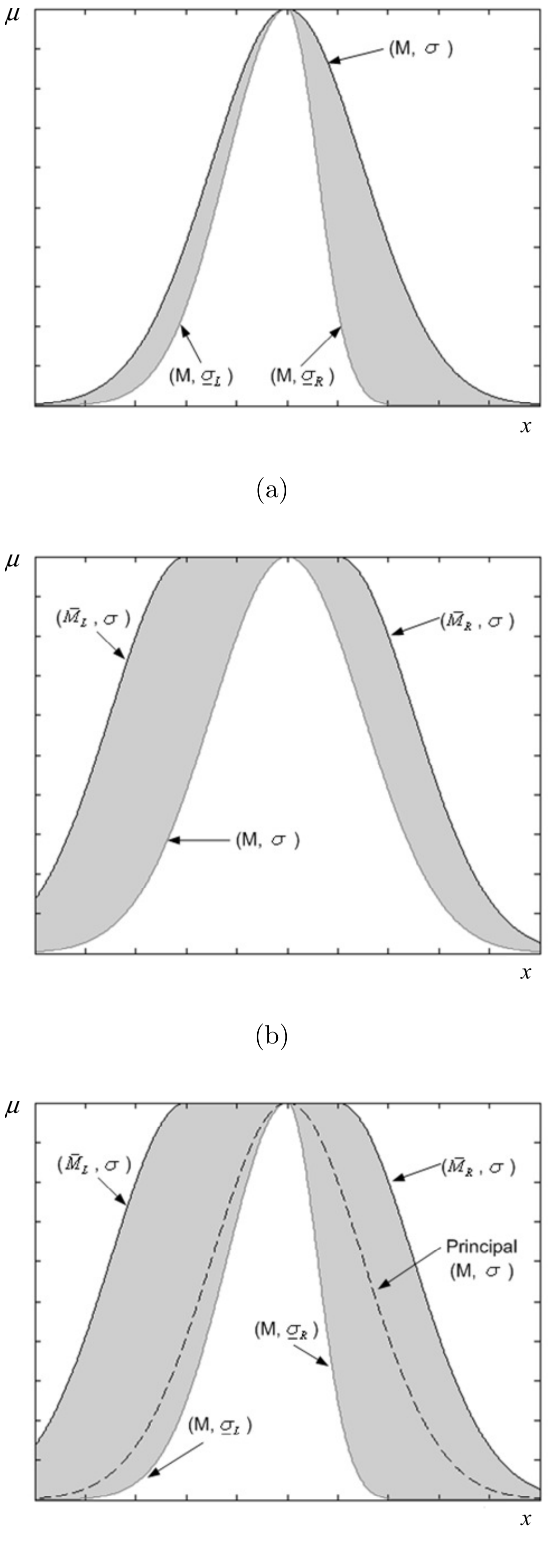
3.3 Experimental Results

In this section, we will carry out five case studies to examine the performances of different types of T2 FRBCs. They are benchmarked on the ECG arrhythmias classification problem. In addition, T2 FRBCs are compared against BL-T1 FRBC. The structure of a BL-T1 FRBC is similar to a T2 FRBC (refer to Section 3.1) except for a few aspects. Firstly, the inference engine will produce a firing strength, f^j for jth rule rather than an interval value. Secondly, the type-reducer does not exist since no type-2 number is involved. For height defuzzification, the crisp output, y can be computed as:

$$y(\mathbf{x}') = \frac{\sum_{j=1}^{T} y^j f^j}{\sum_{j=1}^{T} f^j} \tag{3.17}$$

If all FOUs of a type-2 FRBC disappear, then type-2 FRBC is immediately reduced to type-1 FRBC and there is no difference between the final outputs from both classifiers.

The first case study seeks to examine the importance of the evolved FOUs in T2 FRBCs compared to the BL-T1 FRBC when the means of the MFs are obtained via (3.11). The second case study focuses on whether the more advanced clustering algorithm can further improve the performance of T2 FRBCs. This is



(a)

(b)

(c)

Figure 3.5: Interval type-2 Gaussian membership functions with: (a) uncertain standard deviations, (b) uncertain means, (c) uncertain standard deviations and means.

achieved by computing the means through the FCM algorithm. Subsequently, the third and fourth case studies examine the performances of T2-FRBCs when the evolved antecedent membership functions in case study 1 are later perturbed with noises. In the third case study, only the means of the MFs (as computed in first case study) are corrupted with noises while the fourth case study is configured in such a way that only the standard deviations of the MFs are perturbed with noises. In practice, the uncertainties could be due to different experts' opinions which are used to construct the rule-base. As suggested by Mendel [37], words can mean different things to different people. Therefore, there exists vagueness in the linguistic labels. Moreover, the random disturbances could be due to the noisy training data. The dynamics of the ECG signal is inherently noisy because it is very sensitive to cable movement and muscle activity. In addition, the interference from electrical network can degrade the recording process especially for surface ECG recording. As the T2 FRBCs attempt to the encapsulate the aforementioned uncertainties in the FOUs of the antecedent sets, it may be able to outperform its T1 counterpart. The last case study aims to study if T2 FRBCs can handle the uncertainty associated with *unpredictability* [59, 60] better. Roughly speaking, this is one of the most important issues in ECG classification or any pattern classification problems. It reflects the situation where the applied testing samples deviate from the training samples to some extent. When referring to automated external defibrillator (AED), this occurs when the ECG signals produced by the patient deviate from the training samples during the algorithm development stage. Hopefully, T2 FRBCs can improve the tolerance towards the unpredictability. This case study is modeled by noise corrupted test data while the training data

UMF parameters (means and standard deviations)	LMF parameters (means and standard deviations)	Consequent values

Figure 3.6: Type-2 fuzzy rule-based classifier chromosome structure. UMF: upper membership function, LMF: lower membership function.

are unperturbed.

In case studies 3, 4 and 5, the noise source is modeled as a Gaussian function with zero mean and variance of 0.001:

$$P' = P + \sqrt{0.001} \times uniform() \tag{3.18}$$

where P represents either Gaussian MF mean or standard deviation parameter in case 3 and 4 or the input value in case 5 while P' represents the noise corrupted parameter. $uniform()$ denotes a function that generates a scalar value from a normal distribution with mean 0 and standard deviation 1.

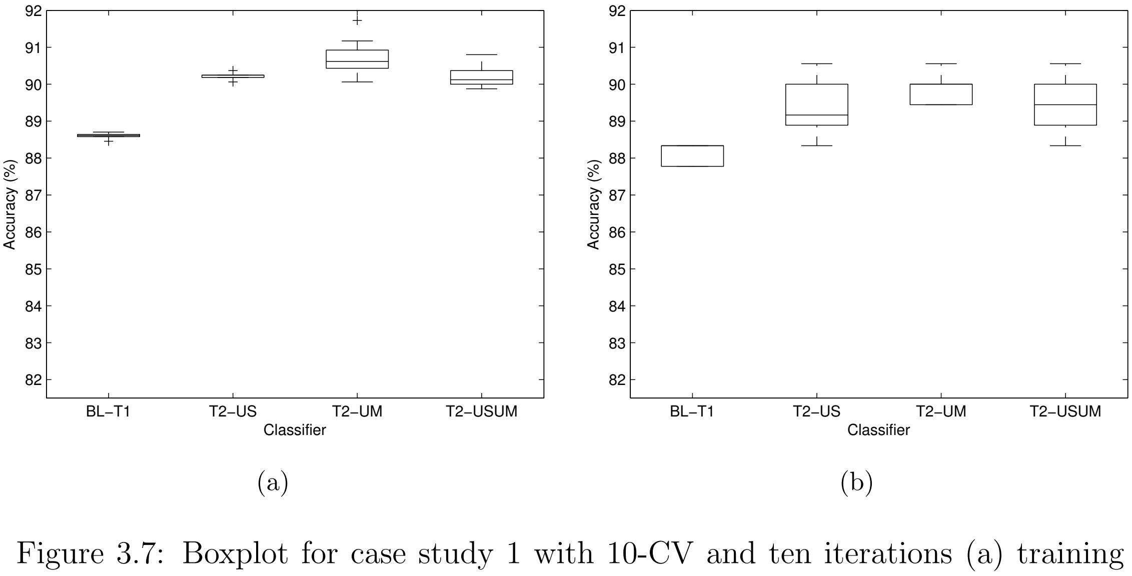
To test the performance of each classifier, we carried out a 10-fold cross-validation (10-CV) procedure. For GA optimisation, all membership function parameters and consequent values are encoded into a binary string. The chromosome structure is given in Fig. 3.6. Each value is represented as a 8-bit binary string. To decode the binary string, (2.5) can be used. The training accuracy is used as the fitness function. Adaptive parameters were not used in order to keep the algorithm as simple as possible. The mutation rate was initially set to 0.1, 0.05, 0.03 and 0.01 respectively. It was noticed that mutation rates of 0.01 can lead to premature convergence occasionally. On the other hand, the convergence speed of the solution can be very slow when the mutation rate was set to 0.1 or 0.05. It worked out that mutation rate of 0.03 gave the best compromise between the convergence speed and classifier's accuracy. Likewise, the crossover rates (0.5, 0.7, 0.8 and 0.9) and 0.8 were compared consistently gave the best GA performance. The population size was 30 and the maximum number of generations was fixed at 100. The optimisation process stops if there is no improvement in the fitness functions of the past 30 generations. In this particular application, the solutions usually converged between 70 and 90 generations. The results of the 10-CV experiments are summarised in Tables 3.2 and 3.3 where each result is obtained from the average of ten iterations for each classifier. Data for the first 4 case studies indicate that all FRBCs have good generalisation capabilities since the training and testing accuracies are very close. In the last case study, the perturbation of the test data set has inevitably decreased the testing accuracy. Comparing the first and the second case studies, the performance differences between the mean calculation method of simple averaging and FCM are minimal. This shows the FCM does not bring any improvement over the simple averaging method. In all cases, T2 FRBCs consistently outperform BL-T1 FRBC, this shows that the FOU is essential for a better FRBC. In particular, the third and the fourth case studies show that T2 classifiers are more robust against the perturbations in the MFs, hence less sensitive to design errors. BL-T1 classifier has significant testing performance drop ($\approx 5\%$) when the test data are corrupted by noises as shown in last case study while all T2 FRBCs remain relatively robust with only $1-2\%$ performance drop. This implies that the issue of unpredictability can be minimised through the proposed type-2 framework. The boxplots are shown in Fig. 3.7-3.11. It is clear that under any types of the perturbations mentioned above, T2 FRBCs remain robust and consistent compared to T1 FRBC.

Table 3.2: Average Training Accuracies of FRBCs (in %)

	BL-T1	T2-US	T2-UM	T2-USUM
Case 1	88.5926	90.2407	90.7407	90.2161
Case 2	88.4259	90.1914	89.9444	90.0432
Case 3	86.1729	88.4444	88.4259	88.3210
Case 4	87.0062	89.5309	89.4568	89.3148
Case 5	88.6914	90.1729	89.9876	89.8272

Table 3.3: Average Testing Accuracies of FRBCs (in %)

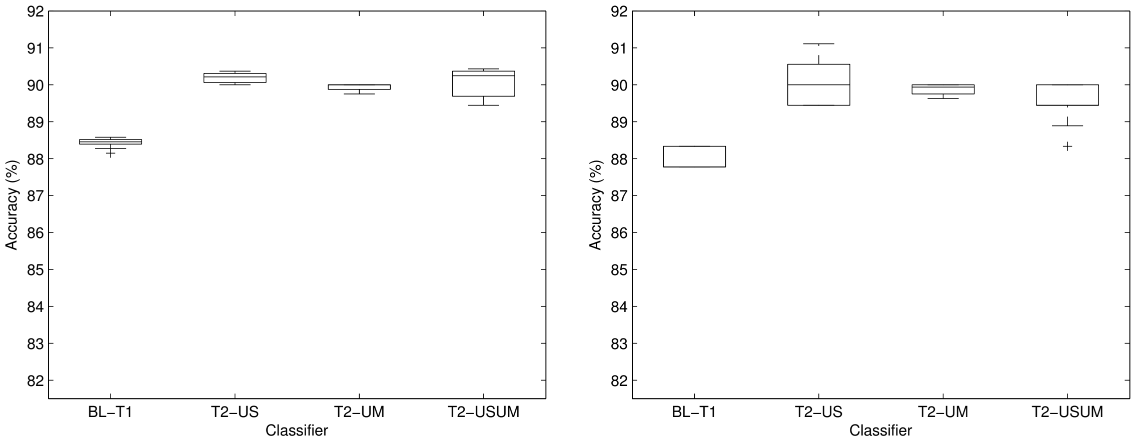
	BL-T1	T2-US	T2-UM	T2-USUM
Case 1	88.1111	89.3333	89.9444	89.3333
Case 2	88.0000	90.1111	89.8765	89.4444
Case 3	85.8334	88.1667	88.4444	88.0555
Case 4	87.1667	89.1667	89.0555	89.4444
Case 5	83.6667	88.6111	87.4444	87.5000



(a)

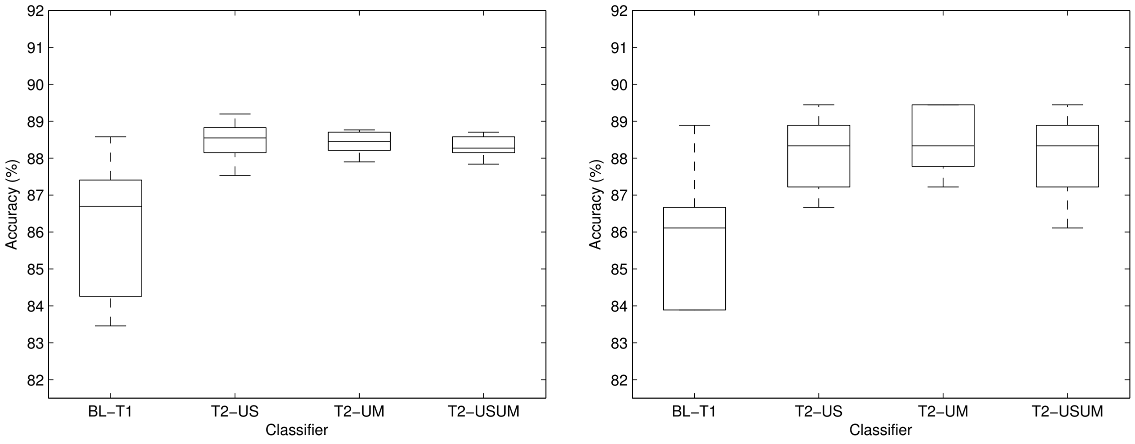
(b)

Figure 3.7: Boxplot for case study 1 with 10-CV and ten iterations (a) training accuracy, (b) testing accuracy.



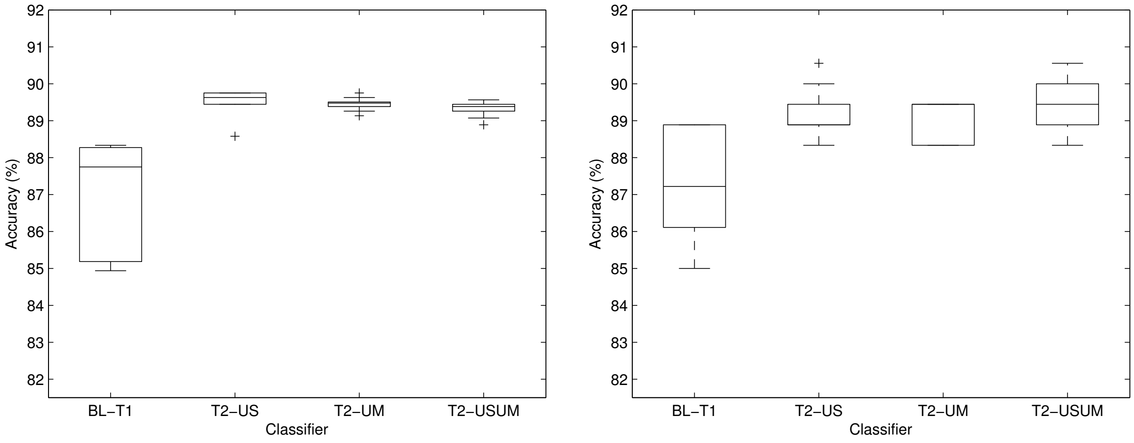
(a) (b)

Figure 3.8: Boxplot for case study 2 with 10-CV and ten iterations (a) training accuracy, (b) testing accuracy.



(a) (b)

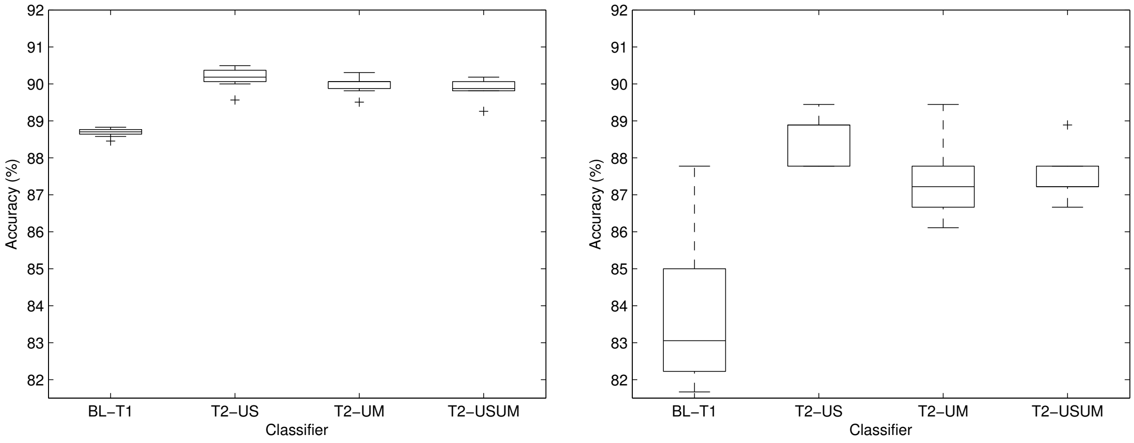
Figure 3.9: Boxplot for case study 3 with 10-CV and ten iterations (a) training accuracy, (b) testing accuracy.



(a)

(b)

Figure 3.10: Boxplot for case study 4 with 10-CV and ten iterations (a) training accuracy, (b) testing accuracy.



(a)

(b)

Figure 3.11: Boxplot for case study 5 with 10-CV and ten iterations (a) training accuracy, (b) testing accuracy.

3.4 Conclusion

This chapter presents some simple yet intuitive approaches to design a type-2 fuzzy rule-based classifier. It becomes clear that the uncertainties associated with the membership functions can be encapsulated by the footprint of uncertainty (FOU) and they are totally characterised by the upper membership functions (UMFs) and lower membership functions (LMFs). To enable the designed membership functions (MFs) to reflect the uncertainties in the data, the structure of the FRBC and the initial parameters are computed directly from the data set. When dealing with biological signals such as ECG, various uncertainties can arise and this will lead to the usability limitation of the algorithm in real-world applications. Through the extensive experimental results, it has been shown that T2 FRBCs has much better and stable performances in face of different sources of uncertainties. The optimisation of the FOUs by using GA is proven to be effective and model-free. While there is no preference for T2-US, T2-UM or T2-USUM FRBCs in this application, there is always a possibility that either one of them will outperform the rests in other applications. Finally, it is believed that the FRBC design strategy described in this chapter provide the general methodology that can be also be applied to other classification tasks.

Chapter 4

Robustness Analysis of Type-1 and Type-2 Fuzzy Rule-Based Classifiers

In recent years, there has been significant growth in interest in type-2 fuzzy logic system due to its ability to handle the uncertainties through the concept of footprint of uncertainty (FOU). While type-2 fuzzy logic has shown promising results in control applications, research showed that classification accuracy of type-2 fuzzy systems may not be better. This is due to the different nature of the control and classification problems. Type-2 fuzzy controller is capable of producing smoother continuous control surface but this advantage may not cogent in classification problem because the classifier output is a discrete class label. Another advantage of type-2 fuzzy controllers is robustness against uncertainties in the form of noise. This chapter seeks to study whether the robustness characteristics of type-2 fuzzy logic controller extends to classification problems.

4.1 Introduction

Uncertainties are unavoidable in real world applications. Hence, any methodology that successfully makes the transition from theory to practice should be equipped with a certain degree of robustness against imprecise data. In the field of classification, an outstanding algorithm needs to provide classification accuracy and be robust against uncertainties that exist during the design and implementation phases. A classifier may be deemed to be robust if it is less sensitive to data variations, and is able to handle insufficient training data scenario [61]. Likewise, in [62] the robustness of the classifier is associated with immunity against missing values in training data and test data. Nanopoulos et al. [63] considered a classifier to be robust if it is able to handle noise which can disrupt the learning process. In other words, a classifier is said to be robust if it is capable of coping well with uncertainties (arising from deficiencies in the available information caused by incomplete, imprecise, ill-defined, not fully reliable, vague, and contradictory information) in various stages of classifier design.

The popularity of fuzzy pattern classification stems from the fact that a FRBC provides a framework to incorporate both subjective (i.e., expert opinion) and objective (i.e., design samples where the knowledge can be extracted) information, hence it may be able to outperform other classifiers [17]. It is possible to integrate this valuable knowledge into the fuzzy logic system due to the system's similar reasoning style to the human being. However, an ordinary (type-1) fuzzy set does not capture uncertainty in all of its manifestations, particularly when it arises from vagueness in the shape of the membership function. Type-2 fuzzy sets have been introduced to capture the uncertainties. The uncertainties of the membership functions (MFs) could arise from differing expert opinions which are used to formulate the fuzzy rules or due to the noisy inputs when they are used to train the FRBC.

In recent years, there are two major categories of type-2 fuzzy application in pattern classification. The first category refers to the methods that incorporate type-2 fuzzy sets to the conventional classifiers. Most techniques in this category either fuzzify the conventional classifier's component or treat type-2 fuzzy logic system (FLS) as the input/output processor. Choi and Rhee [58] used IT2FLS as a pre-processor to generate fuzzy membership based on the distance between the pattern and the centroid of the class. The fuzzy membership values are then used as inputs to back-propagation neural networks. It has been shown that Type-2 fuzzy pre-processing resulted in about 0.92% to 2.16% improvement for Pima Indian Diabetes data set compared to type-1 counterpart. Chen et al. [64] used a type-2 fusion model to combine multiple support vector machines (SVMs) classifiers. By utilising the distance of the data to SVMs hyperplanes and the accuracy information of the individual SVMs, their experiment results show that the fusion model outperform individual SVMs in most cases. In [18, 65], John et al. pre-processed the expertise of clinicians using type-2 fuzzy sets for use with neuro-fuzzy clustering for classification of sports injuries in the lower leg. They concluded that type-2 representation can improve the classification results. Zeng et al. [23] hybridised type-2 fuzzy sets with Gaussian mixture models (GMMs), known as T2 FGMMs whereby the conventional multivariate Gaussian parameters are represented as type-2 membership functions with uncertain mean or uncertain variance. They observed that T2 FGMMs performs as good as or better than GMMs in classification rates under the condition of insufficient training data. In addition, they extended Hidden Markov Models (HMMs) to type-2 fuzzy HMMs (T2 FHMMs). Based on the results in phoneme classification, they concluded that T2 FHMMs outperform classical HMMs in terms of classification rate and robustness against babble noise. Since the works in this category are extensions of the conventional classifiers, the main idea is to improve the existing classifier performance by incorporating type-2 framework. Unfortunately, this is not always the case as there is no guarantee the FOU have good coverage on the system's uncertainty [66]. While not as well studied as in the previous category, the second category refers to the methods based on native type-2 fuzzy rule-based classifier which is still remain a niche field. In [17], type-2 FLCs with non-hierarchical and hierarchical architecture was applied to the classification of battlefield ground vehicles. The input to the system is a set of acoustic features. The input is inherently noisy due to the variation of the vehicle traveling speed, along with the environmental variations (e.g., wind and terrain). To further model the input uncertainties, the input is modeled as an interval type-2 fuzzy set whose membership function (MF) is a Gaussian function that is centered at the measured value but with an uncertain standard deviation. Given the noisy acoustic inputs, it was observed that interval type-2 fuzzy rule-based classifier (FRBC) only gives marginally improvements over type-1 FRBC. The authors have raised a few important questions with regards of fuzzy pattern classification. These include in what way are type-2 FRBCs considered outperform type-1 counterpart (e.g., in terms of the classification error rate, or generalisability or robustness), and how much uncertainty must be present in a problem so that it is worthy of trading the complexity of type-2 FRBC for better performance.

This chapter aims at advancing the understanding of IT2 fuzzy classifiers by addressing some of the above mentioned issues. As was pointed out [67], it is a challenging task to analyse behavior of type-2 FRBC mathematically because there is no closed-form mathematical equation for Karnik-Mendel type-reduction [37]. For this reason, experimental studies using synthetic data sets and a real-world classification problem were carried out. The main interest was to find out if the footprint of uncertainty (FOU) may provide extra degrees of freedoms to improve the robustness and performance of type-2 FRBC. In the following sections, the robustness of type-2 fuzzy rule-based classifier (FRBC) is studied in three aspects. In sub-section 4.2.1, the effectiveness of type-2 fuzzy classifier to handle different levels of noises in unseen data is investigated, given that the classifier itself is only trained with noiseless data. In sub-section 4.2.2, the advantage of type-2 classifier in handling the imprecise boundary associated with improper feature extraction is demonstrated via a real-world problem in the automotive industry. In sub-section 4.2.3, the robustness of type-2 framework against randomness in classifier designs is examined. The results and discussions are presented in each sub-section. Finally, conclusion is drawn in section 4.3.

4.2 Robustness of Type-2 Fuzzy Classifier

Rule base is a key part of a fuzzy classifier. The common method to determine the number of rules is to adopt full rule base where the rule base comprises all possible combinations of the input fuzzy sets. However full rule base may be unnecessarily large, especially if the number of inputs is big. Consequently, another design method is to prune the unnecessary rules using an optimisation algorithm. In this study, four fuzzy rule-based classifiers are proposed to examine if type-2 classifier can outperform type-1 counterpart. They are type-1 FRBCs with full and pruned rule base (T1-FRBC(F), T1-FRBC(P)) and type-2 FRBCs with full and pruned rule base (T2-FRBC(F), T2-FRBC(P)) also. All of them shares the same general IF-THEN rule structure as in Section 3.1 of Chapter 3. For example, a full rule base for a 2 × 2 fuzzy partition consists of four rules:

$$\begin{aligned}
R1 &: \text{ IF } x_1 \text{ is } \textit{small} \text{ AND } x_2 \text{ is } \textit{small}, \text{ THEN } y \text{ is } Class^{R1} \\
R2 &: \text{ IF } x_1 \text{ is } \textit{small} \text{ AND } x_2 \text{ is } \textit{large}, \text{ THEN } y \text{ is } Class^{R2} \\
R3 &: \text{ IF } x_1 \text{ is } \textit{large} \text{ AND } x_2 \text{ is } \textit{small}, \text{ THEN } y \text{ is } Class^{R3} \\
R4 &: \text{ IF } x_1 \text{ is } \textit{large} \text{ AND } x_2 \text{ is } \textit{large}, \text{ THEN } y \text{ is } Class^{R4}.
\end{aligned}$$

The classification decision is determined using winner-takes-all approach. The four fuzzy rule-based classifiers are designed using Genetic Algorithm (GA). For the full rule base, GA is used to optimise the membership functions. In the case of pruned rule base, additional bits are added to the chromosome. These bits serves as flags, where a value of '0' signify that the rule should be omitted and '1' means that the rule should be included. Type-1 Gaussian and Type-2 Gaussian (uncertain standard deviations and means) membership function parameters and rule selection flags are encoded into binary chromosome. The chromosome for type-1 fuzzy classifier is same as that given in Fig. 2.6 with additional rule selection bits. Likewise, the chromosome for type-2 fuzzy classifier is similar to that shown in Fig. 3.6 with additional rule selection bits. The classification accuracy of the classifier is used as the fitness function. The crossover rate and mutation rate are fixed at 0.8 and 0.03 respectively.

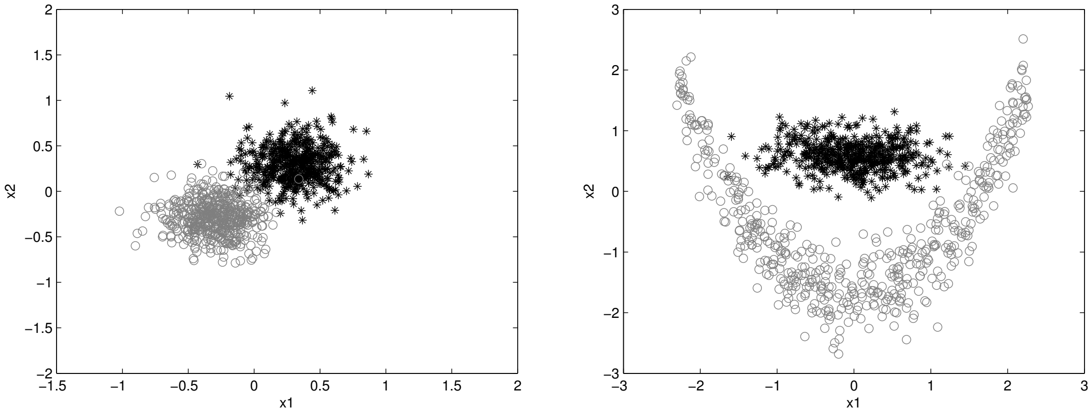
4.2.1 Robustness Towards Noisy Unseen Samples

Different types of fuzzy systems may cater for different sources of uncertainties. Mendel [37] suggests that a non-singleton type-1 fuzzy logic system can be used to handle uncertainties caused by noisy measurements. The inputs are modeled as type-1 fuzzy numbers. A mathematical analysis [68] shows that the non-singleton fuzzifier manages to minimise the effect of noise. Wu et al. [17] take further efforts to model the noisy input that contain simultaneous variations in mean and standard deviations as interval type-2 fuzzy set. While one can perform such input analysis when the noisy data is accessible, most real-world problems may not have access to noisy data. For instance, in fault diagnosis a malfunctioning machine is unable to produce signal measurement. If the symptom is known beforehand, then one can simulate the required measurement to train the classifier. However, it is very common that these simulated measurements often deviate from true data due to the discrepancy between the mathematical model and the system/plant. Likewise, the true data is most likely to be corrupted by environmental disturbances which again deviates from the optimistic simulated training data. In many cases, it may be almost impossible to obtain exact information from a given training data set. The imprecise data representation can leave an adverse effect for later classification. The deviation resembles a form of noise contamination. Thus, a robust classifier must be able to minimise the noise effect. This section aims to study the capability of type-2 FRBC to address this issue.

Two sets of two-dimensional synthetic data– Gaussian and Clown data sets (see Fig. 4.1) are used during the experiment. The Matlab codes that are used to generate these data are given in the Appendix. θ_G and $\theta_C \in [0, 1]$ are the parameters that control the amount of perturbation added to the data. The amount of perturbation is proportional to the value of θ_G or θ_C. For both data sets, the training data consist of 2 classes with 500 points in each class. As illustrated, the training data have minimal overlap thus the decision boundary is distinct. The testing sets for Gaussian and Clown data containing 500 points each are drawn from the same distributions. Noise level-0 testing data have minimal perturbation added (σ_G and σ_C are set to 0.25) as shown in Fig. 4.2(a) and 4.3(a). Thus the boundary between two classes is distinct. Perturbations are then added to the testing data gradually (by varying σ_G, σ_C to 0.5 (noise level-1), 0.7 (noise level-2), 0.9 (noise level-3)) to increase the deviations from the training data (see Fig. 4.2(b)–(d) and Fig. 4.3(b)–(d)). Genetic algorithm is used to optimise fuzzy rule-based classifiers. To obtain unbiased results, GA is run for 20 times with each run consists of 100 generations. Therefore, there are 20 different designs (due to different membership function parameters or number of rules) for each type of classifier. The classifiers are strictly trained with noiseless data only and then tested using both noiseless and noisy data. Unless otherwise stated, the experimental outcomes in this sub-section are based on the average of 20 classification accuracies for each type of classifier.

For Gaussian data set, two fuzzy partitions (i.e., ‘small’, and ‘large’) are created in each feature space so the full rule base has 4 rules as aforementioned. The classification results for Gaussian data set are shown in Table 4.1. The results show that there is no difference in training accuracies between type-1 and type-2 fuzzy classifiers. This observation also applies to testing accuracies (noise level-0). Hence, it is noticed that type-1 and type-2 FRBCs are equally capable of classifying the noiseless data very well. However, type-2 classifiers starts to outperform type-1 counterparts when dealing with noisy testing data (i.e., noise level-1 onwards). Fig. 4.4(a) summarises the testing accuracy improvement of type-2 FRBCs over type-1 FRBCs. From noise level-0 to level-2, the improvement is on the rise. At noise level-3, the improvement starts to saturate for full rule base system but not for pruned rule base system. The graph also shows that more improvement can be achieved when full rule base is used. For Gaussian data set, type-2 FRBCs with pruned rule base outperform the rests as far as noise handling is concerned.

For Clown data set, three fuzzy partitions (i.e., 'small', 'medium', and 'large') are used in each feature space. Thus a full rule base has 9 rules which encompass all possible combinations of the cartesian product. Table 4.2 presents the classification results for Clown data set. As in Gaussian data set, there is no distinguishable performance difference between type-2 and type-1 FRBCs when the data is not corrupted with noise. The advantages of type-2 FRBC become apparent when various levels of noises (level-1 to 3) are added in. Fig. 4.4(b) depicts the testing accuracy improvement that is obtained by using type-2 FRBCs instead of type-1 counterparts. While the improvement starts to decline at noise level-3, there is a constant boost in improvement when noise progresses from level-0 to level-2. As opposed to Gaussian data set, fuzzy classifiers with pruned rule bases resulted a more rewarding advancement performance compared to those with full



(a)

(b)

Figure 4.1: Synthetic train data set: (a) Gaussian, (b) Clown. Each set has 1000 samples.

Table 4.1: Classification Results for Gaussian Data. The Classifiers are Trained With Noiseless Data and Tested With Data Under Different Noise Levels.

Classifier	Training Accuracy (%)		Testing Accuracy (%)							
			Noise Lvl-0		Noise Lvl-1		Noise Lvl-2		Noise Lvl-3	
	Mean	Stdv	Mean	Stdv	Mean	Stdv	Mean	Stdv	Mean	Stdv
T1-FRBC(F)	98.74	0.29	98.00	0.34	83.15	1.51	76.23	1.37	70.46	1.65
T2-FRBC(F)	98.88	0.10	98.08	0.20	83.81	0.88	77.36	1.20	71.59	0.90
T1-FRBC(P)	98.99	0.14	98.07	0.21	84.15	0.84	78.16	1.37	71.94	0.98
T2-FRBC(P)	98.96	0.13	98.14	0.19	84.40	0.73	78.55	0.94	72.96	0.58

rule base. In Clown data set experiment, type-2 FRBCs with pruned rule base again tends to give best results with all noise levels in testing samples.

This section elucidates the importance of type-2 framework to handle uncertainty in the test data set. Although the improvement is on the small scale, type-2 FRBCs allow us to approach pattern classification problem more confidently.

4.2.2 Robustness Against Selected Features

The performance of a classifier is heavily dependent on choices that the designer of the classifier makes based on his/her insights into that problem [17]. One of them

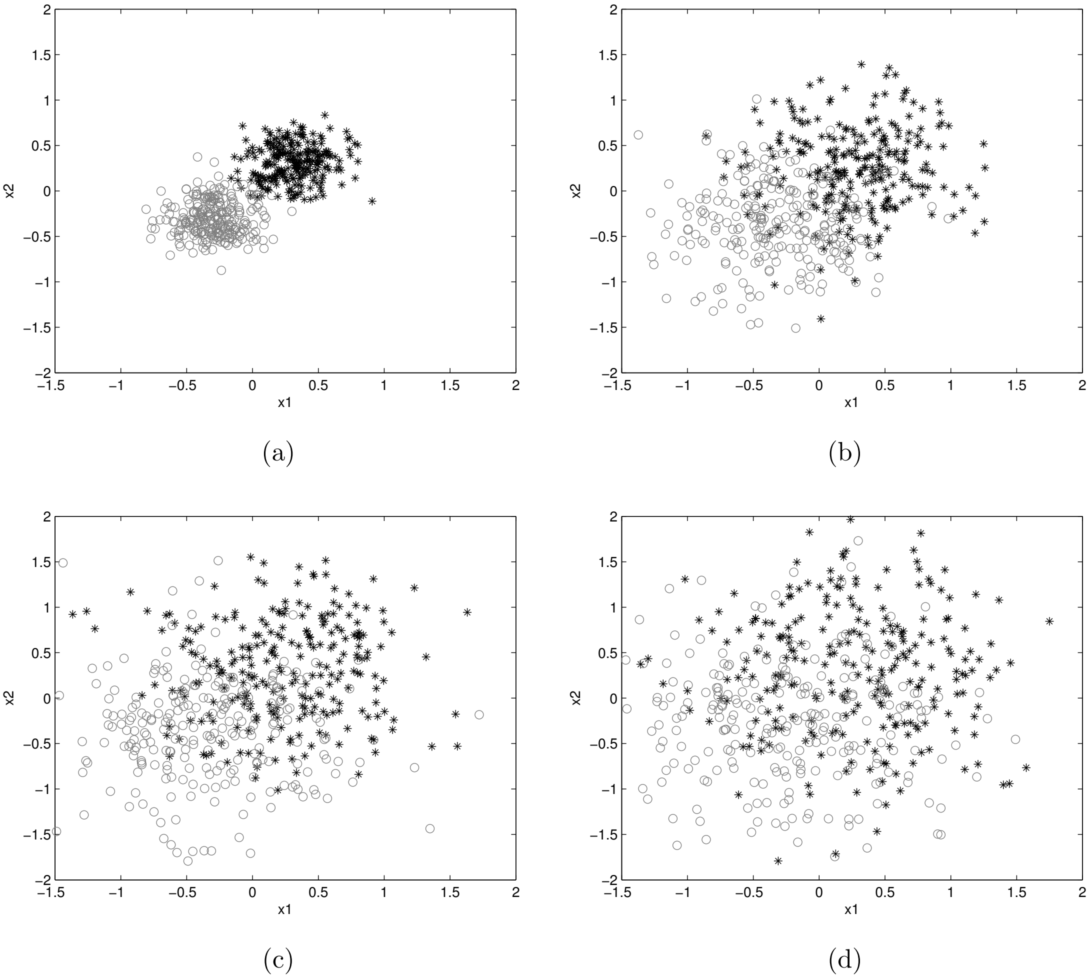
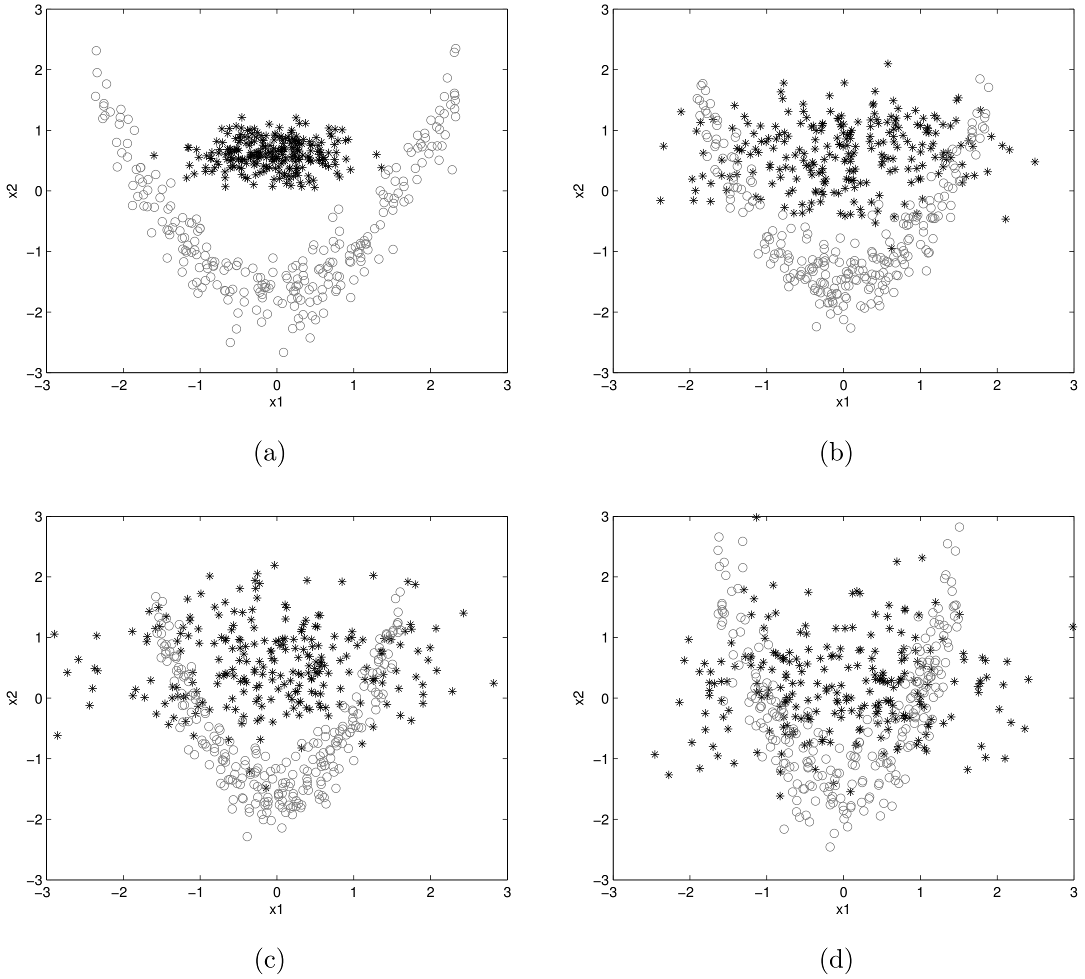


Figure 4.2: Synthetic Gaussian test data set with different noise levels: (a) Level-0 ($\sigma_G = 0.25$), (b) Level-1 ($\sigma_G = 0.5$), (c) Level-2 ($\sigma_G = 0.7$), (d) Level-3 ($\sigma_G = 0.9$). Each set has 500 samples.

Table 4.2: Classification Results for Clown Data. The Classifiers are Trained With Noiseless Data and Tested With Data Under Different Noise Levels.

Classifier	Training Accuracy (%)		Testing Accuracy (%)							
			Noise Lvl-0		Noise Lvl-1		Noise Lvl-2		Noise Lvl-3	
	Mean	Stdv	Mean	Stdv	Mean	Stdv	Mean	Stdv	Mean	Stdv
T1-FRBC(F)	99.53	0.12	99.48	0.15	84.92	0.82	75.74	1.60	62.42	1.94
T2-FRBC(F)	99.45	0.16	99.47	0.16	85.43	1.01	76.60	1.51	63.15	1.61
T1-FRBC(P)	99.42	0.20	99.26	0.34	85.34	1.61	76.32	1.60	62.95	2.08
T2-FRBC(P)	99.43	0.14	99.35	0.23	86.08	1.60	77.46	1.29	63.81	1.72



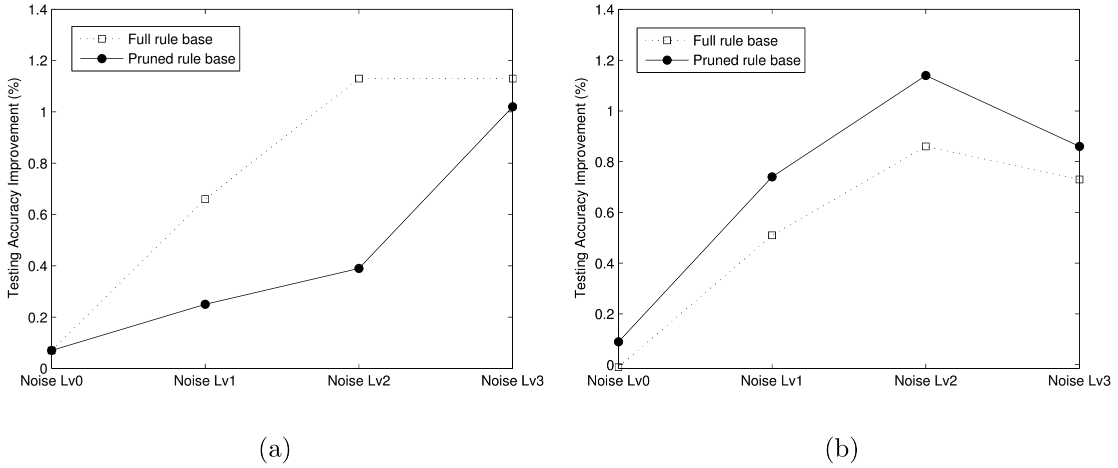
(a)

(b)

(c)

(d)

Figure 4.3: Synthetic Clown test data set with different noise levels: (a) Level-0 ($\sigma_C = 0.25$), (b) Level-1 ($\sigma_C = 0.5$), (c) Level-2 ($\sigma_C = 0.7$), (d) Level-3 ($\sigma_C = 0.9$). Each set has 500 samples.



(a)

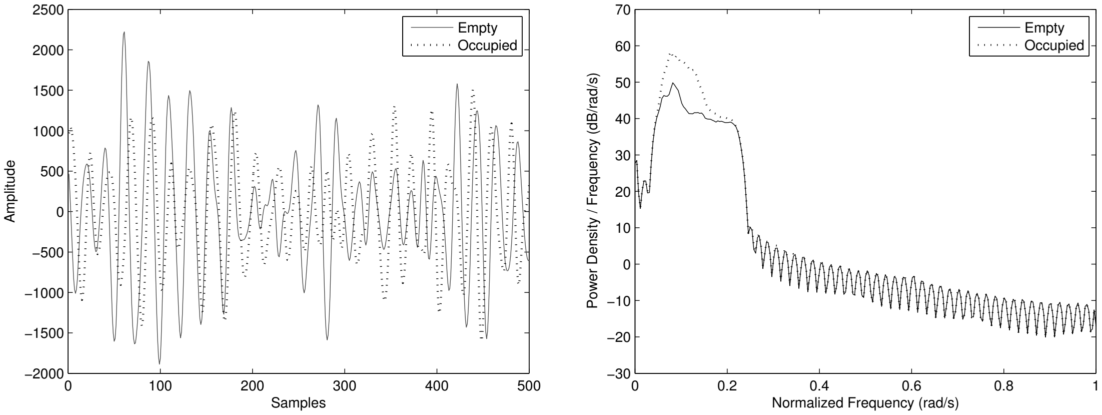
(b)

Figure 4.4: Improvement of testing accuracy of type-2 FRBCs over type-1 FRBCs for data set: (a) Gaussian, (b) Clown.

is the selection of the input feature set. An effective set of features can ease the design of classifier tremendously, especially for fuzzy classifier whereby the number of rules will increase exponentially with the increase of feature dimensions [69]. In contrast, if the feature selection is not optimum then the classification performance will be degraded. Unfortunately, it is difficult, and generally an open problem, to select an optimum set of features for different applications. Designer with experience may incorporate his/her knowledge about the classification problem. For example, by utilising expert knowledge that the ECG amplitude is a better feature to differentiate between ventricular tachycardia and ventricular fibrillation the designer can use this feature to improve the classifier performance [68]. On the other hand, designer with statistical background may try to use statistical tool such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) to select a compact set of projected features. Therefore, one source of uncertainty in pattern classification is the ambiguity in feature selection.

The fuzzy rule-based classifiers used in this study have been applied to Ford automotive data set [70]. In this real-world application, the classifier needs to detect the presence of a human in a vehicle. One possible scenario would be that when a driver returns to his or her vehicle at night, particularly in a deserted location, the knowledge that no one is hiding inside the vehicle can provide peace of mind.

Raw analog signals were collected from a vibration sensor which is located at the vehicle's suspension system. The signals are then filtered by a low pass filter (LPF) and converted into digital signals [71]. Each diagnostic session has 500 sample points. The length of the sequences reflects the time available for



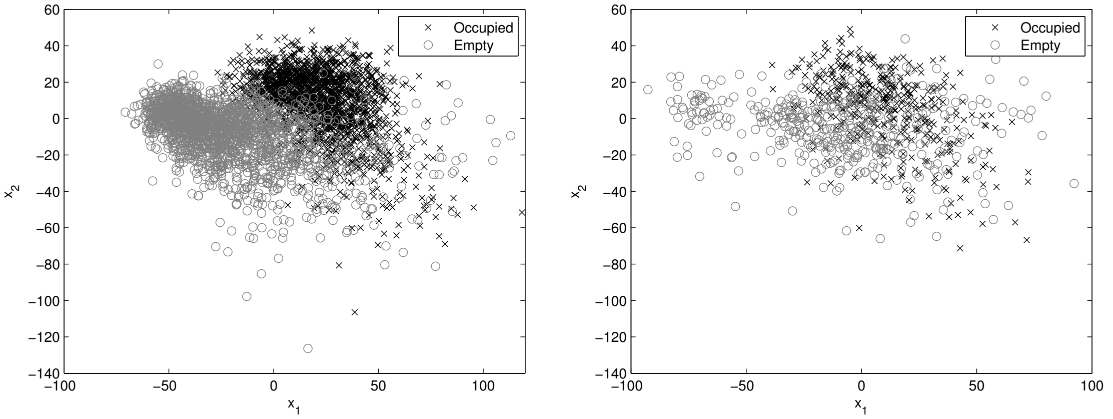
(a)

(b)

Figure 4.5: (a) The vibration signals from two cases, there is no clear feature that distinguishes between both signals by visual inspection. (b) The average periodogram of the training samples, the more discriminative features are concentrated at the lower frequencies.

making the classification decision. Presumably, the task would be easier if the sequence length were increased, but this would violate the requirements of the application. The beginning of the sampling process is not aligned with any external circumstance or any aspect of the observed pattern. The training data (3306 samples) were collected under typical operating condition with minimum noise but the testing data (810 samples) were collected under noisy conditions such as wind disturbances.

The problem does not appear to have a simple solution that emerges from visual inspection of these data sequences as shown in Fig. 4.5(a). Periodogram, a graphical data analysis technique for examining frequency-domain models of an equi-spaced time series, may be useful to reveal any interesting features. In this application, the periodogram are computed with 512-point FFT and triangle window. Thus, the periodogram is a coefficient vectors with length of 257 [72]. The average periodograms of the training samples are shown in Fig. 4.5(b). The figures show that the discriminative features are mostly located at the low frequency regions. If each of the periodogram coefficient is regarded as an input dimension, then the total number of feature dimensions is 257 which is impractical for most classifiers. Therefore, it is necessary to reduce the feature dimensions to lower dimensions. PCA is one of the most popular feature dimensionality reduction techniques. This technique searches for directions in the data that have largest variance and subsequently project the data onto it. However, it is completely unsupervised, knows only about variance, and nothing about different classes of data. In light of this, LDA may reveal class structure better. This technique maximises the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability. Fig. 4.6 and 4.7 show the two-dimensional scatter plots where the features are extracted with PCA and LDA respectively. It is clear that the variance within classes is smaller and the variance between classes is bigger in LDA projection. The feature space produced by LDA is more linearly separable while PCA gives less optimal separation between two classes especially on the noisy test data. As a result, the data produced by PCA require a more sophisticated classifier in order to handle the blurred decision boundary. In contrast, LDA projected data impose a less stringent requirement on the classifier. This analysis shows that feature extraction affects the performance of classifiers. In practice, it may not be feasible to test different input feature sets. Analysis with a large number of variables repeatedly generally requires a large amount of memory and computation power. Although best results can be achieved when an expert constructs a set of application-dependent features, in many applications there is a lack of expert knowledge. Thus, it would



(a)

(b)

Figure 4.6: 2-D scatter plots of PCA projected (a) train data, (b) test data. Test data has higher degree of overlapping between both classes due to the noises inherent in the raw data.

Table 4.3: Confusion Matrix for a Binary Classifier

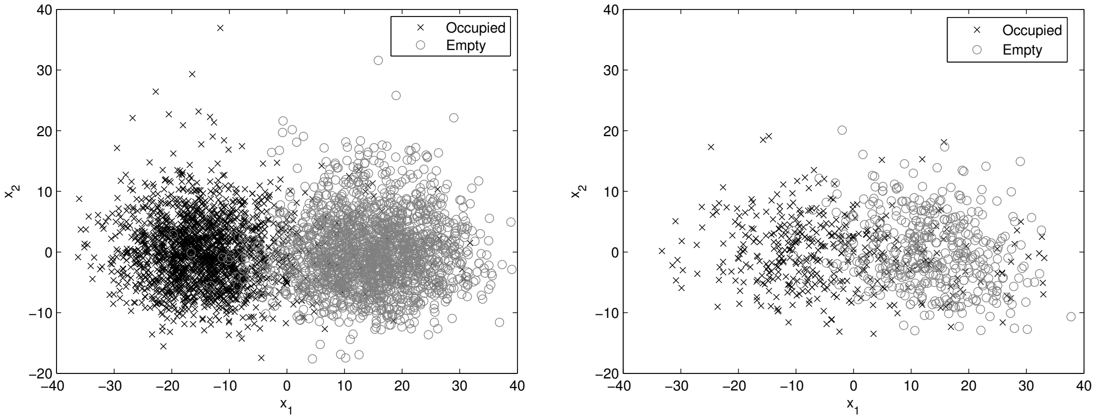
	Predicted Negative	Predicted Positive
Negative Examples	a	b
Positive Examples	c	d

be interesting to investigate if type-2 FRBC can reduce the reliance on good choice on input feature compared to type-1 FRBC.

Similar to sub-section 4.2.1, the robustness performances of four types of FRBCs are compared. Each kind of classifier consists of 10 different designs which are evolved with GAs separately. The performance metrics for this application are the accuracy of the classifier and the false positive rate (FPR) (4.1) respectively.

$$\text{False positive rate} = \frac{b}{(a+b)} \tag{4.1}$$

The confusion matrix is given in Table 4.3. False positive occurs when the classifier



(a)

(b)

Figure 4.7: 2-D scatter plots of LDA projected (a) train data, (b) test data.

reports that the vehicle is occupied when no one is actually in it. In particular, the systems can be sensitive to false positives in windy conditions.

As shown in Fig. 4.6 and Fig. 4.7, the features extracted with PCA has higher degree of overlapping compared to those extracted with LDA. Therefore, features extracted with PCA are harder to be classified. Table 4.4 shows the classification results on PCA extracted features. It is noticed that regardless of which types of design method (pruned or full rule base), type-2 FRBCs generally perform better than its type-1 counterparts. The results appear to confirm that type-2 FRBCs has the capability to resolve imprecise decision boundary better than type-1 FRBCs. On the other hand, type-1 and type-2 FRBCs have comparable performances for the set of features extracted using LDA which is shown in Table 4.5. This is due to the fact that LDA extracted features lead to more distinct decision boundary. Based on the experimental results, it may be argued that type-2 fuzzy classifier has not yet provided groundbreaking solution to feature extraction. Nevertheless, type-2 fuzzy classifier is superior as the dependency of classifier performance on the selection of the input feature can now be reduced to some extent.

Table 4.4: Average and Standard Deviation of Classification Accuracy and False Positive Rate Across 10 Iterations with PCA Based Feature Extraction.

Classifier	Data Set	Average ACC (%)		Average FPR (%)	
		Mean	Stdv	Mean	Stdv
T1-FRBC(F)	Train	90.90	1.83	6.87	2.94
	Test	72.27	3.06	25.02	5.10
T2-FRBC(F)	Train	92.08	0.64	5.44	1.09
	Test	74.19	1.16	22.75	1.69
T1-FRBC(P)	Train	92.63	0.70	4.85	1.23
	Test	73.99	1.43	23.66	2.47
T2-FRBC(P)	Train	92.96	0.47	4.97	1.22
	Test	74.14	1.28	23.98	2.68

Table 4.5: Average and Standard Deviation of Classification Accuracy and False Positive Rate Across 10 Iterations with LDA Based Feature Extraction.

Classifier	Data Set	Average ACC (%)		Average FPR (%)	
		Mean	Stdv	Mean	Stdv
T1-FRBC(F)	Train	95.49	0.00	3.10	0.00
	Test	81.04	0.06	7.14	0.17
T2-FRBC(F)	Train	95.50	0.02	3.09	0.02
	Test	81.01	0.05	7.21	0.13
T1-FRBC(P)	Train	95.47	0.05	3.27	0.41
	Test	81.05	0.47	7.56	0.88
T2-FRBC(P)	Train	95.48	0.04	3.35	0.57
	Test	81.25	0.49	7.50	0.85

Table 4.6: Testing Accuracies and False Positive Rates Comparisons Between The Proposed FRBC and Lv Jun's Classifier

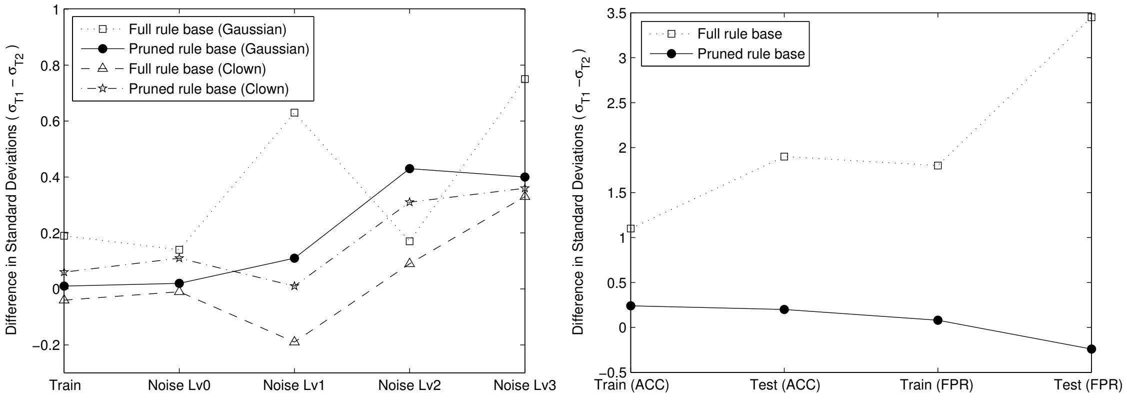
Method	# of Features	Testing Accuracy (%)	False Positive Rate (%)
Proposed FRBC	2	82.2	6.9
Lv Jun's Classifer	100	83.2	27.4

As a benchmark, the performance of the proposed FRBC is compared against the best overall statistical classifier proposed by Lv Jun as reported in Ford Classification Challenge [70]. The comparison is tabulated in Table 4.6. The results show that the merit of fuzzy rule-based classifier in this application is that it only requires very low feature dimensions to achieve reasonably good performance. For example, the proposed FRBC has very close testing accuracy (only 1% difference) and it achieved remarkable improvement on the false positive rate (20.5% lower) than the reported results.

4.2.3 Robustness To Randomness in Design Methods

A classifier reflects the functional relationship between the input patterns and output classes. Before the final decision function is determined, a set of candidate functions known as hypotheses should be evaluated. These hypotheses include different possible parameters. In the case of a fuzzy classifier, hypotheses are essentially the fuzzy rules. Genetic algorithms (GAs), a global stochastic search heuristic to find exact or approximate solutions to optimisation problems often give a pool of candidate solutions. The goal of the learning process is to determine the correct decision function in terms of suitable parameters. Due to the heuristic nature of GAs, each evolution run will produce its own set of parameters. Thus, various classifier designs are obtained after several GA optimisations. The ambiguity in classifier's parameters therefore becomes an entity of uncertainty. A consistent performance of the different designs in both training and testing stages is favorable. This is because it enables us to obtain more satisfactory results in fewer GA evolutions. In this sub-section, the performance consistency of GA evolved classifier designs in sub-sections 4.2.1 and 4.2.2 is investigated. Fig. 4.8(a) summarises the difference in classifier performance standard deviation between type-1 and type-2 FRBCs, that is $(\sigma_{T1} - \sigma_{T2})$. Positive value indicates that type-2 classifier is more consistent than type-1 while negative value indicates that type-2 classifier is less consistent. It becomes clear that for Gaussian data set, type-2 classifiers are always more consistent (indicated as all positive values). The consistency trend for Clown data set still shows that type-2 classifier has the edge although in a few cases type-1 classifier can be slightly more consistent. Likewise, the tendency for type-2 FRBCs with full rule base to deliver more robust performance is high for Ford data set when PCA method is used. Nevertheless, on Ford data (extracted with LDA method) the advantage of type-2 classifier is negligible. This is because the decision boundary is less ambiguous. For both synthetic and real problems, type-2 framework inarguably more consistent throughout the designs evolved by GAs. This is attributed to its FOU which decreases the incongruity between different classifier designs.

As far as the computational speed is concerned, type-1 and type-2 FRBCs with four rules each are tested on 1000 data. The average computation time measured



(a)

(b)

Figure 4.8: Difference in standard deviations ($\sigma_{T1} - \sigma_{T2}$) for (a) synthetic data sets and (b) Ford data set (with PCA method). Positive value denotes type-2 FRBC is more consistent than type-1 FRBC while negative value denotes type-2 FRBC is less consistent than type-1 FRBC.

for type-1 and type-2 FRBCs are 0.14s and 0.54s respectively (see Fig. 4.9). The results are based on the average on 20 independent runs of fuzzy classifiers due to the variable time required to compute Karnik-Mendel type reduction for type-2 FRBCs. Therefore, unless the application require very short computation time, type-2 classifier still offer a more robust performance.

4.3 Conclusion

Rhee [73] suggested that 'if the employment of type-2 fuzzy sets can provide significant improvement on performance, then the increase of computational complexity due to type-2 fuzzy sets may be a small price to pay.' Indeed, this has been validated in this work. Although the improvement is not up to "significant" levels but type-2 framework certainly shows its quality advantage over its type-1 counterpart. From the experiment that was carried out, type-2 fuzzy classifier shows its robustness in various aspects. This includes the ability to achieve better results on

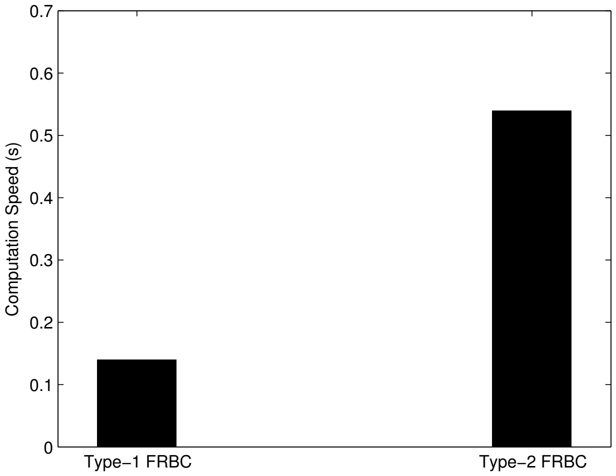


Figure 4.9: Total computation time required for 1000 training samples based on fuzzy system with four rules. Due to different computation time required for KM type-reduction in type-2 FRBC, the values are shown as the average of 20 generations.

unseen noise corrupted data. The significance of this finding is that it enables the use simulated data as an alternative to train the classifier when the real data is not readily available or the cost to collect it is expensive. Although the experiment is performed on synthetic data sets, the results can be further extended to real data. The other advantage offers by type-2 FRBC is that it can resolve the imprecise decision boundary better than type-1 FRBC as demonstrated in Ford automotive problem. This can reduce the impact caused by improper feature extraction. Finally, type-2 FRBC is more tolerate towards the randomness in different classifier designs evolved with GAs. A more consistent design not only reduce the classifier training time but also improve the reliability of the designed classifier. Hopefully, this can balance the tradeoff between the computation load and the reliability of the classifier well. In a nutshell, there is a strong reason to use type-2 fuzzy classifier as the performance of type-2 FRBC is at least comparable, if not better than type-1 FRBC. As a rule of thumb, for applications where the computation speed is not a major consideration, type-2 classifier should be adopted.

Chapter 5

Towards An Efficient Fuzzy Rule-Based Classifier Learning Algorithm with Support Vector Machine

5.1 Introduction

In recent years, fuzzy classifier design has been an active research topic. Many approaches have been adopted to design fuzzy classifiers which can be categorised into three general groups. The first group tunes only the consequent part. In this case, the rule base is fixed and the membership functions of the antecedent part are determined a prior. Two popular methods to tune the consequent part are least-squares and back-propagation algorithms. Next, rules pruning falls under the second group. The antecedent membership functions and consequents are fixed and optimisation algorithm such as GA is used to select the best subset of rules and discharges the rest [74]. The last group is a combination of the first and second methods. The system is built and tuned by simultaneous adaptation of rule-bases and antecedent membership functions. Again, GA has been successfully applied to learn both antecedent and consequent parts with fixed or varying number of rules [31], [38]. The popularity of GA is attributed to its two great advantages over back-propagation approach. Firstly, the functions that can be used in GA can be much more general in nature and knowledge of the gradient of the functions is not required. Secondly, GA is less likely to be trapped in local minima because they explore the solution spaces in multi directions at the same time. While GA is suitable for many optimisation problems, the algorithm is based on empirical risk minimisation, which is to minimise the training error. This could limit the generalisation capability of a classifier when over-fitting issue occurs. Therefore, it would be advantageous to combine GA with a learning method that minimises structural risk. In view of this, support vector machine (SVM) which is based on statistical learning theory is a very promising tool for structural risk minimisation. For classification problem, several studies based on Fuzzy Logic System with SVM can be found in the literature. In [75], fuzzy logic is used as a tool to transform grouped features so as to use the newly generated features to improve SVMs' performance with the aid of genetic optimisation. However, the grouping of features restricts the usability of this algorithm since the authors mentioned that there is no guide on the feature group selection. Other works in this direction include [76], [77], and [78] where fuzzy rule learning based on SVM are introduced. In [76] and [77], SVM is used to automatically generate fuzzy rules with fuzzy singletons in the consequent. The number of fuzzy rules is equal to the number of support vectors, which is usually very large. In [78], Takagi-Sugeno (TS) type kernel is adopted. Rule reduction is based on fuzzy clustering of the input data. However, their proposed method has several drawbacks. Firstly, prior investigation has shown that the classifier performance is sensitive to the initialisation of the first incoming data even though the authors claimed that the one-pass method is robust against the initialisation. According to [79], the one-pass clustering method requires further tuning of parameters. Next, there is a possibility that the firing strengths of some rules are zeros if the input space is not wholly covered by fuzzy rule "patches". This usually happens when the standard deviation of the Gaussian membership function is small and it can reduce the discriminant capability of the classifier. Thirdly, the algorithm has tendency to use overly large rule base. This is because each rule corresponds to a particular data, hence the performance is generally better if more training data is used to form the rule base. This is shown in the vehicle classification example in their paper where 76 percents of the training data were used to form the rule base.

The objective of this chapter is to design an efficient learning algorithm for TSK type fuzzy rule-based classifiers. The motivation of this work is based on the observations that a conventional fuzzy rule-based classifier is unable to handle high dimensional classification tasks and the existing learning techniques are either based on empirical risk minimisation or structural risk minimisation. Therefore, a new technique which makes use of the combination of GA, Fuzzy C-Means (FCM), FLS and SVM is proposed. Both empirical and structural risks can be minimised via GA and SVM simultaneously. Furthermore, while conventional SVM requires the kernel function to be symmetric and positive semidefinite (Mercer conditions), the new fuzzy framework can overcome this problem. In addition, FCM ensures the generated rules are compact enough while maintaining good generalisation. Hopefully, the hybrid techniques inherit all the advantages of each individual algorithm.

This chapter is organised as follows. The architecture of the proposed algorithm is explained in Section 5.2. Next, Section 5.3 delineates the training procedures which is the backbone of the proposed method. This part shows how various building blocks can be hybridised to form an efficient learning process. Simulation results are presented in Section 5.4. Finally, Section 5.5 gives the conclusion.

5.2 Architecture of EFSVM-FCM

There are many forms of TSK type fuzzy logic systems depending on the types of conjunction operators (t-norms) and consequents being used. In [2], five popular TSK type classifiers are explained. In this chapter, the r^{th} rule, R^r, in the rule base of the TSK system has the form of:

$$\text{IF } x_1 \text{ is } A_1^r \text{ and } \dots \text{ and } x_m \text{ is } A_m^r, \text{ THEN } y'\,{}^r = w^r \tag{5.1}$$

where $A_n^r(r = 1, \dots, K,\ n = 1, 2, \dots, m)$ are Gaussian antecedent fuzzy sets and w^r is a fuzzy singleton. The rule base can be illustrated as in Fig. 5.1. The proposed Evolutionary Fuzzy Rule-Based Support Vector Machines classifier with FCM clustering (EFSVM-FCM) architecture for a multi-input single-output (MISO) system is a four-layer fuzzy network. A detailed description of the functionality of each layer is given as:

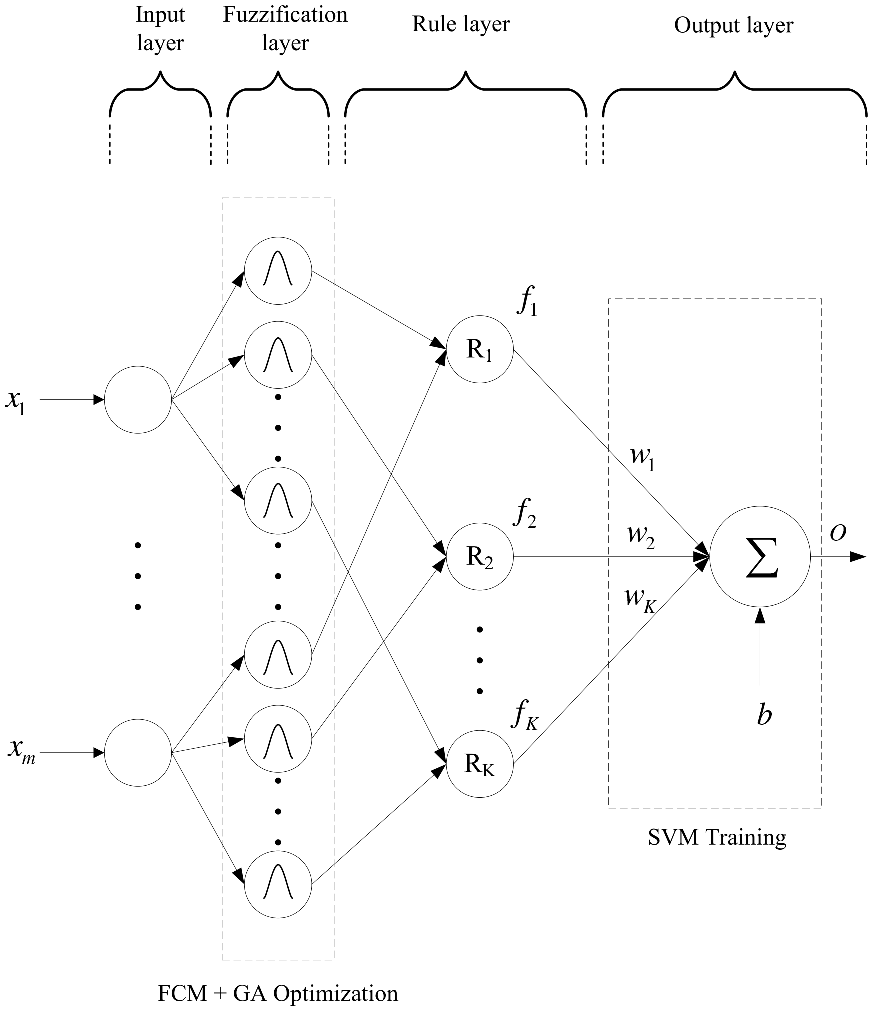


Figure 5.1: Architecture of EFSVM-FCM.

1) *Input Layer*: Neurons in the input layer transmit the input $\vec{x}$ directly to the antecedent layer. The number of nodes in this layer is equal to the number of input vector length, m.

2) *Fuzzification Layer*: Each neuron in this layer is a Gaussian membership function with mean, m_n^r and standard deviation, σ^r, with this the fuzzified input is given as:

$$\mu_n^r(x_n) = \exp\left[-\left(\frac{x_n - m_n^r}{\sigma^r}\right)^2\right] \tag{5.2}$$

Hereby, the fuzzifier is the singleton fuzzifier. The number of nodes in this layer is equal to $K \times m$. Please note that all Gaussian membership functions in a rule share the same standard deviation.

3) *Rule Layer*: Each neuron in this layer corresponds to a fuzzy rule which computes the firing strength or degree of match of the input to the antecedent sets defined for this rule; that is the layer only handles the antecedent part (if-part, or premise). To calculate the firing strength f^r of rule r, product t-norm operator is chosen:

$$\begin{aligned} f^r(\vec{x}) =& \prod_{n=1}^{m} \mu_n^r(x_n) \\ =& \exp\left[-\sum_{n=1}^{m}\left(\frac{x_n - m_n^r}{\sigma^r}\right)^2\right] \\ =& \exp\left[-\left(\frac{\parallel \vec{x} - \vec{m}^r \parallel}{\sigma^r}\right)^2\right] \end{aligned} \tag{5.3}$$

Note that all Gaussian membership functions in a single rule have a common standard deviation. The number of nodes is equal to the number of rules, K.

4) *Output Layer*: This layer consists of single node which sums up the prod-

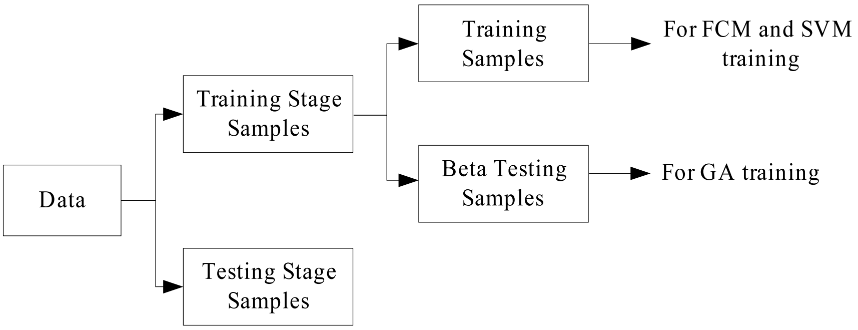


Figure 5.2: Data distribution for training and testing phases.

uct of firing strengths and weights with a bias term. The output takes the form:

$$\begin{aligned} O = & \sum_{r=1}^{K} w^r . f^r + b \\ = & \vec{w}^{\,T} . \vec{f} + b. \end{aligned} \tag{5.4}$$

For binary-class problem, given the data pair $(\vec{x}_i, y_i)$ where the labels, $y_i = \{-1, +1\}$, the decision function is:

$$g(\vec{x}) = sign(O) \tag{5.5}$$

The binary-class classifier above can be extended to multi-class case by using one-against-all scheme. The class labels are now considered as $y_i = \{1, \ldots, C\}$. In this scheme, C classifiers will be constructed, one for each class. The c^{th} classifier will be trained to classify the training data of class c against all other training data. The output for each classifier will be combined to give the final decision function:

$$g(\vec{x}) = \arg\max_c (O^c) \tag{5.6}$$

5.3 Training of EFSVM-FCM

As shown in Fig. 5.2, the whole set of data is divided into two groups: training stage samples and testing stage samples using two-fold cross selection. As the

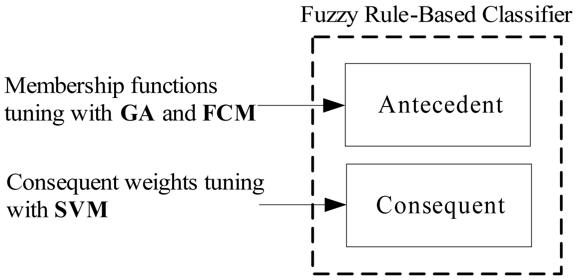

Fuzzy Rule-Based Classifier
Membership functions tuning with GA and FCM
Antecedent
Consequent weights tuning with SVM
Consequent

Figure 5.3: The learning of antecedent part with Genetic Algorithm (GA) and Fuzzy C-Means (FCM) algorithm, and consequent part with Support Vector Machine (SVM).

names suggest, the training stage samples are used during the classifier training stage while the testing stage samples are used during the testing stage. The training stage samples are further divided into the training samples and the beta testing samples using two-fold selection scheme. The training samples are employed during the training processes with FCM and SVM. On the other hand, the beta testing samples are employed during the training process with GA. The reason why GA employs the beta testing samples instead of the training samples is that SVM result will affect the fitness evaluation of GA which will lead to premature convergence.

Fig. 5.3 shows the learning of the fuzzy rule-based classifier. The training process is divided into two parts. The first part involves antecedent learning of the fuzzy rule. Genetic algorithm and Fuzzy C-Means clustering are used to generate the Gaussian membership function parameters and to determine the number of fuzzy rules. The second part involves consequent part learning by linear kernel based support vector machine. The learning details are delineated in Sections 5.3.1 and 5.3.2.

5.3.1 Antecedent Part Learning

One of the main challenges during the design of any fuzzy rule-based system is to determine the number of rules and to decide the specification of fuzzy membership functions. In the proposed framework, the number of rules, K is determined by heuristic while GA and FCM are used to generate the antecedent part Gaussian parameters (mean and standard deviation as given in (5.2)). Based on the classifier architecture shown in Fig. 5.1, there are $K \times m$ Gaussian mean parameters and K Gaussian standard deviation parameters need to be trained.

Initially, each class is divided into n_c $(c = 1, \ldots, C)$ clusters where C is the number of classes. Note that n_c are user defined parameters. Each class can have different number of clusters depending on the complexity of the data. For dense data, a small number of clusters is required. Conversely for sparse data, a larger number of clusters is required. The total number of rules, K is given by $\sum_{c=1}^{C} n_c$ which is normally much smaller than the support vectors in conventional SVMs or the number of rules generated using the input partition method reported in [78]. As a rule of thumb, n_c would be high for more challenging classification task and vice-versa.

Once the means of membership functions are found via FCM, GA is adopted to find the appropriate standard deviations. At the same time, the consequent weights are determined by SVM. This will be explained in Section 5.3.2. Subsequently, the fuzzy classifier is ready to be evaluated once on the beta testing samples to obtain the beta testing accuracy (fitness value). After each of the chromosomes is evaluated and associated with the respective fitness value, the current population undergoes the reproduction process to create the next generation of population. The "tournament with replacement" selection scheme is used to determine the members of the new generation population. When the evolution ends (meet stopping criteria or reach maximum number of generations), the best solution is selected.

5.3.2 Consequent Part Learning

This section starts with the basics of SVM and provides the connection between SVM with TSK FLS for a binary-class problem. The hybrid algorithm is then extended to multi-class problem using one-against-all scheme.

For a binary classification problem, let $S = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_N, y_N)\}$ represents a training data set, where $\vec{x}_i$, $i = 1, \ldots, N$ are vectors and $y_i \in \{-1, +1\}$ are the class labels. SVM tries to find an optimal hyperplane

$$\langle \vec{w}, \vec{x}_i \rangle + b = 0 \tag{5.7}$$

where $\vec{w} \in \Re^n$ and $b \in \Re$. For the linearly separable case, the optimal hyperplane is found by solving the following constrained optimisation problem:

$$\begin{array}{ll} \text{Min}_w & \frac{1}{2} \parallel \vec{w} \parallel^2 \\ \text{Subject to} & y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 1 \end{array} \tag{5.8}$$

For the linearly non-separable case, the constrained optimisation is rewritten as

$$\begin{array}{ll} \text{Min}_w & \frac{1}{2} \parallel \vec{w} \parallel^2 + C_0 \sum_i \xi_i \\ \text{Subject to} & y_i(\langle \vec{w}, \vec{x}_i \rangle + b) \geq 1 - \xi_i \end{array} \tag{5.9}$$

where C_0 is a user defined cost parameters to control trade-off between the training error and the margin while $\xi_i \geq 0$ is a slack variable. According to Kuhn-Tucker theorem, the problem above can be rearranged as:

$$\begin{aligned} \text{Min} \quad & \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \vec{x}_i, \vec{x}_j \rangle \\ \text{Subject to} \quad & \sum_i \alpha_i y_i = 0 \end{aligned} \tag{5.10}$$

where $0 \leq \alpha_i \leq C_0$, $i, j = 1, \ldots, N$, and those with $\alpha_i \neq 0$ are called support vectors. The decision function of a binary-class problem is finally:

$$\begin{aligned} g(\vec{x}) =& \text{sign}(\langle \vec{w}, \vec{x} \rangle + b) \\ =& \text{sign}\left(\sum_i \alpha_i y_i \langle \vec{x}, \vec{x}_i \rangle + b \right) \end{aligned} \tag{5.11}$$

After the firing strength, $\vec{f}$ for each rule is obtained from the antecedent parts, linear kernel SVM is ready for the learning of weights, $\vec{w}$ and bias, b in the output layer of EFSVM-FCM. The firing strengths and class labels form the new training sets given by:

$$\mathbf{Z} = \left\{ (\vec{f}(\vec{x}_1), y_1), (\vec{f}(\vec{x}_2), y_2), \ldots, (\vec{f}(\vec{x}_N), y_N) \right\} \tag{5.12}$$

From (5.11), the decision function can be reformulated:

$$g(\vec{x}) = sign\left(\sum_i y_i \alpha_i \langle \vec{f}(\vec{x}), \vec{f}(\vec{x}_i) \rangle + b \right). \tag{5.13}$$

The lagrange multipliers α_i are solved via (5.10). With this, (5.13) can be simplified to associate with (5.4):

$$\begin{aligned} g(\vec{x}) =& sign\left(\sum_i y_i\alpha_i \sum_r f^r(\vec{x})f^r(\vec{x}_i) + b\right) \\ =& sign\left(\sum_r\left(\sum_i y_i\alpha_i f^r(\vec{x}_i)\right)f^r(\vec{x}) + b\right) \\ =& sign\left(\sum_r w^r f^r(\vec{x}) + b\right) \\ =& sign(\vec{w}\ ^T.\vec{f} + b) \\ =& sign(O) \end{aligned} \tag{5.14}$$

where

$$w^r = \sum_{i\in SV} y_i\alpha_i f^r(\vec{x}_i). \tag{5.15}$$

As explained in Section 5.2, the binary-class classifier above can be extended to the multi-class case by using one-against-all scheme. Hence, the multi-class classifier decision function is given by:

$$\begin{aligned} g(\vec{x}) &= \arg\max_c\left(\sum_i y_i\alpha_i^c \sum_r f^r(\vec{x})f^r(\vec{x}_i) + b^c\right) \\ &= \arg\max_c\left(\sum_r\left(\sum_i y_i\alpha_i^c f^r(\vec{x}_i)\right)f^r(\vec{x}) + b^c\right) \\ &= \arg\max_c\left(\sum_r w^{r,c} f^r(\vec{x}) + b^c\right) \\ &= \arg\max_c\left(\vec{w}^c\ ^T.\vec{f} + b^c\right) \\ &= \arg\max_c(O^c) \end{aligned} \tag{5.16}$$

where

$$w^{r,c} = \sum_{i\in SV} y_i\alpha_i^c f^r(\vec{x}_i) \tag{5.17}$$

The equation above shows that all C classifiers share a set of common antecedent parts, this is denoted by a single firing strength vector, $\vec{f}$. This reduces the number of parameters being stored. On the other hand, each classifier will have its own output weight vector, $\vec{w}^c$ and bias, b^c. Hence, for the binary-class problem, there is a need to store K weights and one bias. The total number of parameters that includes antecedent Gaussian parameters is therefore $K(2+m)+1$. Likewise, for the multi-class problem, $K \times C$ weights and C biases are stored. Therefore, the total number of parameters for both antecedent and consequent parts increases to $K(m+1)+C(K+1)$.

5.4 Performance Evaluation

The effectiveness of the proposed EFSVM-FCM classifier was tested with four classification tasks. From UCI ML Repository [80]: Iris plant database, Wine recognition database, Bupa liver disorders database, and Glass identification database. Of the four databases, only the liver database is a binary class problem; the other databases belong to multi-class classification problems. A summary of the datasets is shown in Table 5.1. The iris plant dataset is a common benchmark in pattern classification studies. The database contains 50 measurements of four features from each of the three iris species: *Setosa*, *Versicolor*, *Virginica.* One class is linearly separable while two other classes are not linearly separable from each other. Next, the wine dataset contains chemical analysis performed on three types of wine produced in Italy from grapevines cultivated by different owners in one specific region. This is a well-posed classification problem. For the liver dataset, it is known to be highly nonlinear and hard to classify. Therefore the crossover rate, mutation rate and FCM parameter n_c are set to higher values. Glass data is another popular benchmark database. It consists of 163 window-glass data and 51 non-window glass data. The six-class problem has 9 continuous valued attributes.

For convenience sake, all attribute values in this chapter were normalised into real number between unit interval $[0, 1]$ as:

$$x_{pn} := \frac{x_{pn} - \min\{x_{n|\forall p}\}}{\max\{x_{n|\forall p}\} - \min\{x_{n|\forall p}\}} \tag{5.18}$$

where $p = 1, \ldots, N,\ n = 1, \ldots, m$. Therefore, the C-class classification problem is defined in the m-dimensional unit cube $[0, 1]^m$. Two-fold cross validation (2CV) and one-against-all classification approach are implemented. Since the testing accuracy in 2CV relies on the initial division of the data, the testing is performed in ten iterations. The cost parameter, C_0 in both EFSVM-FCM and conventional SVM were set to 4096 during training stage. GA was allowed to run for 30 generations and the population size was set to 50. Binary coded GA was implemented in the current framework. Each of the standard deviations was encoded in a 8-bit string. Hence, each chromosome has a total of $8 \times K \times m$ bits. A simple repair function can be carried out to ensure proper overlapping between the evolved membership functions as to avoid the case where the input space is not wholly spanned by fuzzy rule "patches". During the fitness evaluation, the parameters were decoded into real numbers using linear mapping equation as shown below:

$$g_p = G_q^{min} + (G_q^{max} - G_q^{min}) \times \frac{A_q}{2^N - 1} \tag{5.19}$$

where g_p denotes the actual value of the q^{th} parameter, A_q denotes the integer represented by a N-bit string gene, G_q^{max} and G_q^{min} denote the user defined upper and lower limits of the gene respectively. As for the genetic operators, bitwise flipping mutation with the mutation rate, MR and single-point crossover with the crossover rate, CR were implemented. Fuzzifier parameter, m in FCM was fixed at 2. For the rest of the parameters, they are listed in Table 5.2.

Testing results are shown in Tables 5.3-5.6. Comparisons with the results from the conventional SVM classifier (with Radial Basis Function kernel) and some other popular algorithms such as Fuzzy Rule-Based Classifier, Positive Definite Fuzzy Classifier (PDFC), Self-Organizing TS-Type Fuzzy Network with Support Vector Learning (SOTFN-SV), Inverted Hierarchical Neuro-Fuzzy BSP (HNFB^{-1}), Genetic Fuzzy and Neural Networks (NN) are also provided in the tables.

Table 5.1: Summary of Datasets

Dataset	No. of samples	No. of classes	No. of attributes
Iris	150	3	4
Wine	178	3	13
Liver	345	2	6
Glass	214	6	9

Table 5.2: EFSVM-FCM Parameters Used for Classification Tasks

Parameter	Dataset			
	Iris	Wine	Liver	Glass
CR	0.8	0.8	0.9	0.8
MR	0.03	0.03	0.1	0.03
n_c	2	2	10	2
K	6	6	20	12

As seen from the results in Tables 5.3-5.6, the proposed classifier not only capture the essence of SVM – good generalisation, it also manages to outperform conventional RBF kernel based SVM. While comparing with other fuzzy rule-based classifier in the literature (refer to Table 5.2), the number of rules, K generated in

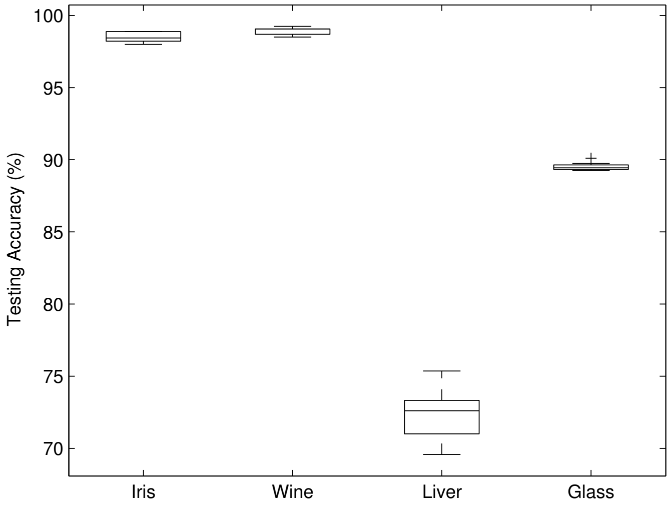


Figure 5.4: Boxplot for testing accuracies of four classification tasks with 10 iterations for each task. Two-fold cross validation method is used.

current framework is relatively small. Furthermore, the number of parameters is less than those obtained with the conventional SVM. As a result, EFSVM-SVM is a fast and memory efficient algorithm. As mentioned, the number of clusters for each class can be different to achieve better result. However, in all presented classification problems, each class is partitioned into same number of clusters (i.e., $n_1 = n_2 = \cdots = n_C$) for simplicity. So far, the results are positive with this scheme.

Fig. 6.6 shows the boxplot for each classification problem. The testing accuracies are quite consistent except for Bupa liver dataset due to its highly non-linear characteristic.

5.5 Conclusion

EFSVM-FCM is the realisation of the functionalities of four popular soft computing techniques, namely GA, FCM, FLS and SVM. It was successfully applied to

Table 5.3: Classification Results of Iris Data with Various Methods

Methods	No. of Rules	Accuracy (%)
EFSVM-FCM[a]	6	98.47
SVM (RBF Kernel)[a]	NA	96.11
Fuzzy Rule (Ishibuchi's Method)[a,b]	47.1	95.47
Fuzzy Rule (Mansoori's Method)[a,b]	47.1	95.47
PDFC[a] [76]	35.46	96.38
SOTFN-SV[a] [78]	14	98.40

[a] Results of two-fold cross validation

[b] Results reported in [81]

Table 5.4: Classification Results of Wine Data with Various Methods

Methods	No. of Rules	Accuracy (%)
EFSVM-FCM[a]	6	98.93
SVM (RBF Kernel)[a]	NA	97.60
Fuzzy Rule (Ishibuchi's Method)[a,b]	137	95.45
Fuzzy Rule (Mansoori's Method)[a,b]	137	95.56
HNFB^{-1} [a,c] [82]	27	94.44

[a] Results of two-fold cross validation

[b] Results reported in [81]

[c] Results reported in [82]

Table 5.5: Classification Results of Liver Data with Various Methods

Methods	No. of Rules	Accuracy (%)
EFSVM-FCM[a]	20	72.38
SVM (RBF Kernel)[a]	NA	58.84
Genetic Fuzzy SVM[a] [75]	NA	70.80
HNFB^{-1} [a,b] [82]	142	73.33
NN[a,b]	NA	60.50

[a] Results of two-fold cross validation

[b] Results reported in [82]

Table 5.6: Classification Results of Glass Data with Various Methods

Methods	No. of Rules	Accuracy (%)
EFSVM-FCM[a]	12	89.51
SVM (RBF Kernel)[a]	NA	87.52
Genetic Fuzzy [31]	NA	64.40
Fuzzy Rule (Ishibuchi's Method)[a,b]	42.3	60.93
Fuzzy Rule (Mansoori's Method)[a,b]	42.3	61.22

[a] Results of two-fold cross validation

[b] Results reported in [81]

four benchmark classification problems. Based on the qualitative results, the algorithm outperforms most popular algorithms. The main advantage of the proposed classifier is the capability to achieve good generalisation inherited from SVM with a compact rule base. As a result, the memory requirement to store the trained parameters is reduced tremendously. This chapter also shows that combination of empirical and structural risks minimisation can deliver a high performance hybrid classifier. While the proposed algorithm seems promising, some limitations exist. The first weakness is the slow training process of GA, an unavoidable tradeoff when GA is adopted. Secondly, the fuzzy rule-base might not be easily interpretable because local fuzzy rules (rule-specific membership functions) are adopted [83] to achieve more design flexibilities. Since the results are promising, future work includes performance enhancement by fine tuning the cost parameter of SVM in the consequent training stage and also to vary the number of sub-clusters for each class in the antecedent training stage.

Chapter 6

On Improving K-Nearest Neighbor Classifier with Fuzzy Rule-Based Initialisation

One of the greatest challenges that hinder the adoption of automated pattern classification systems is the lack of complete training dataset. When “holes” exist in the training data, the decision area formed is unlikely to actually represent the underlying data distribution, thereby affecting the classification accuracy. Research has shown that the performances of conventional crisp and fuzzy K-Nearest Neighbor (K-NN) algorithms trained using finite samples tends to be poor [84], [85]. To address the need to extract more useful information from the limited training samples, a fuzzy rule-based K-NN algorithm was introduced in this chapter. The main feature differentiating our proposed algorithm from the conventional fuzzy K-NN algorithm is a fuzzy rule-based initialisation procedure that allows imprecise inputs (neighborhood density and distance) to be handled through the framework of fuzzy logic system. Furthermore, the membership functions of the fuzzy rule-based K-NN can be tuned to produce a highly versatile decision boundary. This comparative advantage over fuzzy K-NN is demonstrated using a synthetic dataset in two-dimensional space. Another issue that was investigated is the use of weighted Euclidean distance measurement, together with fuzzy rule-based K-NN algorithm, to overcome the curse of dimensionality [86], [87]. The Euclidean distance weights and the parameters of the fuzzy rule-based system were optimised with Genetic Algorithm (GA) simultaneously. The practical applicability of the proposed algorithm was verified using four UCI datasets (Bupa liver disorders, Glass, Pima Indians diabetes and Wisconsin breast cancer) that have very limited training data. Next, the algorithm was applied to Ford automotive dataset whereby noise corruption caused discrepancies to exist between the training and testing data. The efficacy of the proposed method was also compared against Support Vector Machine (SVM) and Bayesian classifiers.

6.1 Introduction

K-nearest neighbor (K-NN) algorithm is a simple non-parametric classifier that assigns class labels to the input patterns based on the class labels of the K-nearest neighbors. For a 1-NN classifier trained using infinite samples, the error rate is bounded above by twice of optimal Bayes error rate [88], [89]. This asymptotic behavior together with its simplicity in implementation is behind its popularity among various classifiers. In addition, it does not need any a priori knowledge about the structure of the training set. Nevertheless, the K-NN algorithm has a major drawback: each prototype is considered equally important in the assignment of input patterns. Keller et al. [90] extended the K-NN algorithm by incorporating fuzzy concepts. The algorithm, fuzzy K-NN, assigns a class membership [0,1] to an input pattern rather than a crisp class label as in K-NN. Similar to K-NN, fuzzy K-NN also searches the labeled prototypes for the K-nearest neighbors. However, in fuzzy K-NN the labeled prototypes need to be preprocessed whereby an initial membership grade will be assigned to each training data point. This procedure ensures the prototypes are more informative. The more accurate the initial membership grade is, the more accurate the classification result will be. As far as the performance is concerned, Kuncheva [2], with her extensive comparative experimental results, suggests that fuzzy K-NN does not necessarily perform better than K-NN but the results are problem-dependent. Despite claiming as to be a fuzzy variant of K-NN, the algorithm does not actually exploit fuzzy set theory other than the simple notion of a fuzzy set. Therefore, the classifier may be unable to handle uncertainties present in the problem. Such uncertainties could be due to overlapping classes, uneven class distribution [91] or imprecision in training data.

In both K-NN and fuzzy K-NN algorithms, the choice of parameter K implicitly defines the size of the neighborhood that is used to predict the class of a test pattern. The classifier performance depends on the selection of K [92]. Sometimes, a small K (with respect to the number of samples) is desirable so that the training data used to determine the class label of a test point are close enough to provide an accurate prediction. However, classifiers based on small K are susceptible to noise in the training data. While a large K reduces sensitivity to noise, the information becomes too global. There is a higher likelihood that the neighborhood (especially in the region where two classes overlap) consists of patterns from more than one class, thereby increasing the difficulty in assigning a class label to the test pattern. Most of the existing K-NN classifiers assume precise data in the sense that exact measurement is achievable. However, in most real-world scenarios, this assumption is not valid so there is always going to be a degree of uncertainty. Motivated by the capability of fuzzy rule-based system to handle the uncertainty and the ambiguity inherent in many problems, we propose to incorporate fuzzy rules into the K-NN algorithm. The use of linguistic variables and fuzzy IF-THEN rules exploits the tolerance for imprecision and uncertainty. A fuzzy rule-based initialisation step, which relies on the imprecise neighborhood distance and density information of the training prototypes to assign membership grades, is introduced. The imprecision in the distance and density information is captured by the fuzzy set. Moreover, the fuzzy rule-based initialisation provides the potential to create a flexible decision area which is not achievable with either crisp or fuzzy K-NN. Besides that, we also use weighted Euclidean distance to cope with heterogeneous input space [93] which is considered as one of the limitations of K-NN algorithm. This improves the classifier's performance by ensuring the distance computation varies with same proportion in every direction in the feature space. A high performance and interpretable fuzzy rule-based K-NN (FRB-KNN) classifier can be fine tuned with evolutionary optimisation algorithm such as Genetic Algorithm (GA).

The remainder of this chapter is structured as follows. Section 6.2 provides background information on the crisp and fuzzy K-NN algorithms. In Section 6.3, the proposed fuzzy rule-based K-NN algorithm and the rationale for using weighted

Euclidean measure are presented. Section 6.4 shows how the proposed algorithm is tuned with GA. Section 6.5 provides the experimental backdrop to the performance assessment of our proposed fuzzy rule-based K-NN algorithm and presents the results. Finally, Section 6.6 concludes this chapter.

6.2 The Crisp and Fuzzy K-NN Algorithms

The crisp and fuzzy K-NN algorithms are reviewed briefly in order to set the context for introducing the proposed fuzzy rule-based variant in the next section. The basis of K-NN algorithm is to classify an unknown pattern to the class most represented by its K nearest neighbors. Conventional crisp K-NN algorithm requires no preprocessing of the labeled prototypes prior to their use whereas fuzzy K-NN algorithm pre-assigns a membership grade to each labeled prototype during initialisation process. In practice, K is usually chosen to be odd for two-class problems, so as to avoid ties. When there are more than two classes, a tie can be handled as follows: the input data is assigned to the class, of those classes that tied, for which the sum of distances from the input data to each neighbor in the class is minimum. If the tie remains unsolved, the input data is assigned to the class of last minimum distance found.

6.2.1 The Conventional Crisp K-NN Algorithm

Let $S = \{x_1, x_2, \cdots, x_n\}$ be a set of n labeled samples. The conventional crisp K-NN algorithm is as follows:

```
BEGIN
  Input $y$, of unknown classification.
  Set $K$, $1 \leq K \leq n$.
  Set $i = 1$.
  DO UNTIL ($K$-nearest neighbors found)
    Compute distance from $y$ to $x_i$.
    IF ($i \leq K$) THEN
      Include $x_i$ in the set of $K$-nearest neighbors
    ELSE IF ($x_i$ is closer to $y$ than any previous nearest neighbor) THEN
      Delete farthest in the set of $K$-nearest neighbors
      Include $x_i$ in the set of $K$-nearest neighbors
    END IF
    Increment $i$.
  END DO UNTIL
  Determine the majority class represented in the set of $K$-nearest neighbors.
  IF (a tie exists) THEN
    Compute sum of distances of neighbors in each class which tied.
      IF (no tie occurs) THEN
        Classify $y$ in the class of minimum sum
      ELSE
        Classify $y$ in the class of last minimum found.
      END IF
  ELSE
    Classify $y$ in the majority class.
  END IF
END
```

6.2.2 Fuzzy K-NN Algorithm

Unlike crisp K-NN, fuzzy K-NN algorithm assigns class membership grade to an input data rather than class label. The membership assignment is based on the distance between the input data and its K-nearest neighbors and those neighbors' memberships (pre-computed during initialisation process) in the possible classes.

Let $S = \{x_1, x_2, \cdots, x_n\}$ be a set of n labeled samples. Also, let $\mu_i(x)$ be the membership assigned to the input data x, and $\mu_{ij}(x)$ be the membership in the ith class of the jth labeled prototype. The algorithm is as follows:

```
BEGIN
  Input $x$, of unknown classification.
  Set $K$, $1 \leq K \leq n$.
  Prototype Membership Initialisation
  Set $i = 1$.
  DO UNTIL ($K$-nearest neighbors to $x$ found)
    Compute distance from $x$ to $x_i$.      IF ($i \leq K$) THEN
      Include $x_i$ in the set of $K$-nearest neighbors
    ELSE IF ($x_i$ closer to $x$ than any previous nearest neighbor) THEN
      Delete the farthest of the $K$-nearest neighbors
      Include $x_i$ in the set of $K$-nearest neighbors.
    END IF
  END DO UNTIL
  Set $i = 1$.
  DO UNTIL ($x$ assigned membership in all classes)
    Compute $\mu_i(x)$ using (6.1).
    Increment $i$.
  END DO UNTIL
END
```

where

$$\mu_i(x) = \frac{\sum_{j=1}^{K} \mu_{ij} \Big(1/ \parallel x - x_j \parallel^{2/(m-1)} \Big)}{\sum_{j=1}^{K} \Big(1/ \parallel x - x_j \parallel^{2/(m-1)} \Big)} \tag{6.1}$$

The prototype membership initialisation step is defined as:

$$\mu_j(x) = \begin{cases} 0.51 + (n_j/K) * 0.49 & \text{if } j = i \\ (n_j/K) * 0.49 & \text{if } j \neq i \end{cases} \tag{6.2}$$

where n_j is the number of the neighbors which belong to the jth class.

6.3 Fuzzy Rule-Based K-NN

As pointed out in [84], [85], the performance of K-NN classifier based on a finite number of training samples is not guaranteed. The situation is even worse when there are insufficient training data to represent the underlying data distribution. This always happens when the cost to collect training data is expensive. In addition, the existence of noise in either training or testing data can further degrade the classifier performance. The initialisation step of conventional fuzzy K-NN suffers from these drawbacks. The algorithm assumes there are sufficient training data to provide meaningful information for further classification, which is not the case in most real-world problems. With respect to this problem, the approach we propose here is discriminative in that it assumes all the training data are uncertain. By introducing fuzzy rule-based initialisation step, we can exploit the fuzzy inferencing mechanism to manage the imprecise inputs derived from the data (i.e., the distance and density information of the neighborhood). The proposed initialisation algorithm is as follows:

```
BEGIN
  FOR each training prototype, $x_i$
    FOR each class, $Class_j$
      - Compute the class density, $v_{ij} = n_j/K$, where $n_j$ denotes the number of
        neighbors which belong to $j$th class.
      - Compute the average distance of the neighbors
          $$d_{ij} = \begin{cases} \frac{\sum_{r=1}^{n_j} \|x_i - x_r\|^2}{n_j} & \text{for } n_j \neq 0 \\ 1 & \text{for } n_j = 0 \end{cases} \qquad (6.3)$$
      - Perform fuzzification (singleton/nonsingleton) for inputs $v$ and $d$.
      - Compute membership grade, $\mu_{i,j}$ from the fuzzy rules
        If $v$ is $\widetilde{A_{Vt}}$ and $d$ is $\widetilde{A_{Dt}}$, then $z$ is $C_m$
          $$\mu_{i,j} = \frac{\sum_{m=1}^{M} f^m z^m}{\sum_{m=1}^{M} f^m}, \qquad z = [\underline{z}, \bar{z}] \qquad (6.4)$$
        where
          $$z = \begin{cases} \bar{z} & \text{if } Class_j = \text{desired class} \\ \underline{z} & \text{otherwise} \end{cases} \qquad (6.5)$$
```

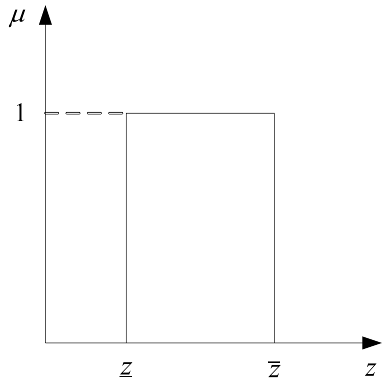


Figure 6.1: Interval type-1 fuzzy set.

$\underline{z}$ denotes the lower bound of the interval type-1 consequent set while $\bar{z}$ is the upper bound (see Fig. 6.1).

t is the fuzzy partition index, M is the total number of rules and f^m is mth rule's antecedent part firing strength

$$f = \sup(\mu_{\widetilde{V}}, \mu_{\widetilde{A}_V}). \sup(\mu_{\widetilde{D}}, \mu_{\widetilde{A}_D}) \tag{6.6}$$

where sup[.] denotes supremum operation [37].

```
    END FOR LOOP
  END FOR LOOP
END
```

Consider an initialisation procedure where the universe of discourse for the data density has two fuzzy partitions (i.e., $\widetilde{A}_V \in$ {HIGH, LOW}), the distance domain has also two fuzzy partitions (i.e., $\widetilde{A}_V \in$ {FAR, CLOSE}) and consequent has four fuzzy partitions (interval type-1) (i.e., $C \in$ {HIGH, MODERATE HIGH, MODERATE LOW, LOW}), then a four rules rule-base can be formed as follows:

If v is HIGH and d is CLOSE, then z is HIGH

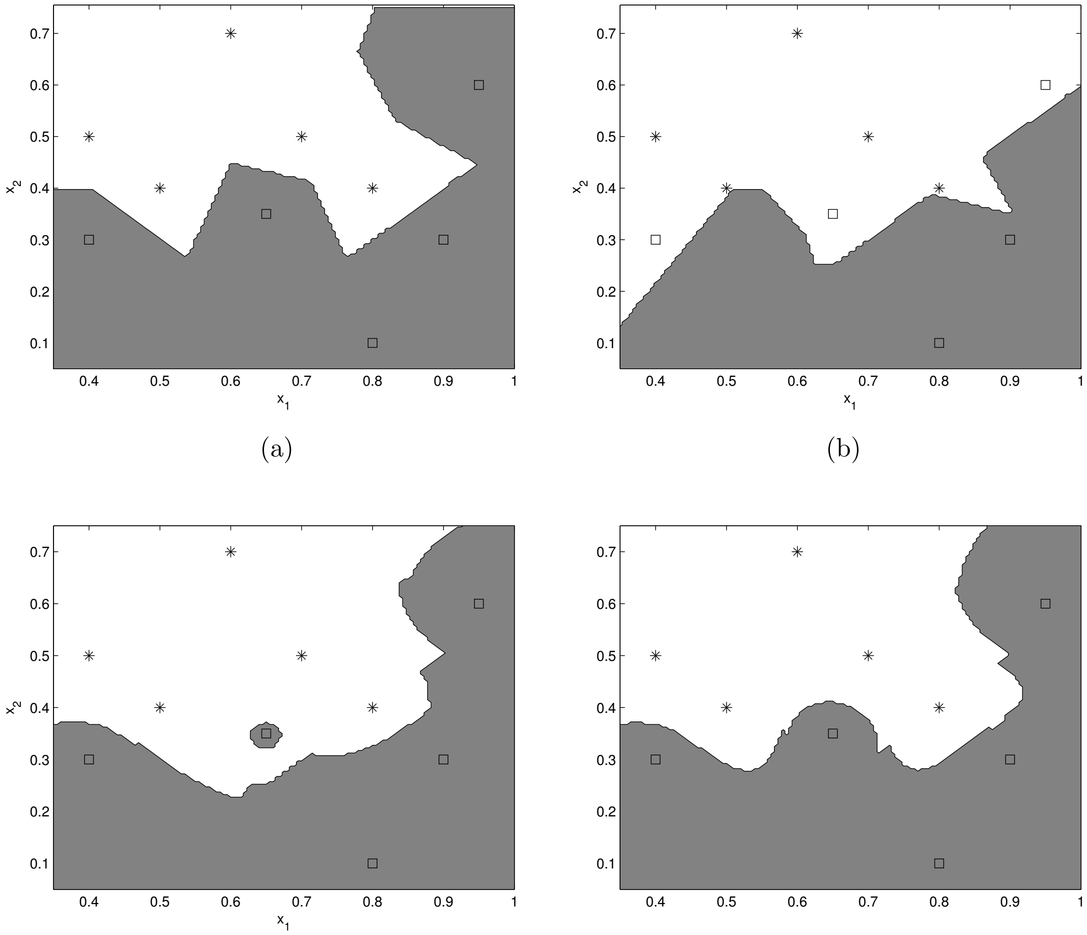
If v is HIGH and d is FAR, then z is MODERATE HIGH

If v is LOW and d is CLOSE, then z is MODERATE LOW

If v is LOW and d is FAR, then z is LOW.

The reason for employing interval type-1 fuzzy set as the consequent set is to ensure higher weight is assigned to the training data in its own class while lower weight is assigned to other classes. When compared to fuzzy K-NN, one important difference is that the summation of memberships in all classes is not equal to one. This is common to most fuzzy rule-based system as the tweaked membership functions of differing classes are not complementary to each other. Nevertheless, it should be the least concern as winner-takes-all approach is used for classification whereby we only care about the relative difference in membership grades.

Our proposed method revises and enhances the fuzzy K-NN algorithm by tuning the fuzzy antecedent and consequent sets in the rule based used to initialise the membership grades of the data points in the training set. The rest of the algorithm is similar to fuzzy K-NN as described in Section 6.2.2. The inclusion of fuzzy rule-based system in the initialisation can ensure richer context information is captured and hence achieve a more suitable decision area. In conventional fuzzy K-NN or crisp K-NN, the decision area is fixed once the parameter K has been determined. These algorithms rely on the assumption that class conditional probabilities are locally constant. On the other hand, the highly versatile and ad-hoc framework of fuzzy rule-based system enable us to have a flexible decision area. The flexibility comes from the ability to modify the membership functions; which in turn change the interpretation of the input variables such as density and neighborhood distance. The potential advantage of the proposed fuzzy rule-base K-NN classifier was demonstrated using a problem with limited training samples. Fig. 6.2 shows the two-dimensional decision area of a two-class classification problem with ten data points. Recall that in K-NN algorithm, the choice of parameter K is critical to ensure good classification. When $K = 1$, there will be no ambiguity as shown by the decision area produced by either crisp K-NN or fuzzy K-NN in Fig. 6.2(a). All data points are correctly classified and the decision area is continuous. However, there is uncertainty in assigning class labels to the samples when K is increased to 3. The points in the area around (0.65, 0.35) are closer to the opposing class (Class '*'). Hence, this point may be regarded as an outlier; but on the other hand, it seems more intuitive to classify this uncertain area to Class '□'. Crisp K-NN (Fig. 6.2(b)) produces erroneous decision boundary as three of the data points from Class '□' are misclassified. While both fuzzy K-NN (Fig. 6.2(c)) and fuzzy-rule based K-NN (Fig. 6.2(d)) manage to classify all data points correctly, fuzzy rule-based K-NN seems to produce a more desirable decision area. Fuzzy K-NN produces a disjoint decision area around point (0.65, 0.35) because most of the data points in this area receive more influences from the three nearest neighbors which belong to another class. However, with proper tuning of membership functions a continuous (in fact smoother) decision area can be formed which resembles the case where $K = 1$. As far as the flexibility is concerned, the proposed fuzzy rule-based K-NN is also able to reproduce the same decision area as in the conventional fuzzy K-NN case while offering an alternative to fine tune the decision boundary further. This simple example elucidates our proposed method has the potential to better handle the uncertainty in classification problems.



(a)

(b)

(c)

(d)

Figure 6.2: Decision area computed with different classifiers (a) $K = 1$, (b) – (d) $K = 3$ for crisp K-NN, fuzzy K-NN, and fuzzy rule-based K-NN respectively. To illustrate the effectiveness of fuzzy rule-based initialisation procedure only, weighted Euclidean distance measurement is not used. It is clear that the decision area produced by fuzzy rule-based K-NN resembles the one with $K = 1$ with minimal uncertainty.

6.3.1 Weighted Euclidean Distance Measure

Classification algorithms based on simple Euclidean distance measure, such as K-NN and fuzzy K-NN, assume that the input space is isotropic or homogeneous. However, in most real classification problems, the input patterns tend to be heterogeneous [86]. Variations in the structure of the input space can lead to severe bias, especially when the feature dimension is high. One method to minimise the impact of non-homogeneous input patterns is to manipulate the original input space by modifying the importance of each attribute to reflect its relevance for classification. In particular, the neighborhoods are elongated along less relevant attribute dimensions (with small weight) and constricted along most relevant one (with large weight) [87]. This work employed the weighted Euclidean distance measure, defined below, to control the effect of each individual feature

$$D(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{q} w_i (x_i - y_i)^2} \tag{6.7}$$

where q is the number of features and w_i is the weight for ith feature. Consider the example in Fig. 6.3. Based on the principle of nearest-neighbor, the query point x_0 will be assigned to the same class as x_2 because $D_1 > D_2$. Suppose the desired decision is to assign x_0 to the label of x_1, then the underlying distance relationship between the prototypes x_1 and x_2 should be adjusted such that $D_1 < D_2$. It can be verified that this is achievable as long as the condition in (6.8) is satisfied where $\delta_{1i} = (x_{1i} - x_{0i})$, $\delta_{2i} = (x_{2i} - x_{0i})$ for $\forall i$. By solving (6.9) for $\frac{w_1}{w_2}$ subject to (6.8), x_0 will be assigned to the class represented by x_1 when $\frac{w_1}{w_2} > 1.477$. For example, if $w_1 = 0.8$ and $w_2 = 0.5$, then ${D_1}^2 = 0.0089$ and ${D_2}^2 = 0.0093$.

$$\min_{\forall i} \frac{|\delta_{1i}|}{|\delta_{2i}|} < 1 \tag{6.8}$$

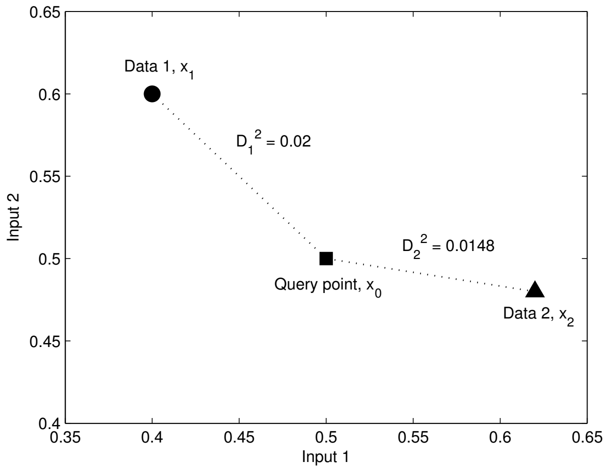


Figure 6.3: When the Euclidean distance measure is unweighted, the query point is assigned to the same class as data 2 as both of them are closer to each other.

$$\sum_{i=1}^{q} w_i {\delta_{1i}}^2 < \sum_{i=1}^{q} w_i {\delta_{2i}}^2 \tag{6.9}$$

While the numerical example demonstrates that the feature space can be adjusted, the task of solving (6.9) is very tedious when the feature dimension is higher than two. In order to alleviate this difficulty, the genetic algorithm described in the next subsection is used to evolve the feature weights and the parameters of the fuzzy rule-base simultaneously. As a result, an optimal set of feature weights can be obtained even when the data from different classes are overlapped.

6.4 Genetic Learning of Fuzzy Rule-Based K-NN

A fuzzy rule-based system has no self-learning capability. This does not hinder the usefulness of fuzzy system as advances in machine learning techniques have opened up a wide range of learning algorithm choices. Tuning methods such as Least-Square, Back-Propagation, Singular-Value-QR Decompositions etc. can be found in the literature [37]. To obtain a set of optimal parameters (membership functions, weights), a genetic algorithm based method was employed.

Binary coded GA was adopted in the current framework. Each of the parameter was encoded in a 8-bit string. The pre-evaluation training accuracy (see Section 6.5) was chosen as the fitness function. The chromosome structure is shown in Fig. 6.4. The chromosome contains the parameters that need to be optimised. Based on the four fuzzy rules structure defined in Section 6.3, there are four antecedent sets and four consequent sets. Each of the antecedent set is a sigmoidal membership function and is characterised by two parameters a and c as shown in (7.10). The interval type-1 consequent set is characterised by two parameters: lower bound $\underline{C}$ and upper bound, $\bar{C}$. In (6.4), the bounds for interval type-1 consequent sets are represented by the parameter z. The number of feature weights is equal to the dimension of the feature vector, q. Thus, there will be a total of $(16 + q)$ parameters need to be tuned.

$$f_{sig}(x) = \frac{1}{1 + e^{-a(x-c)}} \tag{6.10}$$

During the fitness evaluation, the parameters were decoded into real numbers using linear mapping equation as shown below:

$$g_p = G_q^{min} + (G_q^{max} - G_q^{min}) \times \frac{A_q}{2^N - 1} \tag{6.11}$$

where g_p denotes the actual value of the q^{th} parameter, A_q denotes the integer represented by a N-bit string gene, G_q^{max} and G_q^{min} denote the user defined upper and lower limits of the gene respectively. The selection method is tournament size of two with elitism. As for the genetic operators, bitwise flipping mutation (mutation rate = 0.03) and single-point crossover (crossover rate = 0.8) are implemented.

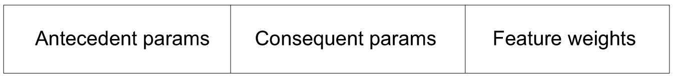


Figure 6.4: The structure of the chromosome. First part encodes the parameters for the antecedent sets while the middle part encodes the consequent parameters which describe a set of interval type-1 fuzzy sets. The last part contains the feature weights used in weighted Euclidean distance measure.

6.5 Computational Experiments

In our experiments, we used five different real-world datasets. Four datasets: Bupa liver, Glass, Pima Indians diabetes and Wisconsin breast cancer, were taken from the UCI Machine Learning Repository [94]. The last dataset – Ford automotive, was obtained from [70]. The characteristics of the five datasets are summarised in Table 6.1. All datasets were randomly divided into training sets (200 for Bupa liver, 120 for Glass, 400 for Pima Indians diabetes, 250 for Wisconsin breast cancer and 3306 for Ford) and test sets consisting of the remaining data points, as shown in Table 6.1. 50% hold-out cross-validation was performed during the classifier training stage. This means that only half of the training data will be used to train the classifier, while the other half will be used to pre-evaluate the classifier performance. Hold-out cross-validation was employed because the training accuracy tends to be 100% if the initialisation step is performed on the training prototypes and this leads to premature convergence during evolution. Since GA is a heuristic search based algorithm, for each hold-out dataset, the optimisation was performed ten times and the individual testing accuracy was averaged. The hold-out process was then repeated four times independently and the average cross-validation classification rate was reported. The experiment setup described above was designed to evaluate the performance of fuzzy rule-based K-NN clas-

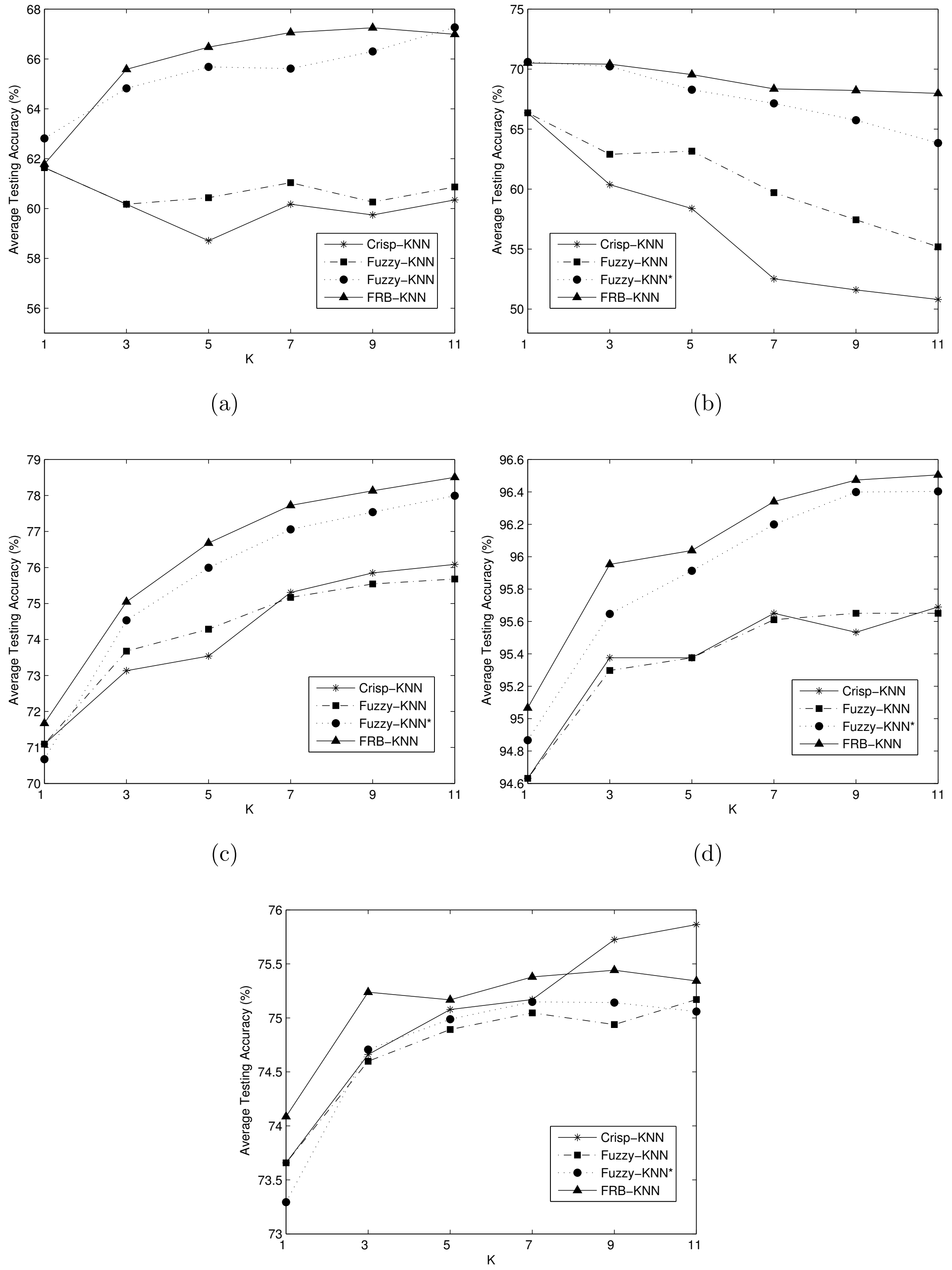
Table 6.1: Summary of Datasets

Dataset	Features	Classes	Total Samples	Training Samples	Testing Samples
Bupa Liver	6	2	345	200	145
Glass	9	7	214	120	94
Pima Indians Diabetes	8	2	768	400	368
Wisconsin Breast Cancer	30	2	569	250	319
Ford	2	2	4116	3306	810

sifier in two ways. Firstly, the UCI dataset poses a problem where the training data is insufficient. Secondly, the Ford dataset poses another problem whereby the testing data is atypical to the training data due to some disturbances. In the following, four competing K-NN variants are compared: crisp K-NN, fuzzy K-NN, fuzzy K-NN with weighted Euclidean distance measure (denoted as fuzzy K-NN* with an asterisk sign hereafter) and the proposed fuzzy rule-based K-NN. In all experiments, all data were first normalised into the range [0,1].

6.5.1 Minimising the Effect of Insufficient Training Data

An insufficiency of training data often results in a poorly learned classifier. The experiments were carried out to investigate if the proposed fuzzy rule-based initialisation procedure can mitigate the adverse effect of incomplete training data on the classification accuracy. Four UCI datasets were used in the study and the data was split in such a way that the number of training data is much smaller than the number of testing data. The ratios of training data to testing data for each training stage varied from approximately 40% to 70% (Bupa Liver 68.97%, Glass 63.83%, Pima Indians diabetes 54.35%, and Wisconsin breast cancer 39.19%). Fig. 6.5 shows the average classification accuracies for different K sizes, ranging from 1 to 11. In general, all NN based classifiers perform better when K is increased except for the Glass dataset due to the uneven class distribution as the data ratio of Class 1 to Class 6 is 76 : 9. Furthermore, the dataset is dominated by Class 1 and Class 6 which contribute 68.22% of the total number of samples. Due to the nature of this dataset, there is a drastic performance drop (K is increased) when using crisp K-NN followed by fuzzy K-NN. Even so, the proposed fuzzy rule-based K-NN is least affected by the uneven class distribution. Another observation is that the performance of the fuzzy K-NN algorithm is not necessarily better than the crisp K-NN algorithm throughout all experiments. For Liver and Glass datasets, the fuzzy K-NN may have the edge but not for the rests. This finding is consistent with the works by Kuncheva [2]. Table 6.2 presents the improvement of the fuzzy K-NN* and FRB-KNN classifiers over the conventional fuzzy K-NN on UCI dataset. The results reveal that the evolutionary optimised Euclidean distance weights can greatly improve the overall performance. This is due to the fact that not all features share the same class-discriminating power. By enhancing the more important feature and lowering the impact of irrelevant feature, the classification task can be easier. In addition, it is observed that the fuzzy rule-based initialisation procedure enhances the overall performance further (Bupa Liver +0.45%, Glass +2.00%, Pima Indians diabetes +0.66%, and Wisconsin breast cancer +0.16%). This elucidates the efficacy of fuzzy rule-based system to tweak the decision area in contrast to the fixed decision area produced by other competitors when the parameter K is predetermined.



(a)

(b)

(c)

(d)

(e)

Figure 6.5: Comparison of average testing accuracies with different K-NN algorithms for dataset (a) Bupa liver, (b) Glass, (c) Pima Indians diabetes, (d) Wisconsin breast cancer and (e) Ford automotive. In overall, fuzzy rule-based K-NN outperforms other NN variants.

Table 6.2: The Classification Accuracy Improvement of Fuzzy K-NN with Weighted Euclidean Distance (Fuzzy KNN*) and Fuzzy Rule-Based K-NN (FRB-KNN) Compared to Conventional Fuzzy K-NN on four UCI Datasets.

Dataset	Improvement (in %)	
	Fuzzy KNN*	FRB-KNN
Bupa Liver	4.68	5.13
Glass	6.85	8.85
Pima Indians Diabetes	1.39	2.05
Wisconsin Breast Cancer	0.53	0.69

6.5.2 Handling the Issue of Noise Uncertainty

The Ford dataset (as seen in Chapter 4) poses another interesting problem whereby the testing data and training data have dissimilar characteristics because of the presence of noises and disturbances in the test stage. By applying Principal Component Analysis (PCA) on the periodogram signal, the two most dominant principal components are extracted. Fig. 4.6 shows the scatter plots of the first and second principal components of PCA projected training data. Both the training and testing data have no clear decision boundary, but the testing data are more overlapped in comparison. This poses a real challenge to the classifiers to resolve "fuzzy" decision boundary. Fig. 6.5(e) reveals that the advantage of weighted Euclidean distance is virtually non-existing on Ford automotive dataset probably because the original data have undergone PCA projection. PCA has a similar effect of distorting the original feature spaces. The original data are weighted using eigenvectors (directions in which the variances of the projection is maximised). Since the effect of GA evolved feature weight is negligible,it is deduced that all the improvement is solely contributed by the fuzzy rule-based initialisation procedure.

Based on the experimental results on five datasets, the ranking of the classifier performance in descending order is given as follows: FRB-KNN, fuzzy K-NN*, fuzzy K-NN, and crisp K-NN. The findings indicate that the proposed system is superior to other K-NN classifiers not only for its ability to alleviate the adverse effect of insufficiency of training data, but also for its noise handling capability.

To further evaluate the proposed fuzzy rule-based K-NN classifier, its performance is compared against support vector machine (SVM) with Gaussian kernel (with default width, $\sigma = 0.5$) and Bayesian classifier. The Bayesian classifier consists of several conditional probability models which are approximated by Gaussian probability density function

$$p(\mathbf{x}|Class_j) \sim N(\mathbf{x}; \mathbf{m}_j, \Sigma_j) \tag{6.12}$$

where $\mathbf{m}_j$ and Σ_j denote the mean vector and covariance matrix of the multivariate Gaussian distribution associated with $Class_j$. Given an unlabeled input $\mathbf{x}'$, its label is associated with the class with maximum log-likelihood [17]

$$\begin{aligned} Class(\mathbf{x}') &= \arg\max_j \quad \log p(\mathbf{x}'|Class_j) \\ &= \arg\max_j \quad -\log|\Sigma_j| - (\mathbf{x}' - \mathbf{m}_j)^t \Sigma_j^{-1} (\mathbf{x}' - \mathbf{m}_j). \end{aligned} \tag{6.13}$$

Based on Table 6.3, FRB-KNN outperforms SVM when tested on four datasets (Bupa Liver +3.96%, Pima Indians diabetes +9.89%, Wisconsin breast cancer +1.00% and Ford automotive +8.43%) and it is only inferior to SVM when evaluated on Glass dataset (-18.22%). This probably indicates that FRB-KNN is less capable to handle uneven class problem. Nevertheless, the results still indicate that fuzzy rule-based K-NN is superior compared to other K-NN classifiers in face of uneven class issue. When compared to Bayesian classifier, the proposed method has the absolute advantage (Bupa Liver +9.56%, Glass +16.51%, Pima Indians

Table 6.3: Average Testing Accuracies (in %) on Different Datasets with Six Competing Classifiers.

Method \ Dataset	Liver	Glass	Diabetes	Breast	Ford
Crisp KNN	60.13	56.67	74.17	95.38	75.03
Fuzzy KNN	60.73	60.79	74.24	95.37	74.72
Fuzzy KNN*	65.42	67.64	75.63	95.90	74.72
FRB-KNN	65.86	69.17	76.29	96.06	75.11
Bayesian	56.29	52.66	74.22	89.66	73.94
SVM	61.90	87.39	66.41	95.06	66.68

'*' denotes fuzzy K-NN with weighted Euclidean distance measure

diabetes +2.07%, Wisconsin breast cancer +6.41% and Ford automotive +1.17%). This shows that the performance of Bayesian classifier is unreliable when the data is very limited or corrupted with noise.

It would be interesting to measure how well a method performs with respect to the best available method. [95] defines the robustness of a method as the ratio of the error rate to the smallest error rate over all other methods. To be consistent with the presented results, for mth algorithm, the classification accuracy, instead of error rate, is used to compute the robustness, $\mathbb{R}_m$

$$\mathbb{R}_m = \frac{\eta_m}{max_{1 \leq i \leq 6} \ \eta_i} \quad \text{where } 0 \leq \mathbb{R}_m \leq 1. \tag{6.14}$$

The most robust algorithm $m*$ for a particular problem has $\mathbb{R}_{m*} = 1$ while all other inferior algorithms $m \neq m*$ have $\mathbb{R}_m < 1$. The average of $\mathbb{R}_m$ across all problems provides a measure that corresponds to the robustness index. Table 6.4 shows the robustness index computed with (6.14) while Fig. 6.6 demonstrates the boxplot of the distribution of robustness ratio for each algorithm over the five datasets. It becomes clear that FRB-KNN is the most robust algorithm across all datasets. In 4 out of 5 problems, its accuracy is the highest with an average

Table 6.4: Robustness Index for Six Different Classifiers.

Method \ Dataset	Liver	Glass	Diabetes	Breast	Ford
Crisp K-NN	0.9130	0.6485	0.9721	0.9929	0.9989
Fuzzy K-NN	0.9222	0.6957	0.9731	0.9928	0.9948
Fuzzy K-NN*	0.9933	0.7740	0.9913	0.9984	0.9949
FRB-KNN	1.0000	0.7915	1.0000	1.0000	1.0000
Bayesian	0.8548	0.6025	0.9728	0.9333	0.9844
SVM	0.9399	1.0000	0.8704	0.9896	0.8878

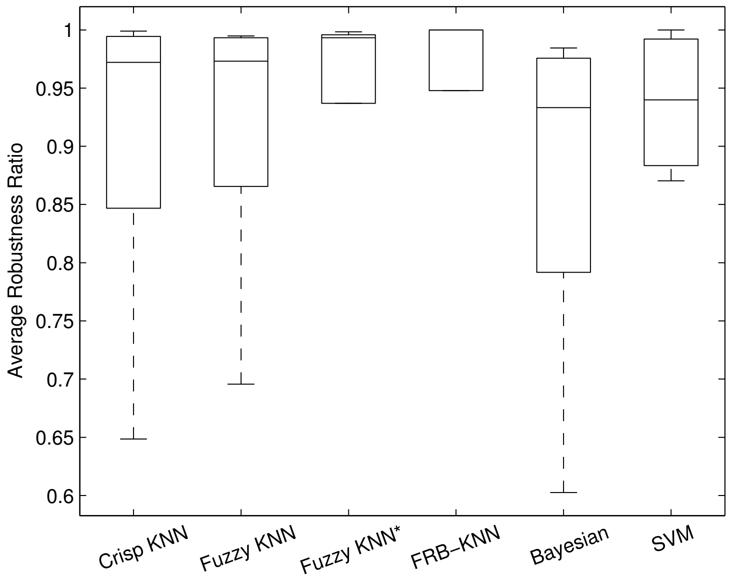


Figure 6.6: Performance distribution for each algorithm is computed by averaging the robustness ratio over the 5 datasets. The box represents the lower and upper quartiles of the distribution separated by the median while the outer vertical lines show the entire range of the distribution.

robustness ratio, $\bar{\mathbb{R}} = 0.9583$. This is followed by Fuzzy K-NN*, SVM, Fuzzy K-NN, Crisp K-NN and Bayesian with $\bar{\mathbb{R}} = 0.9504, 0.9375, 0.9157, 0.9051, 0.8696$ respectively.

6.6 Conclusion

In this chapter, a fuzzy rule-based K-NN algorithm that is an extension of fuzzy K-NN algorithm is presented. Unlike the conventional fuzzy K-NN which only uses simple notion of fuzzy membership value, the proposed algorithm incorporates a fuzzy rule-based initialisation procedure. The fuzzy sets and inference engine reinforce the management of uncertainties propagate throughout the system. To increase the quantitative properties of the fuzzy rule-base, a binary-coded GA that simultaneously optimises the parameters of the antecedent membership functions, rule consequents and the feature weights, is used. As a result, the decision boundary produced by fuzzy rule-based K-NN is very flexible and thus it can handle situation where the training data is insufficient or corrupted by noise. In contrast, crisp K-NN and fuzzy K-NN may not be able to overcome the situation as there is no way to change the decision boundary once the parameter K has been fixed. In case where transparency issue is of concern, the proposed fuzzy rule-base during initialisation is interpretable as only simple if-then rules are involved as shown in Section 6.3. The proposed algorithm was successfully applied to five real-world problems: Bupa liver disorders, Glass, Pima Indians diabetes, Wisconsin breast cancer, and Ford automotive. From the experimental results, fuzzy rule-based K-NN shows promising improvement over the competing crisp and fuzzy K-NN algorithms. The proposed method is also compared to SVM and Bayesian classifiers under the same experiment setup. The proposed fuzzy rule-based K-NN delivers better and more consistent results.

The proposed algorithm is able to overcome the shortcoming of the conventional crisp KNN and fuzzy K-NN when the training data is insufficient. Therefore it is suitable for problems where the data collection process is expensive. One limitation of the method is the slow computational speed which is also faced by the K-NN and fuzzy K-NN algorithms. However, this can be solved by incorporating indexing step to speed-up the searching process.

Chapter 7

Practical Application of Fuzzy Rule-Based Classifier for Inverter-Fed Induction Motor Fault Diagnosis

7.1 Introduction

Induction motor is the most common electric motors in the world. They are found in washing machines, refrigerators and fans, as well as conveyors, pumps, winders, wind tunnels and other industrial equipment. The relatively low manufacturing cost and rugged construction together with its simplicity in implementation has gained the induction motor popularity among the various types of motor. The reason that induction motor has these characteristics is because the rotor is a self-contained unit, with no external connections.

Due to thermal, electrical and mechanical stresses, failures on induction motors are unavoidable. These failures can result in a total loss of the machine, in addition to a costly downtime of the whole plant [96]. Thus, early detection of incipient motor faults is of paramount importance. Bearing and rotor related faults account for 40% and 10% of total induction motor failures respectively [97]. In particular, the main factors behind bearing faults are corrosion and dust [98]. There is a significant amount of research effort on the preventive maintenance of motors. This includes the conventional vibration [99, 100, 101, 102] and thermal analysis [103] and the state of the art motor current signature analysis (MCSA) [96, 104, 105, 106, 107]. MCSA has advantages such as ease of data collection because the current sensor is simple and can be installed nonintrusively. It has been suggested that stator current monitoring can detect the same fault indication as vibration monitoring without requiring access to the motor [105]. MCSA techniques can be divided into nonparametric, parametric, and high-resolution spectrum analysis methods. Nonparametric-based approaches rely on Fourier transform and periodicity detection in the frequency domain. These approaches include fast Fourier transform (FFT), power spectral density (PSD) and periodogram. In addition, Önel and Benbouzid [98] used Park and Concordia transform to detect bearing faults. The faults are indicated as the deformation of the ellipse trajectory plot for Park transform and the deformation of the circle trajectory plot for Concordia transform. In parametric-based approaches, autoregressive (AR) models are fitted to the time series of signal. Yule AR method has been proven to provide a stable model and its autocorrelation matrix is guaranteed to be nonsingular. The model parameters are then used to compute the frequency spectrum [96]. On the other hand, high-resolution spectrum methods refer to eigenvalue analysis of the autocorrelation matrix of the current signal. Two well-known eigenanalysis-based frequency estimators are multiple signal classification (MUSIC) and ROOT-MUSIC[108]. Both methods were proposed to solve resolution problem. In many cases, intelligent techniques are incorporated into the analysis to automate the process of diagnosis. Two popular techniques are neural network [109, 110, 111] and fuzzy logic system [112, 113]. Neural Network has the ability to implicitly detect complex nonlinear relationships from the data without formal statistical training. On the other hand, fuzzy logic system is more tolerant of imprecise input measurement.

Although MCSA has been successfully applied in practice, it is comparatively more computationally intensive and requires longer data train than time domain fault detection and diagnosis techniques. A classical time domain analysis focuses principally on the statistical characteristics of the signal such as peak level, standard deviation, skewness, kurtosis and crest factor. Zeraoulia et al. [114] proposed a fuzzy logic approach to diagnose stator related faults such as voltage unbalance and open phase by detecting the abnormalities in the current amplitude. Unlike stator condition monitoring where the current amplitude is useful to detect the abnormalities of the stator winding [115], there is very limited research on rotor/bearing related fault detection using time domain analysis. This is because current waveforms from motors with broken rotor bar or defective bearing can be very similar to the one from healthy motors. However, with proper feature selection and noise filtering techniques, time domain analysis is potentially valuable in providing extra information to confirm the presence of abnormalities. As the success of time domain analysis depends heavily on the availability of distinguishing features, fault diagnosis algorithms based on inputs obtained from transformation algorithms such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Independent Component Analysis (ICA) have been developed. Park et al. [116] used PCA and LDA to perform fault diagnosis for induction motor. A preliminary study using fixed supply-fed induction motor fault detection shows that time domain approach could be promising [117]. In particular, time-domain stator current signals projected onto two dimensional (2-D) feature space using ICA can yield distinctively separated clusters for three classes: healthy, bearing, and broken rotor bar, as shown in Fig. 7.2.

Thus far, research has focused on induction motors running at fixed speed. Normally, inverter-fed induction motors used in industries will operate at several predetermined speeds. For example, milling machines usually have speed settings at *low*, *medium*, *high*. In adjustable speed applications, AC induction motors are powered by inverters. The inverter converts DC power to AC power at the required amplitude and frequency. Unlike a fixed supply-fed motor line current, the inverter-fed motor line current includes electromagnetic interference (EMI) noise that adversely affects the fault diagnosis algorithm. The inherent floor noise reduces the possibility of detecting the true fault pattern. Moreover, the current signal is always corrupted by noise originated from the high frequency switching components inside the inverter. Although MCSA is one of the most powerful methods for diagnosing motor faults, it is reported that the noise can causes uncertainty when separating the healthy bearing pattern from the faulty pattern [118]. Jung et al. [107] notice that the abnormal harmonics of bearing faults are in the higher frequency bands than those of rotor faults. These bearing frequency bands may be close to or coincide with noise bands. When the noise amplitude is comparable with the bearing fault signature amplitude, it is then very difficult to detect incipient bearing defects. Therefore, there is a need to have another measure to assist or confirm the existence of bearing fault when carrying out frequency domain analysis. In addition, time domain analysis using ICA projection [117] reveals that the cluster locations are not invariant to the motor operating frequencies as illustrated in Fig. 7.10. As a result, it is almost impossible to define a fixed decision boundary for all operating frequencies. Thus, the challenge of extending existing time and frequency domain methods to variable speed drives remains unsolved.

To address the need for robust monitoring of induction motors in variable speed applications, this chapter introduces a novel hybrid time-frequency domain analysis method for detecting broken rotor bar and bearing faults in inverter-fed motors. The architecture of the proposed fault diagnosis system in Fig. 7.1 shows that motor condition monitoring is achieved by using a fuzzy logic system to fuse time and frequency domain features of the motor stator current. The main advantage of using fuzzy rule-based systems compared to other machine learning methods such as Neural Network is that fuzzy rules are interpretable; therefore the expert can verify the rules. The algorithm works as follows: first, ICA is used to project time domain current signals into 2-D feature space in order to extract the input distance information from the healthy cluster. Next, frequency domain analysis is performed by computing FFT spectrum and extracting the abnormal harmonics amplitudes. By using fuzzy logic system to combine the results from time-domain

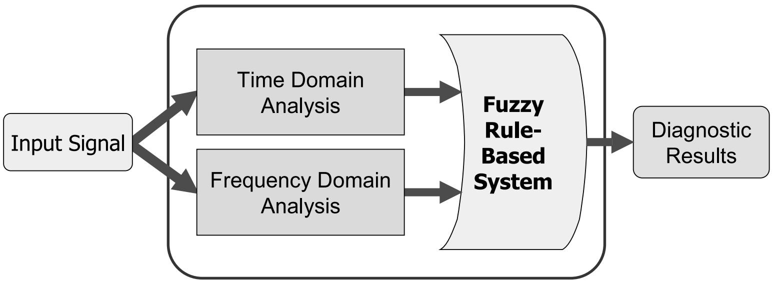


Figure 7.1: Overview of the proposed hybrid time-frequency domain analysis algorithm.

analysis and MCSA, robust fault diagnosis can be achieved. In addition, the computationally efficient time domain analysis can be used as an early warning system to detect the presence of faults. Lastly, the problem of stator current noise caused by high frequency switches in the inverter is handled by introducing a new noise reduction technique known as “Ensemble and Individual Noise Reduction (EINR)”. This technique was initially proposed in our recent work [119] and was proven to be able to reduce noises in current signals effectively.

In the following section, MCSA technique which uses FFT spectrum analysis to detect broken rotor bar and bearing related faults will be introduced. Section 7.3 gives a brief introduction to Independent Component Analysis and explains how this analysis can be used to extract features from time domain signals. Due to the random stator current noise from the inverter-fed motor, a noise reduction algorithm known as *Ensemble and Individual Noise Reduction* (EINR) is employed in Section 7.4. The proposed hybrid algorithm is introduced in Section 7.5 while the experimental results in Section 7.6 elucidate the effectiveness of the algorithm. Finally, conclusions are drawn in Section 7.7.

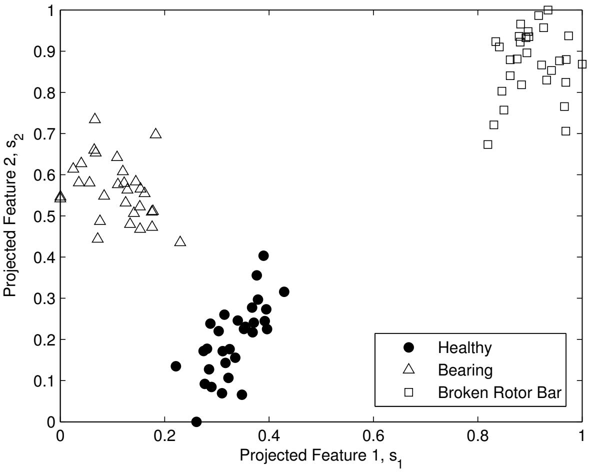


Figure 7.2: Scatter plot of the extracted 2-D features for fixed supply-fed motor (50Hz) using Independent Component Analysis (ICA) method.

7.2 Motor Current Spectral Analysis

Among all the techniques in Motor Current Signature Analysis (MCSA), the most classical and widely used technique is Fast Fourier Transform (FFT). FFT is an algorithm to compute Discrete Fourier Transform (DFT) in a more efficient way. FFT decomposes a sequence of values into components of different frequencies.

Motor current acts as an excellent transducer for detecting faults in motors. A noiseless healthy motor current signal is a perfect sinusoidal wave. As there is no harmonics, we can only see one peak in the frequency spectrum. During actual operation, many harmonics will present in the motor signal. Certain harmonics come from the supply and they are of little consequence. Thus, FFT spectrum will show many peaks including line frequency and its harmonics. This is known as the motor's current signature. Harmonics also generated from various electrical and mechanical faults. MCSA operates on the principle that faults causing a change in the internal flux distribution, thus generating the harmonics.

7.2.1 Broken Rotor Bar Fault

In the current signature, the motor pole passing frequency will show up as a sideband to the line frequency. Broken rotor bar can cause anomaly in the magnetic field[105]. As the rotor bars start degrading (i.e., high resistance joints are present or a crack starts developing), the rotor impedance rises. Due to this, the current drawn at the following sideband frequency rises

$$f_{brb} = f_s(1 \pm 2s) \tag{7.1}$$

where

f_s : supply frequency;

s : per-unit slip;

A typical current spectrum of an induction motor with broken rotor bars is given in Fig. 7.3. The presence of broken rotor bars is indicated by the difference in amplitude between the fundamental and the left sideband. The amplitude of the left sideband frequency component is proportional to the amount of broken rotor bars. As a general guide, it has been reported that a difference less than 50dB is a sign of broken rotor bar faults[105, 108, 120, 121]. (7.1) shows that the rotor bar frequencies are a function of the machine slip. As the load is reduced, the separation between the fundamental and sideband will become smaller and detection becomes difficult because the left sideband frequency component will be masked by the fundamental frequency component. Therefore, this method will work well only if the motor is loaded.

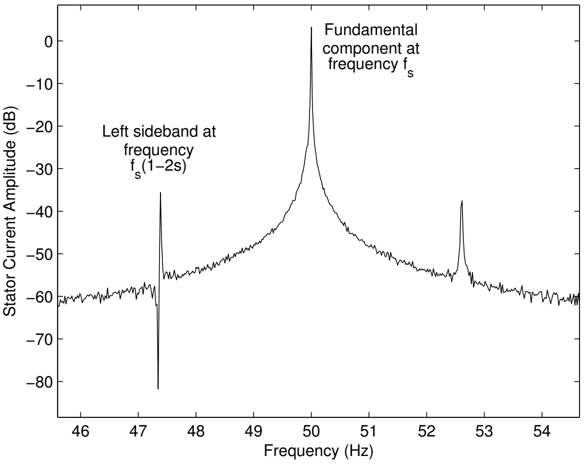


Figure 7.3: Current spectrum of an induction motor with broken rotor bars.

7.2.2 Bearing Fault

Bearing faults can be divided into inner race, outer race, ball defect, or cage defect, which are the main sources of machine vibration. Inner or outer race are the two more common bearing faults. The mechanical vibrations in the air-gap can be considered as slight rotor radial displacements which result in air-gap eccentricity[118]. The relationship of bearing vibration to stator current spectrum results from the fact that any air-gap eccentricity produces anomalies in the air gap flux density[104]. The bearing related faults can be detected by determining the stator current spectral frequencies induced by characteristic vibration frequencies f_v. The induced current frequencies for bearing fault, f_{bng} given by

$$f_{bng} = |f_s \pm k f_v| \tag{7.2}$$

where $k = 1, 2, 3, \ldots$ are the harmonic indexes and f_v can be either inner race defect frequency, f_i or outer race defect frequency,f_o

$$f_{i,o} = \frac{n}{2} f_r \left[1 \pm \frac{bd}{pd} \cos\phi \right] \tag{7.3}$$

where

n : number of bearing balls;

f_r : mechanical rotor speed in Hz;

bd : ball diameter;

pd : bearing pitch diameter;

ϕ : contact angle of the balls on the races.

Fig. 7.4 compares the bearing frequency components for healthy motor and motor with inner race bearing fault. It is noticed that the amplitudes information can be uncertain due to the noise level or baseline drifting as mentioned in Section 7.1. Unlike broken rotor bar fault whereby only left sideband needs to be monitored, notice that (7.2) have unspecified bearing fault harmonic numbers (k=1,2,3,...). Dominant harmonic (specific k) must be identified in order to detect the faults. Jung et al. [107] proposed a frequency auto search algorithm to search for the most dominant harmonic. The algorithm first creates an N-dimensional candidate searching index vector which stores different harmonic frequencies. N denotes the number of unspecified harmonic numbers. Next, the algorithm normalizes the magnitude of the incoming FFT signal to the healthy signal according to the frequency index. The most dominant harmonic is the one with the highest ratio. Due to the noise issue, the bearing fault component may spread over a wider frequency band compared to the broken rotor bar component (see Fig. 7.4). Therefore, the amplitude of the bearing fault component is averaged across the frequency band of range ±0.5Hz.

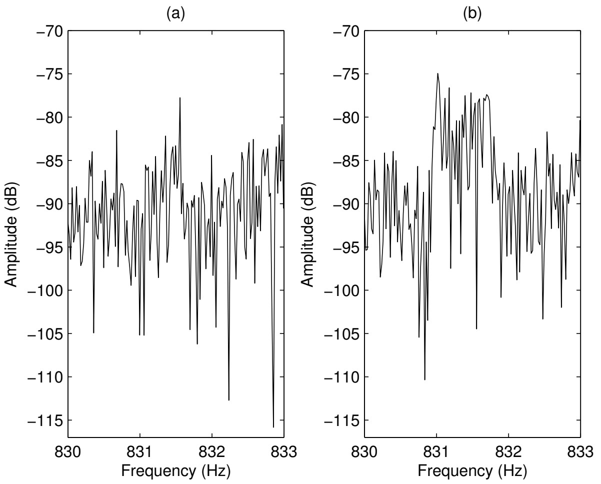


Figure 7.4: Uncertain bearing frequencies components between (a) healthy motor (b) motor with inner race bearing fault. Due to the noise, the amplitude difference between two classes are less obvious.

7.3 Independent Component Analysis

Independent Component Analysis (ICA) is a statistical technique for decomposing a complex dataset into independent sub-parts. In other words, ICA is a technique that represents a multidimensional random vector as a linear combination of nongaussian random variables ('independent components') that are as independent as possible. ICA is a nongaussian version of factor analysis, and is somewhat similar to principal component analysis. ICA has many applications in data analysis, source separation, and feature extraction[122].

ICA problem can be formulated as

$$\mathbf{x} = \mathbf{A} \bullet \mathbf{s} \tag{7.4}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_m)^T$ is an observed m-dimensional random vector , $\mathbf{s} = (s_1, s_2, \ldots, s_n)^T$ is an n-dimensional (latent) random vector whose components are assumed mutually independent, and $\mathbf{A}$ is an unknown $m \times n$ mixing matrix to be estimated. The basic assumptions of this algorithm are that the components are statistically independent and the independent components must have a non-gaussian distributions. As a matter of fact, ICA is an extension to Principal Component Analysis (PCA). The key difference between ICA and PCA is that PCA obtains principal components which are uncorrelated whereas ICA obtains components that are statistically independent. Independence is a stronger condition than uncorrelatedness. In (7.4), the (pseudo)inverse of the mixing matrix $\mathbf{A}$ is known as transformation matrix $\mathbf{W}$. The most widely used algorithm for estimating $\mathbf{W}$ is FastICA algorithm[123]. FastICA learning rule finds a transformation matrix $\mathbf{W}$ that minimises the mutual information of the transformed component, $\mathbf{s}$. This is roughly equivalent to finding directions in which the independency is maximised. Denote $\mathbf{W} = [\mathbf{w}_1\mathbf{w}_2\ldots\mathbf{w}_m]^T$, one-unit fastICA algorithm has the following form

$$\mathbf{w}(k) = E\{\mathbf{x}g(\mathbf{w}(k-1)^T\mathbf{x})\} - E\{g'(\mathbf{w}(k-1)^T\mathbf{x})\}\mathbf{w}(k-1) \tag{7.5}$$

where k is the iteration index, the weight vector $\mathbf{w}$ is also normalized to unit norm after every iteration, and the function g is the derivative of the function G (any quadratic function) used in the general objective function. One-unit fastICA is then extended to multiple-unit to estimate the full $\mathbf{W}$[123]. By rearranging (7.4), the transformed variables $\mathbf{s}$ are given as

$$\mathbf{s} = \mathbf{W} \bullet \mathbf{x}. \tag{7.6}$$

In pattern classification problem, the transformation can be considered as feature extraction whereby s_i is the coefficient of the i^{th} feature in the observed data vector $\mathbf{x}$. To overcome the curse-of-dimensionality, it is always feasible to have low-dimensional data (i.e., $i \leq 3$). For example, if two-dimensional input $\mathbf{s} = (s_1, s_2)^T$

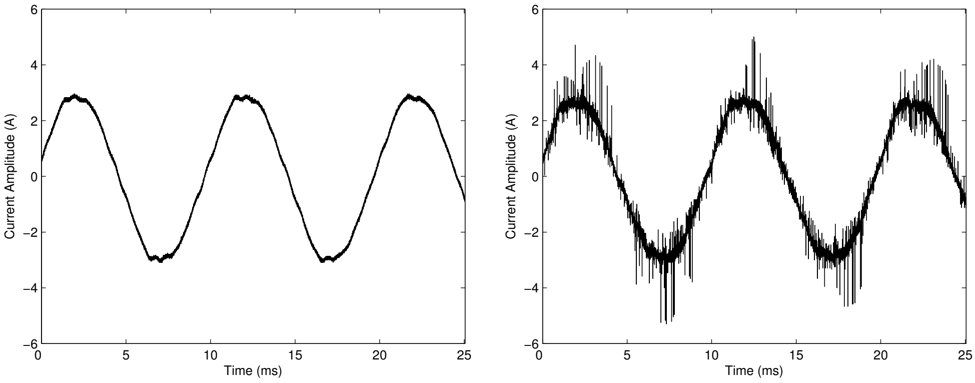
is needed, then transformation matrix $\mathbf{W}$ dimension is chosen to be $2 \times m$.

7.4 Ensemble and Individual Noise Reduction

The preliminary work on using Independent Component Analysis (ICA) to extract time domain features is very successful on fixed supply-fed induction motor as mentioned in Section 7.1. Therefore, there is a strong interest to extend this technique to inverter-fed induction motor. The experimental data in Fig. 7.5 shows that the stator current from inverter-fed motor has relatively higher noise. As a result, the clusters corresponding to different motor conditions overlap with each other in the ICA feature space as shown in Fig. 7.6. In particular, the healthy cluster and bearing cluster have high degree of overlap. In view of this, a noise reduction method known as *Ensemble and Individual Noise Reduction* (EINR) is employed to overcome the noise problem in inverter-fed motor stator current. Fig. 7.7 shows the procedures involved in EINR. Since noise is a random event, it is possible to reduce the noise by averaging a predetermined number of corrupted signals. Denote $(x(t), x(t+T), \ldots, x(t+(N-1)T))$ as a set of aligned periodic noisy base signals from the same source where N is the number of signals and T is the length of the signal, the first step is to compute the profile signal, $P(t)$ which is the summation of all the noisy base signals

$$P(t) = \sum_{k=0}^{N-1} x(t+kT). \tag{7.7}$$

It is clear from Fig. 7.7 that most of the noises have been eliminated in the profile signal. In the next step, for every incoming noisy signal $y(t)$, the denoised signal



(a)

(b)

Figure 7.5: 50Hz Stator current signal from the (a) fixed supply-fed induction motor (b) inverter-fed induction motor.

$\hat{y}(t)$ can be computed with the following equation

$$\hat{y}(t) = \frac{P(t) + y(t)}{(N+1)}. \tag{7.8}$$

In this way, the denoised signal shares the common profile yet retains its own information. The merit of this unique noise reduction method is that it can minimise the intra-class variation. In other words, as long as there are differences in the "profiles" from different classes, then the data will form well separated clusters in the feature space. EINR can be easily extended to any signal that has specific waveform (in this study, the stator current has sinusoidal waveform) yet corrupted by noise, prior to the feature extraction step such as PCA, LDA or ICA. Fig. 7.8 depicts the effect of applying EINR technique with parameters $N = 15$ and $T = 25000$. The overlapping problem has been solved. Interestingly, the clusters are now even more compact than those from fixed supply-fed motor (refer to Fig. 7.2).

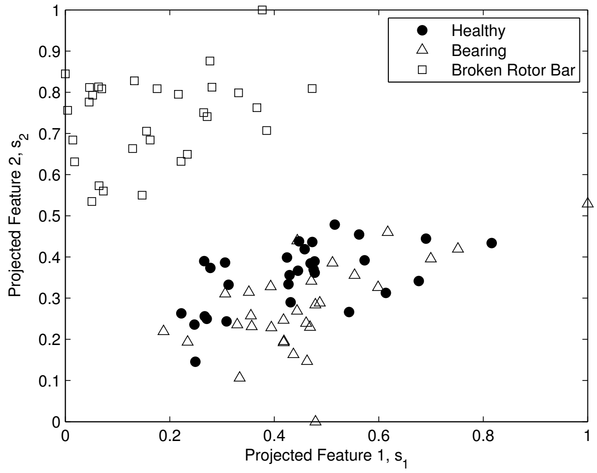


Figure 7.6: Scatter plot of the extracted 2-D features for inverter-fed motor (50Hz) using Independent Component Analysis (ICA) method.

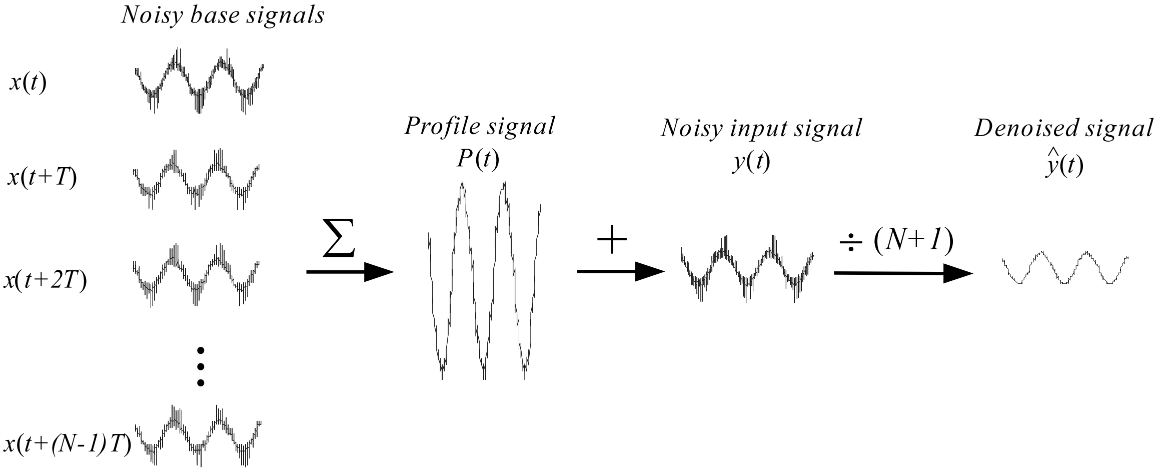


Figure 7.7: Ensemble and individual noise reduction procedures.

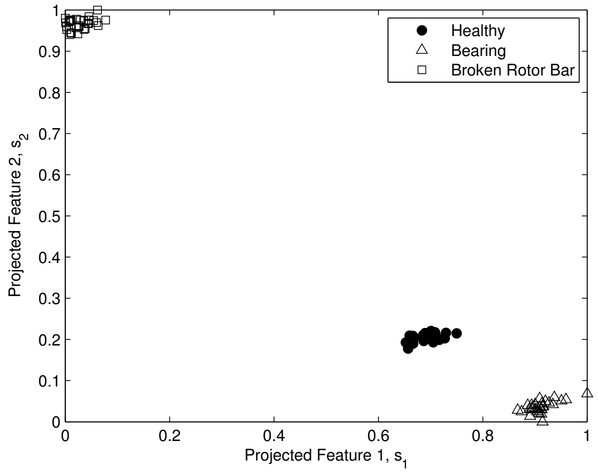


Figure 7.8: Scatter plot of the ICA extracted 2-D features for inverter-fed motor (50Hz) after applying *Emsemble and Individual Noise Reduction* technique.

7.5 Proposed Algorithm

The literature survey in Section 7.1 indicates that the MCSA approach to detect bearing related faults can be difficult when the noise amplitude is comparable to the bearing fault frequency component. On the other hand, the time domain projection using ICA algorithm has been shown to give promising results by producing distinctly separated clusters in 2-D feature space for fixed supply-fed induction motor. However, as shown in Fig. 7.10, the clusters are not invariant to speed in inverter-driven motors (i.e., the locations of the clusters vary with respect to different operating frequencies). Therefore, the decision boundaries are not fixed for all speeds. For these reasons, a hybrid time-frequency domain analysis that aims at increasing the robustness of the frequency domain analysis by utilizing the cluster information obtained from time-domain feature projection is proposed. This hybrid approach enables the advantages of time and frequency domain analysis to complement each other for the needs of different application. Since time domain analysis requires much shorter segment of data and is computationally simpler than FFT, it provides the possibility of detecting the presence of a fault quickly and efficiently. This property is vital in applications where power is limited. An example being induction motors that are installed in a remote area and the communication link is provided by solar powered transmitters. Power can be saved by first transmitting a short data train to enable ICA projection to be performed. Any abnormal condition will be presented as an outlier to the healthy cluster in the feature space. When an abnormality is suspected, a further command can be sent to the transmitter to request for a longer signal for more detailed diagnosis.Thus, time domain analysis serves as the first cut fault detector. When both time and frequency domain data are fused together by the fuzzy rule-based classifier, a robust diagnosis result can be obtained irregardless of the motor operating speed. Details of the hybrid algorithm are further depicted in Fig. 7.9. The following subsections will discuss each block in greater detail.

7.5.1 Data Requirement and Processing

Before the raw stator current signal can be used, the signal must be preprocessed to remove the unwanted noise using the EINR approach described in Section 7.4. In order to compute FFT analysis, the length of the signal must be long enough (typically $>$ 20s depending on the sampling rate) to provide sufficient resolution to detect frequency components corresponding to the various fault conditions. While the lengthy signal is required for FFT analysis, the number of data points is enormous for ICA feature extraction in time domain. Therefore, for time domain analysis only one cycle of the signal will be used. This can reduce the computation time tremendously so as to provide faster response.

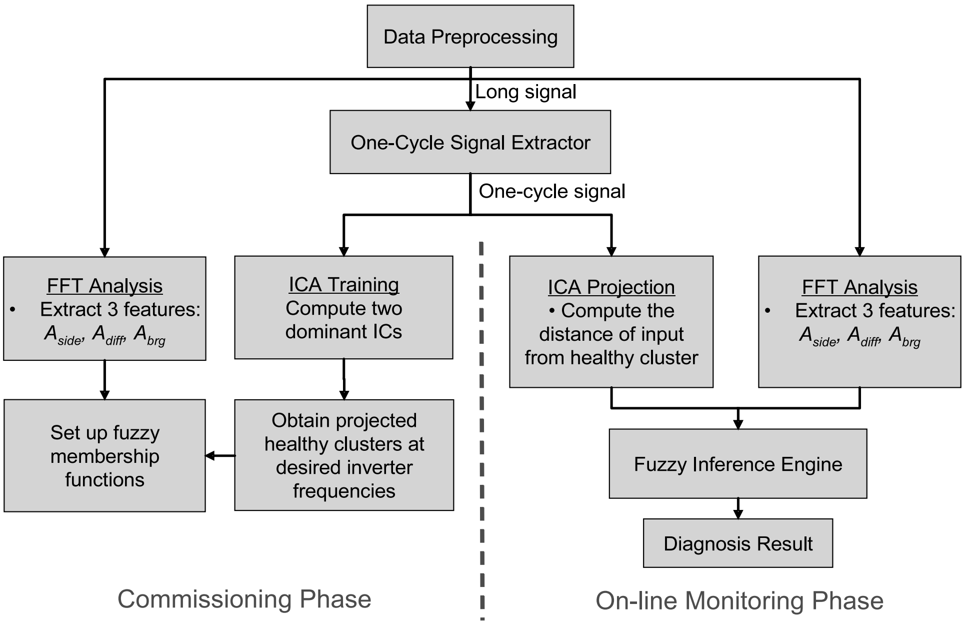

Data Preprocessing
Long signal
One-Cycle Signal Extractor
One-cycle signal
FFT Analysis
• Extract 3 features: A_{side}, A_{diff}, A_{brg}
ICA Training
Compute two dominant ICs
ICA Projection
• Compute the distance of input from healthy cluster
FFT Analysis
• Extract 3 features: A_{side}, A_{diff}, A_{brg}
Set up fuzzy membership functions
Obtain projected healthy clusters at desired inverter frequencies
Fuzzy Inference Engine
Diagnosis Result
Commissioning Phase
On-line Monitoring Phase

Figure 7.9: Details of the proposed hybrid time-frequency domain analysis algorithm.

7.5.2 Commissioning Phase

The commissioning phase involves both time and frequency domain analysis as well as finding the parameters for the fuzzy rule-based classifier.

7.5.2.1 Time Domain

For time domain analysis, a single cycle of current signal (one cycle signal in Fig. 7.9) is discretised into the predetermined number of data points, T. For unbiased training, equal number of data sets are collected from healthy and faulty motor. From ICA analysis, the two most dominant independent components $\mathbf{w}_1$ and $\mathbf{w}_2$ will be recorded. The resultant transformation matrix $\mathbf{W}$ is a $2 \times m$ matrix, $\mathbf{W} = [\mathbf{w}_1 \mathbf{w}_2]^T$, where m is the length of the signal. The features extracted from current signals using (7.6) is shown in Fig. 7.2. ICA only needs to be computed once as the ICs stored in $\mathbf{W}$ can be used repeatedly to extract other signals with different frequencies. Fig. 7.10 shows that the healthy clusters are very compact,

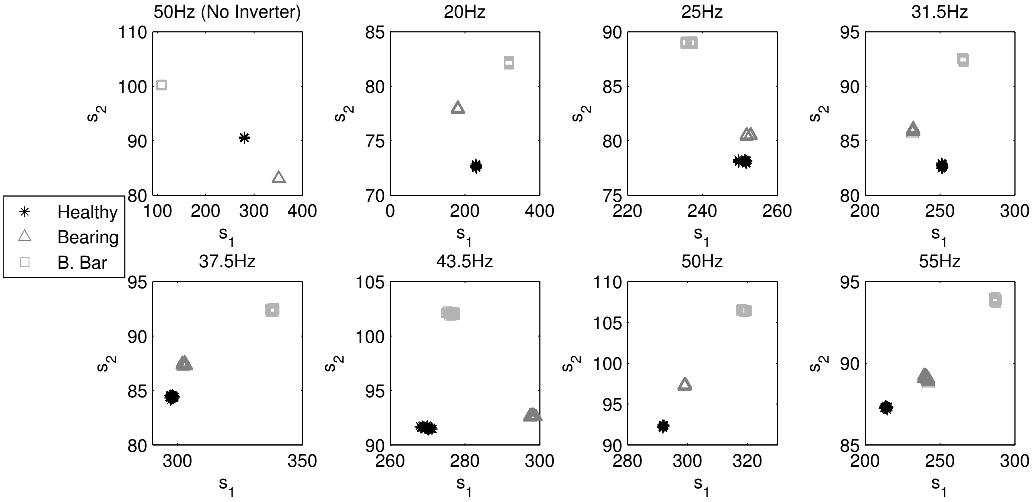


Figure 7.10: Healthy and faulty clusters (bearing and broken rotor bar) for variable inverter frequencies during training stage except the left top one for fixed supply frequency. Each cluster contains 30 training data points.

hence by measuring the distance of the incoming data from the healthy cluster during the testing phase, one can decide if the signal is faulty or not. In particular, a faulty signal will be clearly shown as an outlier. Thus in time domain analysis, the Euclidean distance information (d) extracted using the following equation is used as one of the inputs to the fuzzy rule-based classifier to provide quick diagnosis

$$d = \sqrt{(s_1 - \bar{s_1})^2 + (s_2 - \bar{s_2})^2} \tag{7.9}$$

where $(\bar{s_1}, \bar{s_2})$ is the centroid of the healthy cluster.

7.5.2.2 Frequency Domain

The frequency domain features are extracted from the FFT spectrum generated from the raw signal (long signal in Fig. 7.9). The components at the broken rotor bar frequency, f_{brb} and the bearing fault frequency, f_{brg} will be examined. There are three features to be extracted. The first feature is the amplitude of the left sideband, A_{side} while the second feature is the amplitude difference of the fundamental component and left sideband, A_{diff}. Both features are used to detect the existence of broken rotor bar. The third feature is the amplitude of the bearing fault component, A_{brg}. Since the signatures of the various types of bearing faults occur at different frequencies, A_{brg} is selected as the most dominant frequency component as extracted by the frequency auto search algorithm [107].

7.5.2.3 Fuzzy Rule Base

A fuzzy rule-based system allows experts' knowledge to be incorporated through intuitive *if-then* rules. The numerical data are represented as linguistic information which are in fact a set of fuzzy membership functions. The following rules are used to classify healthy motor and motor with bearing fault or broken rotor bar fault

R_1**:** If A_{side} is *small* and A_{diff} is *large* and A_{brg} is *small* and d is near, then STATUS is *Healthy*

R_2**:** If A_{side} is *small* and A_{diff} is *large* and A_{brg} is *large* and d is far, then STATUS is *Bearing Fault*

R_3**:** If A_{side} is *large* and A_{diff} is *small* and A_{brg} is *small* and d is far, then STATUS is *Broken Rotor Bar.*

Product t-norm and Max t-conorm are used to perform the fuzzy inferencing in this study. The diagnosis result is determined by the rule with the highest firing level, based on winner-takes-all strategy. Each of the antecedent set is a sigmoidal membership function and is characterised by two parameters a and c as shown in (7.10). Each universe of discourse has two fuzzy partitions.

$$f_{sig}(x) = \frac{1}{1 + e^{-a(x-c)}}. \tag{7.10}$$

For distance membership function, the parameter c in (7.10) is made adaptive with respect to the motor speed and is determined by

$$c = \tau \times d_{\bar{H}-\bar{F}} \tag{7.11}$$

where $\bar{H}$ is the centroid of the healthy cluster, $\bar{F}$ is the centroid of the nearest faulty cluster and $0 < \tau < 1$ is a threshold that controls the radius of the healthy cluster boundary. If $\tau \approx 0$, the incoming data must be very close to the healthy cluster centroid to be considered as a healthy signal. In contrast, for $\tau \approx 1$ the incoming data is still considered as a healthy data even its distance is nearer to faulty cluster than the healthy cluster. In other words, the parameter τ determines the separation needed for a data point to be considered as an outlier (i.e., faulty data). In a more critical application, τ can be set to a smaller value ($\tau \leq 0.5$) to avoid the false classification of a faulty motor as a healthy motor. The rest of the fuzzy membership function parameters are determined manually based on the observations during feature extraction steps (see Fig. 7.11(b)-(d)).

7.5.3 On-line Monitoring Phase

During the on-line monitoring phase, pre-processing of the motot stator current into both time and frequency domain data are conducted in a manner similar to the commissioning phase. The single cycle ("short") test stator current input vector is projected onto the 2-D feature space using the pre-trained ICA transformation matrix **W**. As mentioned earlier, the distance of projected test input from the healthy cluster provides a measure of how similar the signal is to the healthy signal. If the distance is small, then the signal likely belongs to the healthy class.

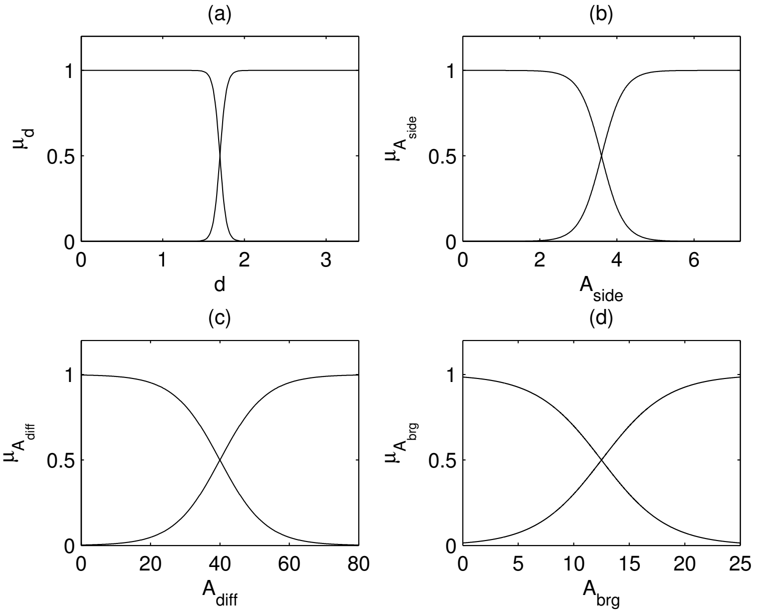


Figure 7.11: Fuzzy membership functions for four inputs: (a) distance, d (b) amplitude of the left sideband, A_{side}, (c) amplitude difference of the fundamental component and left sideband, A_{diff} (d) amplitude of the bearing fault component, A_{brg}. Note that the distance membership function is adaptive with respect to the operating speed, (a) only shows one of the instances.

Otherwise, it probably belongs to the faulty class. For frequency domain analysis, three amplitude features are extracted from the FFT spectrum of each incoming "long" test data sequence. The extracted distance and amplitude information are then fuzzified as fuzzy singletons and fed into the fuzzy rule-based classifier with the membership functions designed during the commissioning phase. Finally, the fuzzy inference engine gives the diagnosis result based on winner-takes-all principle.

7.6 Experimental Results and Discussion

The experimental setup in the laboratory used to assess the hybrid time-frequency domain method for monitoring the condition of inverter-fed induction motor is

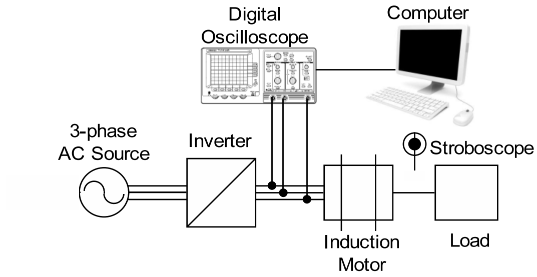


Figure 7.12: Experiment setup.

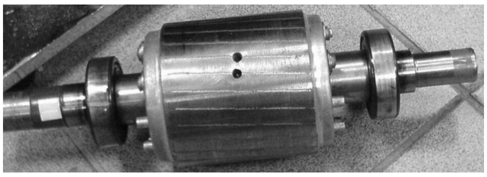


Figure 7.13: Two holes are drilled on the rotor bar to simulate broken rotor bar.

illustrated Fig. 7.12. The proposed algorithm is tested on three identical motors. The technical specifications of the motor and its bearings are given in Tables 7.1 and Table 7.4 respectively. Two motors are artificially modified in which one of them has dent on the inner race while the other has two holes drilled on the rotor bar as shown in Fig. 7.13. All induction motors are driven by a voltage-fed pulse-width modulated inverter at variable frequencies. Segments of raw stator current is collected using a digital oscilloscope at the sampling rate of 20 kHz for 50 seconds. A stroboscope is used to measure the mechanical rotor speed so as to compute the slip. The motor is loaded by a DC generator with variable resistances.

Since the motor is connected to an inverter, its speed is variable. In this study,

Table 7.1: Rated Parameters of the Induction Motor Under Study

Power	1.1kW
Voltage (△/Y)	230/400V
Current (△/Y)	4.5/2.6A
Frequency	50Hz
Speed	1410rpm
Pole pairs	2

Table 7.2: Measured Rotor Speeds and Computed Broken Rotor Bar Frequencies

Inverter Frequency (Hz)	Rotor Speed (rpm)	Per-unit slip	f_{brb} (Hz)
20.0	582	0.0300	18.80
25.0	729	0.0280	23.60
31.5	920	0.0265	29.83
37.5	1094	0.0276	35.43
43.5	1271	0.0261	41.23
50.0	1461	0.0260	47.40
55.0	1600	0.0303	51.67

the inverter is set to seven speeds: 20Hz, 25Hz, 31.5Hz, 37.5Hz, 43.5Hz, 50Hz and 55Hz. At each inverter frequency setting, 30 signal segments are collected for each class and used for training phase. On the other hand, the classification accuracy of the proposed algorithm is tested by 150 samples (50 segments of stator current data for each class). The one-cycle stator current signal which is processed by ICA projection has 360 data points. A "long" signal segment, containing 1×10^6 data points, is used to generate the FFT spectrum in order to provide sufficient resolution to detect abnormal current signature. From the FFT spectrum, the frequency domain parameters (A_{side}, A_{diff} and A_{brg}) can be extracted. Firstly, the inverter frequencies, the measured motor speed, (7.1) and (7.2) are used to calculate the broken rotor bar frequencies and bearing fault frequencies. Table 7.2 shows the estimated broken rotor bar frequencies. The bearing fault frequencies are difficult to obtain as there are unspecified harmonic number in (7.2). By applying the frequency auto search algorithm [107], the most dominant harmonic frequency is found at $k = 2$ and $f_v = f_i$. The estimated bearing fault frequencies for the motors used in this study are tabulated in Table 7.3. The amplitude of the frequency components (A_{side}, A_{diff}, A_{brg}) are therefore extracted at these estimated frequencies.

Table 7.3: Measured Rotor Speeds and Computed Inner Race Bearing Fault Frequencies

Inverter Frequency (Hz)	Rotor Speed (rpm)	f_{brg} (Hz)
20.0	584	331.507
25.0	731	414.916
31.5	922	523.284
37.5	1098	623.170
43.5	1274	723.097
50.0	1464	830.890
55.0	1603	910.025

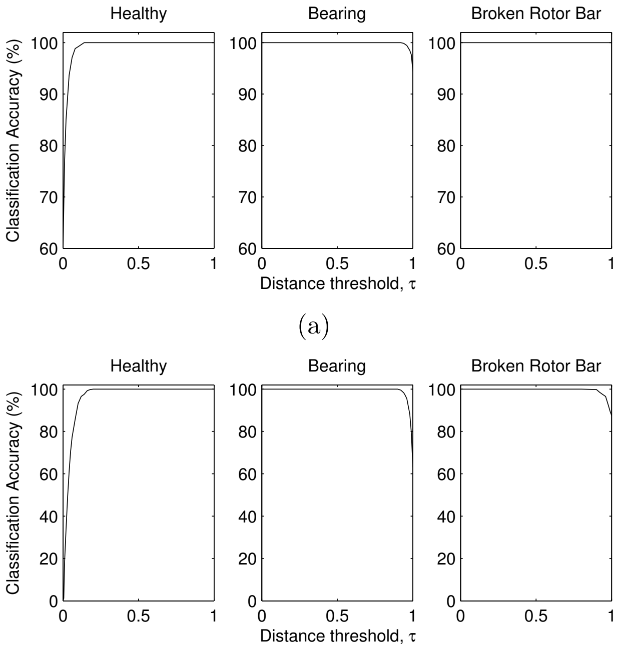
Table 7.4: Bearing Parameters

Ball diameter, bd	8.89mm
Bearing pitch diameter pd	38.5mm
Number of bearing balls, n	13
Contact angle, ϕ	0°

Table 7.5 shows the classification results of the proposed hybrid algorithm obtained by using 0.5 as the Euclidean distance threshold, τ in (7.11) that controls the radius of the healthy cluster boundary. All 150 test samples are classified correctly. The results demonstrate that the combination of time and frequency domain analysis can give excellent classification results. Since τ is a design parameter that determines the sensitivity of the proposed algorithm to the presence of faults, its effect on the performance of the hybrid algorithm is investigated. As illustrated by Fig. 7.14(a), the number of healthy motor samples that are misclassified as faulty increase as τ is reduced. The dependence of the classification accuracy of healthy motor on τ is apparent only when it is close to zero or one because the cluster corresponding to a healthy motor is very compact. While the use of a more stringent condition (small τ) to confirm a healthy signal increase the false negative rate (healthy signal being classified as faulty), it reduces false positive rate (faulty signal being classified as healthy) at the same time. Con-

versely, using a large τ to define the boundary of the healthy cluster increases the likelihood that motors with bearing fault are misclassified as healthy. The bearing fault data cluster is nearest to the one corresponding to a healthy motor (See Figure 7.10), so a large τ increases the probability that a bearing fault sample is not treated as an outlier. Coupled with the uncertainty in the amplitude data obtained from frequency domain analysis shown in Fig. 7.4, the misclassification rate increases. The results of the study also show that the classification accuracy of broken rotor bar fault is least sensitive to the selection of τ as its cluster is further away from the healthy cluster across different motor speeds. Even when the broken rotor bar fault signal falls within the healthy cluster radius, the fuzzy inference engine manages to utilize the distinct sideband information from frequency domain analysis to arrive at a correct classification. Thus, the proposed hybrid system allows the results from time and frequency domain analysis complement each other.

A feature of the proposed hybrid approach is that the data length (360 data points per signal) and computational requirement for ICA projection is less demanding than the long signal used for FFT analysis (1×10^6 data points per signal). Consequently, in applications where motor condition monitoring need to be performed using less sophisticated hardware, the Euclidean distance data from ICA projection of the time domain data can be used to provide a quick, but rough, determination of the motor condition. Fig. 7.14(b) shows the classification results with respect to the changes of the threshold, τ when the diagnosis is solely based on time-domain ICA feature. The classification accuracy trend is very similar to the one obtained from the hybrid system except that some broken rotor bar sig-



(a)

(b)

Figure 7.14: The effect of Euclidean distance threshold, τ towards the classification accuracies for (a) hybrid time-frequency domain analysis algorithm, (b) independent time domain analysis algorithm.

Table 7.5: Proposed Hybrid Algorithm Performance

Class	Diagnosis Accuracy (%)
Healthy	100
Bearing	100
Broken Rotor Bar	100

nals are misclassified when $\tau \approx 1$. Without the additional information from the frequency domain analysis, the classifier performance is slightly more sensitive to the changes of distance threshold when τ approaches zero or one.

7.7 Conclusion

This chapter presents a methodology for the diagnosis of broken rotor bar and bearing faults for inverter-fed induction motor by making use of features extracted from time and frequency domain analysis. The method is suitable for both offline and online applications. One advantage of the proposed hybrid algorithm over the conventional motor current signature analysis approach is that the classification accuracy is more robust against the uncertainties in the extracted features. Moreover, the algorithm can be made modular whereby the cluster information obtained from the time domain ICA analysis of the motor stator current carried out as a quick initial test to identify abnormal signal as an outlier before proceeding with the more computationally complex frequency domain analysis to identify the types of faults.

One limitation of the method is that frequency domain analysis requires the knowledge of motor parameters such as number of pole pairs, bearing diameter etc. To overcome this, it would be useful to apply ICA on the FFT spectral in future work.

Although results presented here are tested on motors with broken rotor bar and bearing related faults, the proposed hybrid method is extendable to more fault classes.

Chapter 8

Conclusion

Although the word "fuzzy" carries the connotation of uncertainty, research has shown that there may be limitations in the ability of conventional type-1 singleton fuzzy classifier to model and minimise the effect of uncertainties. There are two major restrictions. Firstly, the input is modeled as singleton regardless of the level of uncertainties that the input carries. Secondly, the membership function of a type-1 fuzzy set has no uncertainty associated with it. Once the type-1 membership function has been chosen, all the uncertainty disappears because type-1 membership function is totally precise [37]. In view of these limitations, this thesis has delved at length into various aspects of extensional fuzzy rule-based classifiers: classifier characteristics, assessment of performances, and classifier design methodologies. The results presented in this thesis may enrich the entire wealth of fuzzy logic system for pattern classification.

In Chapter 2, non-singleton type-1 fuzzy rule-based classifier (NSFRBC) was investigated. The analysis suggested that NSFRBC is capable of producing variable boundary which may be useful to overcome the fuzzy boundary between classes. In order to show the advantage of NSFRBC, a non-singleton fuzzy logic classifier evolved using genetic algorithm was assessed using a benchmark cardiac arrhythmias classification problem. It was found that fuzzy classifier consistently outperformed its singleton counterpart regardless of the types of input features used. The results has provided clear evidence that NSFRBC can better cope with input uncertainties. Moreover, the results revealed that NSFRBC has noise suppression capability. This is because NSFRBC can achieve good classification performance even when a shorter window length was used for feature extraction. A shorter window length implies that there is less averaging effect, thus the data is susceptible to noise. This finding is in agreement with the mathematical analysis shown in Section 2.1. An important contribution of this study is that NSFRBC may extend the possibility to use the features that are easier to extract but contain more uncertainties; thus it can reduce the cost or complexity in feature extraction.

As far as the uncertainty is concerned, non-singleton fuzzy rule-based classifier is only capable to handle uncertain inputs. The uncertainties associated with linguistic (antecedent and consequent) are not adequately accounted for. Therefore, there is a need for a more sophisticated fuzzy classifier. Chapter 3 aims to seek a better understanding of how the extra degrees of freedom provided by footprint of uncertainty (FOU) in type-2 fuzzy membership function can be utilised to capture these uncertainties. A four-step design strategy for interval type-2 fuzzy rule-based classifier was proposed. This method is intuitive in such a way that the FOUs can be conveniently generated from the data itself. By using the same ECG arrhythmias classification problem as in previous chapter, five case studies were carried out to study the performance of type-2 FRBCs compared against type-1 FRBC in face of different sources of uncertainties. Different sources of noises have been included to model the uncertainties associated with the vagueness in membership functions and the unpredictability of the data. The results showed that the proposed strategies to design the FOUs are essential to achieve a high performance type-2 fuzzy rule-based classifier when dealing with the uncertainties. The significance of this study is that it provides a feasible solution to design the interval type-2 fuzzy membership functions when one is uncertain about the descriptions for the features. It becomes clear that the uncertainties associated with the membership functions can be encapsulated by the footprint of uncertainty (FOU).

In Chapter 4, the robustness of interval type-2 fuzzy rule-based classifier was studied in three aspects. Firstly, the effectiveness of type-2 fuzzy classifier to handle different levels of noises in unseen data was investigated, given that the classifier itself was only trained with noiseless data. Based on the experimental results on two synthetic data sets, it was shown that the performance of type-2 fuzzy classifier is at least comparable with competing type-1 fuzzy classifier when the test data is not contaminated with noise. When the noise level is raised, type-2 classifier consistently outperforms type-1 counterpart. The significance of this finding is that it enables the use of simulated data as an alternative to train the classifier when the real data is not readily available or the cost to collect it is expensive. Secondly, the advantage of type-2 classifier in handling the imprecise boundary associated with improper feature extraction was demonstrated through the classification of real-world automotive problem. The classification results revealed that type-2 classifier has the edge over type-1 classifier when the decision boundary is imprecise. Thirdly, since all fuzzy classifiers were optimised by genetic algorithm (GA), the robustness of type-2 framework against randomness in classifier designs was examined. The lower standard deviations in training and testing accuracies suggested that different type-2 classifiers optimised by GA can deliver more consistent performances. A more consistent design not only reduce the classifier training time but also improve the reliability of the designed classifier. Hopefully, this can balance the tradeoff between the computation load and the reliability of the classifier well. The experiment elucidated the robustness of type-2 fuzzy classifier not limited to the simulation problems, but also applies to the real world classification problems where the uncertainty is usually difficult to be estimated. As such, there is a strong reason to use type-2 fuzzy classifier as the performance of type-2 FRBC is at least comparable, if not better than type-1 FRBC. As a rule of thumb, for applications where the computation speed is not a major consideration, type-2 classifier should be adopted.

A hybrid learning algorithm for fuzzy rule-based classifier was presented in Chapter 5. The study was motivated by the facts that conventional fuzzy rule-based classifier is unable to handle high dimensional classification task and the existing learning techniques are either based on empirical or structural risk minimisation. Fuzzy c-means clustering and genetic algorithm were used to optimise the number of rules and antecedent parameters. Since the Takagi-Sugeno-Kang (TSK) fuzzy logic system bears a resemblance to a support vector machine (SVM), it was then possible to use SVM to learn the consequent parts of the TSK fuzzy classifier. The performance of the proposed hybrid fuzzy classifier was verified through extensive tests and comparison with other popular methods. The resulting hybrid fuzzy classifier has a compact rule base and good generalisation capabilities compared to existing algorithms in the literature. One key contribution of this study is that the curse of dimensionality which is often associated with fuzzy rule-based classifier can be avoided.

Chapter 6 proposed a fuzzy rule-based initialisation procedure for fuzzy K-nearest neighbor (fuzzy K-NN) classifier. The aims were to alleviate the impact of insufficient training data as well as to to handle the input noise. These problems have always been the bottleneck of conventional crisp K-NN and fuzzy K-NN classifiers. The performance of the proposed algorithm was evaluated on four UCI data sets and Ford automotive data set. The results and analysis showed that the decision boundaries produced by fuzzy rule-based K-NN classifier is very flexible. Therefore, it can resolve the situation where the training data is insufficient or corrupted with noise. In contrast, crisp K-NN and fuzzy K-NN may not be able to overcome the same situations as there is no way to change the decision boundary once the parameter K has been fixed. Based on the promising experimental results obtained, it appears that fuzzy rule-based K-NN has strong potential to outperform crisp and fuzzy K-NN when it comes to practical problems where uncertainty is hard to estimate.

In Chapter 7, the practical applicability of fuzzy rule-based classifiers for inverter-fed induction motor fault detection is investigated. It is worthwhile to mention that only type-1 singleton FRBC is used instead of the extensional FRBCs. This is because type-1 FRBC is sufficient to deliver perfect classification accuracy for the given problem. Therefore, the extra degrees-of-freedom of type-2 FRBC is not required. This is consistent with the general guideline mentioned in [124] that type-2 fuzzy logic systems should be tried only if type-1 fuzzy logic systems have failed.

It should be noted that genetic algorithms have been applied to learn both the antecedent and consequent part of fuzzy rules in this thesis. The transparency of the resulting rule base is not considered to be of importance as compared to the classification accuracy. In such cases, the fuzzy rule-based classifier may become a black box and one can question the rationale for applying fuzzy classifier instead of other classifiers like, e.g., neural networks. Apparently, this is greatly depending on the choice of the classifier designer to decide the relative importance of the interpretability of the rule and classification accuracy. It is always possible to design a fuzzy rule which is easy to understand by limiting the number of antecedents and employing rules selection procedure [125, 126]. As such, the flexibility in deciding the trade-off between these two design considerations should be regarded as an asset but not as a liability. Recently evolutionary multiobjective optimization (EMO) algorithms were used to search for singleton type-1 fuzzy rule-based systems with different accuracy-complexity tradeoffs [127, 128]. Therefore, for future work it is worthwhile to investigate the use of EMO on the extensional fuzzy logic systems (non-singleton and type-2 FRBs).

Non-singleton input in this thesis is modeled as type-1 fuzzy set. It has been argued that when the input is time-varying, type-2 fuzzy sets may be able to capture the input uncertainties more effectively [37]. Thus, it would be interesting to investigate the effect of using type-2 fuzzy sets as the non-singleton input in future work. It is also worth noting that the hybrid learning algorithm presented in Chapter 5 is not limited to type-1 fuzzy rule-based classifier. The design procedures may lay the foundation for designing a type-2 fuzzy classifier as well. It should be evident that design methodologies for a type-2 fuzzy system are very natural extensions of those for a type-1 fuzzy system.

Author's Publications

The author has contributed to the following publications:

Book Chapter

[1] T. W. Chua and W. W. Tan, "Interval type-2 fuzzy system for ECG arrhythmic classification," in Fuzzy Systems in Bioinformatics and Computational Biology, Y. Jin, and L. Wang (Eds.), pp. 297-314, Springer, 2009.

Book Review

[1] W. W. Tan and T. W. Chua, "Review of uncertain rule-based fuzzy logic systems: introduction and new directions," IEEE Computational Intelligence Magazine, vol. 2, no. 1, pp. 72-73, 2007.

Journal Papers

[1] T. W. Chua and W. W. Tan, "Non-singleton genetic fuzzy logic system for arrhythmias classification," manuscript accepted to be published in *Engineering Applications of Artificial Intelligence*, doi: 10.1016/j.engappai.2010.10.003.

[2] T. W. Chua and W. W. Tan, "Comparative analysis of the robustness of type-2 fuzzy rule-based classifier," manuscript submitted to *IEEE Trans. Fuzzy Syst.*

[3] T. W. Chua and W. W. Tan, "Fuzzy rule-based K-nearest neighbor classifier", manuscript submitted to *Information Sciences.*

[4] T. W. Chua, W. W. Tan, Z.-X. Wang, and C. S. Chang, "Hybrid time-frequency domain analysis for robust inverter-fed induction motor fault detection," manuscript submitted to *IEEE/ASME Trans. Mechatronics.*

Conference Papers

[1] T. W. Chua, W. W. Tan, Z.-X. Wang, and C. S. Chang, "Hybrid time-frequency domain analysis for inverter-fed induction motor fault detection," manuscript submitted to *IEEE International Symposium of Industrial Electronics*, Bari, Italy, 2010.

[2] Z.-X. Wang, C. S. Chang, T. W. Chua, and W. W. Tan, "Ensemble and individual noise reduction method for induction-motor signature analysis," manuscript accepted for *8th IET International Conference on Advances in Power System Control, Operation and Management (APSCOM)*, Hong Kong, 2009.

[3] T. W. Chua and W. W. Tan, "A new fuzzy rule-based initialization method for K-nearest neighbor classifier," in *Proc. of FUZZ-IEEE Conference*, Jeju, Korea, 2009.

[4] T. W. Chua and W. W. Tan, "Genetically evolved fuzzy rule-based classifiers and application to automotive classification," in *Proc. of the Simulated Evolution and Learning (SEAL) Conference*, Melbourne Australia, pp. 101-110, 2008.

[5] T. W. Chua and W. W. Tan, “EFSVM-FCM: Evolutionary fuzzy rule-based support vector machines classifier with FCM clustering,” in *Proc. of the FUZZ-IEEE Conference*, Hong Kong, pp. 606-612, 2008.

[6] T. W. Chua and W. W. Tan, “GA optimisation of non-singleton fuzzy logic system for ECG classification,” in *Proc. of the IEEE Congress of Evolutionary Computation (CEC)*, Singapore, pp. 1677-1684, 2007.

[7] W. W. Tan, C. L. Foo, and T. W. Chua, “Type-2 fuzzy system for ECG arrhythmic classification,” in *Proc. of FUZZ-IEEE Conference*, London, UK, pp. 859-864, 2007.

Bibliography

[1] R. E. Bellman, R. Kalaba, and L. A. Zadeh, "Abstraction and pattern classification," *J. Math. Anal. Appl.*, vol. 13, pp. 1–7, 1966.

[2] L. I. Kuncheva, *Fuzzy Classifier Design.* Heidelberg, Germany: Physica-Verlag, 2000.

[3] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 1999.

[4] J. C. Bezdek and S. K. Pal, *Fuzzy Models For Pattern Recognition: Methods That Search for Structures in Data.* New York: IEEE Press, 1992.

[5] S. Abe, *Support Vector Machines for Pattern Classification.* London: Springer, 2005.

[6] J. M. Mendel and R. I. John, "Type-2 fuzzy sets made simple," *IEEE Trans. Fuzzy Syst.*, vol. 10(2), pp. 117–127, 2002.

[7] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* Norwell, Massachusetts: Kluwer, Academic Publishers, 1981.

[8] M. Grabisch and F.Dispot, “A comparison of some methods of fuzzy classification on real data,” in *Proc. of IIZUKA Conference*, Iizuka, Japan, 1992, pp. 659–662.

[9] W. Pedrycz, “Fuzzy sets in pattern recognition: Accomplishments and challenges,” *Fuzzy Sets and Systems*, vol. 90, pp. 171–176, 1997.

[10] 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*, vol. 20(1), pp. 21–45, 1999.

[11] J. Watada, H. Tanaka, and K. Asay, “Fuzzy discriminant analysis in fuzzy groups,” *Fuzzy Sets and Systems*, vol. 19, pp. 261–271, 1986.

[12] M. Keller and D. J. Hunt, “Incorporating fuzzy membership functions into the perceptron algorithm,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 7(6), pp. 693–699, 1985.

[13] J. Valente de Oliveira and W. Pedrycz, *Advances in Fuzzy Clustering and its Applications.* New York, NY, USA: John Wiley & Sons, Inc., 2007.

[14] G. C. Mouzouris and J. Mendel, “Nonsingleton fuzzy logic systems: Theory and application,” *IEEE Trans. Fuzzy Syst.*, vol. 5(1), pp. 56–71, 1997.

[15] Y. Hayashi, J. J. Buckley, and E. Czogala, “Fuzzy neural network with fuzzy signals and weights,” *International Journal of Intelligent Systems*, pp. 527–537, 1993.

[16] W. Wei and J. M. Mendel, “A fuzzy logic method for modulation classification in nonideal environments,” *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 3, pp. 333–344, 1999.

[17] H. Wu and J. M. Mendel, “Classification of battlefield ground vehicles using acoustic features and fuzzy logic rule-based classifiers,” *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 1, pp. 56–72, 2007.

[18] R. I. John, P. Innocent, and M. Barnes, “Neuro-fuzzy clustering of radiographic tibia image data using type-2 fuzzy sets,” *Information Sciences*, vol. 125, no. 1, 2000.

[19] Q. Liang and J. Mendel, “MPEG VBR video traffic modeling and classification using fuzzy technique,” *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 1, pp. 183–193, Feburary 2001.

[20] H. B. Mitchell, “Pattern recognition using type-II fuzzy sets,” *Information Sciences*, vol. 170, pp. 409–418, 2005.

[21] J. Zeng and Z.-Q. Liu, “Type-2 fuzzy hidden markov models and their application to speech recognition,” *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 3, pp. 454–467, 2006.

[22] ——, “Type-2 fuzzy markov random fields and their application to handwritten chinese character recognition,” *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 3, pp. 747–760, 2008.

[23] J. Zeng, L. Xie, and Z.-Q. Liu, “Type-2 fuzzy gaussian mixture models,” *Pattern Recognition*, vol. 41, no. 12, pp. 3636–3643, 2008.

[24] C. Hwang and F. Rhee, "Uncertain fuzzy clustering: interval type-2 fuzzy approach to c-means," *IEEE Trans. Fuzzy Syst.*, vol. 15(1), pp. 107–120, 2007.

[25] H. Ishibuchi, N. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," *Fuzzy Sets and Systems*, vol. 52, pp. 21–32, 1992.

[26] ——, "Efficient fuzzy partition of pattern space for classification problems," *Fuzzy Sets and Systems*, vol. 59, pp. 295–304, 1993.

[27] D. P. Mandal, "A multivalued approach for uncertainty management in pattern recognition problems using fuzzy sets," PhD Thesis, Indian Statistical Institute, Calcutta, India, 1992.

[28] D. P. Mandal, C. A. Murthy, and S. K. Pal, "Formulation of a multivalued recognition system," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 607–620, 1992.

[29] S. K. Pal and D. P. Mandal, "Linguistic recognition system based on approximate reasoning," *Information Sciences*, vol. 61, pp. 135–161, 1992.

[30] S. Abe and M. Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 18–28, 1995.

[31] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 601–618, 1999.

[32] D. Nauck and R. Kruse, "How the learning of rule weights affects the interpretability of fuzzy systems," in *Proc. 7th IEEE-FUZZ Conference*, Anchorage, AK, May 1998, pp. 1235–1240.

[33] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Trans. Fuzzy Syst.*, vol. 9(4), pp. 506–515, 2001.

[34] L.-X. Wang, "Analysis and design of fuzzy systems," PhD Thesis, University of Southern California, Los Angeles, CA, 1992.

[35] ——, *Adaptive fuzzy systems and control: design and stability analysis.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994.

[36] L.-X. Wang and J. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 807–814, Sep 1992.

[37] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions.* Upper Saddle River, NJ: Prentice Hall, 2001.

[38] C.-H. Wang, T.-P. Hong, and S.-S. Tseng, "Integrating fuzzy knowledge by genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 2, no. 4, pp. 138–149, Nov 1998.

[39] M. Bereta and T. Burczyński, "Comparing binary and real-valued coding in hybrid immune algorithm for feature selection and classification of ecg signals," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 5, pp. 571–585, 2007.

[40] R. Jager, “Fuzzy logic in control,” PhD Thesis, Delft University of Technology, 1995.

[41] A. Kaufman and M. M. Gupta, *Introduction to Fuzzy Arithmetic: Theory and Applications.* New York: Van Nostrand Reinhold, 1991.

[42] R. E. Kerber *et al.*, “Automatic external defibrillators for public access defibrillation: Recommendations for specifying and reporting arrhythmia analysis algorithm performance, incorporating new waveforms, and enhancing safety,” *Circulation 95*, pp. 1677–1682, 1997.

[43] MIT-BIH Arrhythmia Database. Available from: Massachusetts Inst. Technol, Cambridge, MA, 1988.

[44] X.-S. Zhang, Y.-S. Zhu, N. V. Thakor, and Z.-Z. Wang, “Detecting ventricular tachycardia and fibrillation by complexity measure,” *IEEE Trans. Biomed. Eng.*, vol. 46(5), pp. 837–843, 1990.

[45] S. Mitaim and B. Kosko, “The shape of fuzzy sets in adaptive function approximation,” *IEEE Trans. Fuzzy Syst.*, vol. 9(4), pp. 637–656, 2002.

[46] L. I. Kuncheva, “How good are fuzzy if-then classifiers?” *IEEE Trans. Syst., Man, Cybern. B*, vol. 30(4), pp. 501–509, 2000.

[47] T.-H. S. Li, N. R. Guo, and C. L. Kuo, “Design of adaptive fuzzy model for classification problem,” *Engineering Applications of Artificial Intelligence*, vol. 18(3), pp. 297–306, 2005.

[48] Y.-H. Shi, R. Eberhart, and Y.-B. Chen, “Implementation of evolutionary fuzzy systems,” *IEEE Trans. Fuzzy Syst.*, vol. 7(2), pp. 109–119, 1999.

[49] N. V. Thakor, Y. S. Zhu, and K. Y. Pan, “Ventricular tachycardia and fibrillation detection by a sequential hypothesis testing algorithm,” *IEEE Trans. Biomed. Eng.*, vol. 37(9), pp. 837–843, 1990.

[50] U. Ayesta, L. Serrano, and I. Romero, “Complexity measure revisited: A new algorithm for classifying cardiac arrhythmias,” in *Proc. of the 23rd Annual EMBS International Conference*, 2001, pp. 1589–1591.

[51] S. W. Chen, P. M. Clarkson, and Q. Fan, “A robust sequential detection algorithm for cardiac arrhythmia classification,” *IEEE Trans. Biomed. Eng.*, vol. 43(11), pp. 1120–1125, 1996.

[52] E. Chowdhury and L. C. Ludeman, “Discrimination of cardiac arrythmias using a fuzzy rule-based method,” *Computers in Cardiology*, pp. 549–552, 1994.

[53] J. Ruiz, E. Aramendi, S. R. de Gauna, A. lazkano, L. A. Leturiondo, and J. J. Gutiérrez, “Distinction of ventricular fibrillation and ventricular tachycardia using cross correlation,” *Computers in Cardiology*, vol. 30, pp. 729–732, 2003.

[54] Y. Sun, K. L. Chan, and S. M. Krishnan, “Life-threatening ventricular arrhythmia recognition by nonlinear descriptor,” *Biomedical Engineering Online*, vol. 4(6), 2005.

[55] D. Wu and W. W. Tan, “Type-2 FLS modeling capability analysis,” in *Proc. of FUZZ-IEEE Conference*, Reno, USA, May 2005, pp. 242–247.

[56] "Advances in type-2 fuzzy sets and systems," *Information Sciences*, vol. 177, pp. 84–110, 2007.

[57] J. M. Mendel, H. Hagras, and R. I. John, "Standard background material about interval type-2 fuzzy logic systems that can be used by all," 2008. [Online]. Available: http://www.ieee-cis.org/technical/standards/

[58] B. I. Choi and F. Rhee, "Interval type-2 fuzzy membership function generation methods for pattern recognition," *Information Sciences*, vol. 179, no. 13, pp. 2102–2122, 2009.

[59] O. Wolkenhauer, *Data Engineering: Fuzzy Mathematics in Systems Theory and Data Analysis.* New York: John Wiley and Sons, 2001.

[60] ——, *Possibility Theory with Applications to Data Analysis.* Tauton, Somerset, England: Research Studies Press, 1998.

[61] S. Gao, W. Wu, C. hui Lee, and T. seng Chua, "A maximal figure-of-merit (MFoM)-learning approach to robust classifier design for text categorization," *ACM Transactions on Information Systems*, vol. 24, pp. 190–218, 2006.

[62] H. Hu and J. Li, "Using association rules to make rule-based classifiers robust," in *Proc. of ADC Conference*, Newcastle, Australia, 2005, pp. 47–54.

[63] R. Nanopoulos, A. N. Papadopoulos, T. Welzer-druzovec, and Y. Manolopoulos, "Item-based classification for robustness against noise?" in *Proc. ADMK Conference*, Thessaloniki, Greece, 2006.

[64] X. Chen, Y. Li, R. Harrison, and Y. Zhang, "Type-2 fuzzy logic-based classifier fusion for support vector machines," *Applied Soft Computing Journal*, vol. 8, no. 3, pp. 1222–1231, 2008.

[65] R. I. John, P. R. Innocent, and M. R. Barnes, "Type-2 fuzzy sets and neuro-fuzzy clustering of radiographic tibia images," in *Proc. of IEEE-FUZZ Conference*, vol. 2, 1998, pp. 1373–1376.

[66] J. Zeng and Z. Q. Liu, "Type-2 fuzzy sets for pattern classification: A review," in *Proc. of FOCI Conference*, Honolulu, HI, April 2007, pp. 193–200.

[67] J. T. Starczewski, "What differs interval type-2 FLS from type-1 FLS?" in *Proc. of ICAISC Conference*, Zakopane, Poland, June 2004, pp. 381–387.

[68] T. W. Chua and W. W. Tan, "GA optimisation of non-singleton fuzzy logic system for ecg classification," in *Proc. of IEEE-CEC Conference*, 2007, pp. 1677–1684.

[69] V. Ravi, P. J. Reddy, and H. Zimmermann, "Pattern classification with principal component analysis and fuzzy rule bases," *European Journal of Operational Research*, vol. 126, no. 3, pp. 526–533, 2000.

[70] M. Abou-Nasr and L. Feldkamp, "Ford classification challenge." [Online]. Available: http://home.comcast.net/~nn_classification/

[71] C. Eagen, L. Feldkamp, S. Ebenstein, K. Prakah-Asante, Y. Rodin, and G. Smith, "System and method for detecting presence of a human in a vehicle," U.S. Patent 7,353,088B2, 2008.

[72] T. W. Chua and W. W. Tan, “Genetically evolved fuzzy rule-based classifiers and application to automotive classification,” in *Proc. of SEAL Conference*, Melbourne, Australia, 2008, pp. 101–110.

[73] F. Rhee, “Uncertain fuzzy clustering: Insights and recommendations,” *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 44–56, Febuary 2007.

[74] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, “Selecting fuzzy if-then rules for classification problems using genetic algorithms,” *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 260–270, Aug. 1995.

[75] B. Jin, Y. C. Tang, and Y. Q. Zhang, “Support vector machines with genetic fuzzy feature transformation for biomedical data classification,” *Information Sciences*, vol. 177, pp. 476–489, 2007.

[76] Y. Chen and J. Z. Wang, “Support vector learning for rule-based classification systems,” *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 716–728, Dec. 2003.

[77] J. H. Chiang and P. Y. Hao, “Support vector learning mechanism for fuzzy rule-based modeling: A new approach,” *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 1, pp. 1–12, Feb. 2004.

[78] C. F. Juang, S. H. Chiu, and S. W. Chang, “A self-organizing TS-type fuzzy network with support vector learning and its application to classification problems,” *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 5, pp. 998–1008, Oct. 2007.

[79] C. F. Juang and C. T. Lin, “An on-line self-constructing neural fuzzy inference network and its applications,” *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 1, pp. 12–32, Feb. 1998.

[80] C. L. Blake and C. J. Merz, “UCI repository of machine learning databases,” University of California, Department of Information and Computer Science, Irvine, CA, 1998.

[81] 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, pp. 583–591, 2007.

[82] L. B. Goncalves, M. M. B. R. Vellasco, M. A. C. Pacheco, and F. J. de Souza, “Inverted hierarchical neuro-fuzzy BSP system: A novel neuro-fuzzy model for pattern classification and rule extraction in databases,” *IEEE Trans. Syst., Man, Cybern. C*, vol. 36, no. 2, Mar. 2006.

[83] M. Russo, “FuGeNeSys - a fuzzy genetic neural system for fuzzy modeling,” *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 3, pp. 373–388, Aug. 1998.

[84] R. Paredes and E. Vidal, “Learning weighted metrics to minimize nearest-neighbor classification error,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1100–1110, 2006.

[85] R. D. Short and K. Fukunaga, “The optimal distance measure for nearest neighbor classification,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 622–627, 1981.

[86] C. Domeniconi, J. Peng, and D. Gunopulos, “Locally adaptive metric nearest-neighbor classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1281–1285, 2002.

[87] J. Peng, D. R. Heisterkamp, and H. Dai, “LDA/SVM driven nearest neighbor classification,” in *Proc. of the IEEE-CVPR Conference*, 2001, pp. 940–942.

[88] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[89] T. Cover, “Estimation by the nearest neighbor rule,” *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 50–55, 1968.

[90] J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy K-NN neighbor algorithm,” *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 4, pp. 580–585, 1985.

[91] W. Shang, H. Huang, H. Zhu, Y. Lin, Z. Wang, and Y. Qu, “An improved KNN algorithm - fuzzy KNN,” in *Proc. of CIS Conference*, Xi’an, China, 2005, pp. 741–746.

[92] F. Rhee and C. Hwang, “An interval type-2 fuzzy k-nearest neighbor,” in *Proc. of the 12th IEEE-FUZZ Conference*, vol. 2, 2003, pp. 802–807.

[93] W. Pedrycz, “Fuzzy sets in pattern recognition: accomplishments and challenges,” *Fuzzy Sets and Systems*, vol. 90, no. 2, pp. 171–176, 1997.

[94] A. Asuncion and D. Newman, “UCI machine learning repository,” 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[95] J. H. Friedman, “Flexible metric nearest neighbor classification,” Dept. of Statistics, Standford University, Tech. Rep., 1994.

[96] B. Ayhan, H. Trussell, M.-Y. Chow, and M.-H. Song, “On the use of a lower sampling rate for broken rotor bar detection with DTFT and AR-based spectrum methods,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1421 – 1434, 2008.

[97] W. Thomson and M. Fenger, “Current signature analysis to detect induction motor faults,” *IEEE Industry Applications Magazine*, vol. 7, no. 4, pp. 26 – 34, 2001.

[98] I. Y. Önel and M. E. H. Benbouzid, “Induction motor bearing failure detection and diagnosis: Park and concordia transform approaches comparative study,” *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 2, pp. 257–262, April 2008.

[99] D. Dorrell, W. Thomson, and S. Roach, “Analysis of airgap flux, current, and vibration signals as a function of the combination of static and dynamic airgap eccentricity in 3-phase induction motors,” *IEEE Trans. Ind. Appl.*, vol. 33, no. 1, pp. 24–34, Jan./Feb. 1997.

[100] C. Riley, B. Lin, T. Habetler, and G. Kliman, “Stator current harmonics and their causal vibrations: a preliminary investigation of sensorless vibration monitoring applications,” *IEEE Trans. Ind. Appl.*, vol. 35, no. 1, pp. 94–99, Jan./Feb. 1999.

[101] S. Poyhonen, P. Jover, and H. Hyotyniemi, “Signal processing of vibrations for condition monitoring of an induction motor,” *First International Symposium on Control, Communications and Signal Processing, 2004*, pp. 499–502, 2004.

[102] T. Wang, F. Wang, H. Bai, and G. Zhang, “Study on vibration behavior of large induction motors,” *Journal of Mechanical Strength*, vol. 31, no. 1, pp. 140–143, 2009.

[103] D. Boothman, E. Elgar, R. Rehder, and R. Wooddall, “Thermal tracking-a rational approach to motor protection,” *IEEE Trans. Power Apparatus and Syst.*, vol. PAS-93, no. 5, pp. 1335–1344, Sept. 1974.

[104] L. Eren and M. Devaney, “Bearing damage detection via wavelet packet decomposition of the stator current,” *IEEE Trans. Instrum. Meas.*, vol. 53, no. 2, pp. 431–436, Apr. 2004.

[105] M. E. H. Benbouzid, “A review of induction motors signature analysis as a medium for faults detection,” *IEEE Trans. Ind. Electron.*, vol. 47, no. 5, pp. 984–993, Oct 2000.

[106] H. Douglas, P. Pillay, and A. Ziarani, “A new algorithm for transient motor current signature analysis using wavelets,” *IEEE Trans. Ind. Appl.*, vol. 40, no. 5, pp. 1361–1368, Sept.-Oct. 2004.

[107] J.-H. Jung, J.-J. Lee, and B.-H. Kwon, “Online diagnosis of induction motors using MCSA,” *IEEE Trans. Ind. Electron.*, vol. 53, no. 6, pp. 1842–1852, Dec. 2006.

[108] M. E. H. Benbouzid, M. Vieira, and C. Theys, “Induction motors’ faults detection and localization using stator current advanced signal processing techniques,” *IEEE Trans. Power Electron.*, vol. 14, no. 1, pp. 14–22, Jan. 1999.

[109] H. Nejjari and M. E. H. Benbouzid, “Monitoring and diagnosis of induction motors electrical faults using a current park’s vector pattern learning approach,” *IEEE Trans. Ind. Appl.*, vol. 36, no. 3, pp. 730–735, May/Jun 2000.

[110] K. Kim and A. Parlos, “Induction motor fault diagnosis based on neuropredictors and wavelet signal processing,” *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 2, pp. 201–219, Jun 2002.

[111] D. Diallo, M. E. H. Benbouzid, D. Hamad, and X. Pierre, “Fault detection and diagnosis in an induction machine drive: A pattern recognition approach based on concordia stator mean current vector,” *IEEE Trans. Energy Convers.*, vol. 20, no. 3, pp. 512–519, Sept. 2005.

[112] F. Zidani, M. E. H. Benbouzid, D. Diallo, and M. Nait-Said, “Induction motor stator faults diagnosis by a current concordia pattern-based fuzzy decision system,” *IEEE Trans. Energy Convers.*, vol. 18, no. 4, pp. 469–475, Dec. 2003.

[113] F. Zidani, D. Diallo, M. E. H. Benbouzid, and R. Nait-Said, “A fuzzy-based approach for the diagnosis of fault modes in a voltage-fed PWM inverter

induction motor drive," *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 586–593, Feb. 2008.

[114] M. Zeraoulia, A. Mamoune, H. Mangel, and M. E. H. Benbouzid, "A simple fuzzy logic approach for induction motors stator condition monitoring," *Journal of Electrical Systems*, vol. 1, no. 1, pp. 15–25, 2005.

[115] P. Jover Rodríguez and A. Arkkio, "Detection of stator winding fault in induction motor using fuzzy logic," *Applied Soft Computing*, vol. 8, pp. 1112–1120, 2008.

[116] W. J. Park, S. H. Lee, W. K. Joo, and J. I. Song, "A mixed algorithm of pca and lda for fault diagnosis of induction motor," in *ICIC (2), Lecture Notes in Computer Science*, vol. 4682, 2007, pp. 934–942.

[117] Z. Wang, C. S. Chang, X. German, and W. W. Tan, "Online fault detection of induction motors using independent component analysis and fuzzy neural network," in *8th IET International Conference on Advances in Power System Control, Operation and Management (APSCOM)*, Hong Kong, Nov 2009, pp. 1–6.

[118] B. Akin, U. Orguner, H. Toliyat, and M. Rayner, "Low order PWM inverter harmonics contributions to the inverter-fed induction machine fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 610–619, Feb. 2008.

[119] Z. Wang, C. S. Chang, T. W. Chua, and W. W. Tan, "Ensemble and individual noise reduction method for induction-motor signature analysis," in

8th IET International Conference on Advances in Power System Control, Operation and Management (APSCOM), Hong Kong, Nov 2009, pp. 1–6.

[120] R. Hirvonen, "On-line condition monitoring of defects in squirrel cage motors," in *1994 Int. Conf. Electrical Machines*, vol. 2, Paris, France, pp. 267–272.

[121] M. E. H. Benbouzid and G. B. Kliman, "What stator current processing-based technique to use for induction motor rotor faults diagnosis?" *IEEE Trans. Energy Convers.*, vol. 18, no. 2, pp. 238–244, June 2003.

[122] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, June 2000.

[123] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 626–634, 1999.

[124] R. John and S. Coupland, "Type-2 fuzzy logic: A historical view," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 57–62, Febuary 2007.

[125] H. Ishibuchi, Y. Kaisho, and Y. Nojima, "Designing fuzzy rule-based classifiers that can visually explain their classification results to human users," in *Proc. of 3rd International Workshop on Genetic and Evolving Fuzzy Systems*, Witten-Bommerholz, Germany, 2007, pp. 5–10.

[126] H. Ishibuchi and Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *Int. J. Approx. Reasoning*, vol. 44, no. 1, pp. 4–31, 2007.

[127] H. Ishibuchi, "Multiobjective genetic fuzzy systems: Review and future research directions," in *Proc. of FUZZ-IEEE Conference*, July 2007, pp. 1–6.

[128] ——, "Evolutionary multiobjective design of fuzzy rule-based systems," in *Proc. of IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, April 2007, pp. 9 –16.

Appendix

Matlab Codes to Generate Gaussian Data:

```
% n = number of data; theta_G = noise level (0 to 1); x = Gaussian data

theta_G = theta_G/1.5;

x1 = theta_G*randn(1,n)+0.3;  x2=theta_G*randn(1,n)-0.3;

y1 = theta_G*randn(1,n)+0.3;  y2=theta_G*randn(1,n)-0.3;

x = [x1 x2; y1 y2]';

xlabel = [ones(1,n) -ones(1,n)]'; %labels
```

Matlab Codes to Generate Clown Data:

```
% n = number of data; theta_C = noise level (0 to 1); x = Clown data

n = round(n/2); theta_C = theta_C*2;

x1 = (6*rand(n,1)-3);

x2 = x1.^2 + randn(n,1);

x0 = theta_C*randn(n,2)+(ones(n,1)*[0 6]);

x = [x0;[x1 x2]];

x = (x-ones(2*n,1)*mean(x))*diag(1./std(x)); %normalisation

xlabel = [ones(n,1); -ones(n,1)]; %labels
```