

A Novel Framework for the Implementation and Evaluation of Type-1 and Interval Type-2 ANFIS



Chao Chen

School of Computer Science
University of Nottingham

This dissertation is submitted for the degree of
Doctor of Philosophy

Nottingham

January 2018

I would like to dedicate this thesis to my loving family.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Chao Chen

January 2018

Acknowledgements

I would like to thank my supervisors Prof. Jon Garibaldi, Dr Jamie Twycross and Prof. Robert John for their guidance and support during this PhD research.

I would also like to thank Dr Jaume Bacardit, Dr Paul Tennent and Dr Michel Valstar for their support in my PhD application.

I would like to extend my thanks to all my colleagues, friends for the great memories we have made together in Nottingham, Vancouver, Naples and so on.

Finally, I would like to especially thank my wife and my precious daughter for bringing joy, love, and happiness into my life.

Abstract

This thesis explores a novel framework for implementing and evaluating type-1 (T1) and interval type-2 (IT2) models of Adaptive Network Fuzzy Inference Systems (ANFIS). A fundamental requirement for this research is the capability to reliably and efficiently implement ANFIS models. In the last ten years, many studies have been devoted to creating IT2 ANFIS models. However, a clear architecture for IT2 ANFIS has not yet been presented. This somehow has been an obstacle to the research of IT2 ANFIS and its application to real-world problems. In this thesis, we introduce an extended ANFIS architecture that can be used for both T1 and IT2 models. In conjunction with this, a crucial obstacle to the use of IT2 fuzzy systems in general (and including IT2 ANFIS) is that IT2 models are often more computationally expensive than T1 models. Note that a bottle-neck for IT2 ANFIS is to aggregate the output of each rule produced by the inference process of the Karnik-Mendel (KM) algorithm. Many enhanced algorithms have been proposed to improve the computational efficiency of the KM algorithm. However, all of these algorithms are still based on iterative procedures to determine the switch points required for the lower and upper bounds of defuzzification. This thesis introduces a ‘direct approach’ which can be used to determine these switch points based on derivatives, without the need for multiple iterations. When comparing various models (including T1 and IT2 ANFIS models), it is necessary to conduct fair comparisons. Partly to address this issue, a new accuracy measure is proposed which combines the best features of various alternative measures without having their common drawbacks. Experimental comparisons are made between T1 and IT2 ANFIS using the novel accuracy measure in addition to the commonly used RMSE, on both synthetic and

real-world data. Finally, it is shown that IT2 ANFIS models are not easy to optimise from scratch due to difficulties with the output intervals, that are not present in T1 ANFIS models. Detailed experiments are carried out to evaluate the comparative performance of IT2 ANFIS models, including the best method for initialising the IT2 membership functions. In summary, a coherent framework for efficiently implementing IT2 ANFIS models and fairly evaluating their comparative performance is presented. This framework allows the implementation of IT2 ANFIS in any application context, and the resultant performance to be carefully considered, since clear performance improvement compared to T1 ANFIS may not always be found.

Table of contents

List of figures	xv
List of tables	xix
Nomenclature	xxiii
1 Introduction	1
1.1 Background and Motivation	2
1.2 Aims and Objectives	4
1.3 Contributions to Knowledge	6
1.4 Organisation of the thesis	6
2 Background	9
2.1 Introduction	9
2.2 Fuzzy Sets and Systems	10
2.2.1 Basic Definitions	11
2.2.2 Fuzzy Inference System	17
2.2.3 The Adaptive Network-based Fuzzy Inference System	24
2.3 Accuracy measure	30
2.3.1 Scale-dependent Measures	32
2.3.2 Percentage-based Measures	33
2.3.3 Relative-based Measures	35
2.4 Summary	37

3	A New Accuracy Measure: UMBRAE	41
3.1	Introduction	41
3.2	A New Accuracy Measure	42
3.3	Evaluation and Results	44
3.3.1	Evaluation with Synthetic Data	45
3.3.2	Evaluation with the M3-Competition Data	50
3.4	Discussion	55
3.5	Summary	62
4	A Direct Approach for the Switch Points in the KM Algorithm	65
4.1	Introduction	65
4.2	The KM algorithm	66
4.3	The iterative algorithms	67
4.3.1	The EKM algorithm	67
4.3.2	The EIASC algorithm	68
4.4	The Direct Approach (DA) algorithm	71
4.5	Proof of the DA algorithm	78
4.6	Experimental comparison	79
4.6.1	Examples of IT2 fuzzy sets	79
4.6.2	Generalised IT2 fuzzy sets	81
4.7	Discussion	82
4.8	Further Investigation	84
4.9	Further Optimisation	85
4.9.1	The EKM algorithm	85
4.9.2	An enhanced DA algorithm	85
4.9.3	A non-derivative based DA algorithm	86
4.9.4	Computational complexity	88
4.9.5	Computational time	91
4.10	Further Discussion	91
4.11	Summary	95

5	An Extended ANFIS Architecture	97
5.1	Introduction	97
5.2	An extended architecture of ANFIS	98
5.2.1	T1 ANFIS	99
5.2.2	IT2 ANFIS	102
5.3	The learning algorithms of ANFIS	105
5.3.1	The gradient decent method	105
5.3.2	Linear equation for T1 ANFIS	106
5.3.3	Linear equation for IT2 ANFIS	107
5.3.4	LSE and its inappropriateness for IT2 ANFIS	107
5.4	Evaluation	109
5.4.1	Training with pure LSE	110
5.4.2	Training with BP-LSE	111
5.5	Discussion	114
5.6	Summary	115
6	Experimental Comparisons between T1 and IT2 ANFIS	117
6.1	Introduction	117
6.2	ANFIS Architecture	118
6.3	Evaluation	120
6.3.1	Mackey-Glass time series prediction	121
6.3.2	Low voltage electrical line length estimation	126
6.3.3	Medium voltage electrical line maintenance cost estimation	127
6.3.4	Statistical comparison	130
6.4	Discussion	133
6.5	Summary	135
7	Conclusions	137
7.1	Thesis Summary	137
7.2	Contributions	140

7.3	Limitations	143
7.4	Future Work	144
References		147

List of figures

2.1	Examples of two T1 fuzzy sets, A (triangle) and B (trapezoidal).	12
2.2	A 3D example of an IT2 fuzzy set.	15
2.3	A 2D example of an IT2 fuzzy set on the xu -plane.	15
2.4	An illustration of (a) union and (b) intersection of two T1 fuzzy sets A and B.	16
2.5	General structure of a T1 FIS.	19
2.6	General structure of a T2 FIS.	19
2.7	Three types of fuzzy reasoning mechanisms in Fig. 2.7.	25
2.8	The typical structure of ANFIS.	26
3.1	Evaluation on the resistance of accuracy measures to a single forecasting outlier.	46
3.2	Evaluation on the symmetry of accuracy measures to over-estimates and under-estimates.	48
3.3	Evaluation on the scale dependency of Accuracy Measures.	49
3.4	Box-and-whisker plot and kernel density estimates for the absolute errors used by MAE.	56
3.5	Box-and-whisker plot and kernel density estimates for the squared errors used by RMSE.	56
3.6	Box-and-whisker plot and kernel density estimates for the absolute scaled errors used by MASE.	56

3.7	Box-and-whisker plot and kernel density estimates for the absolute scaled errors used by AvgRelMAE (log-scale).	57
3.8	Box-and-whisker plot and kernel density estimates for the relative absolute errors used by MRAE and GMRAE (log-scale, forecasts with zero or undefined error excluded).	57
3.9	Box-and-whisker plot and kernel density estimates for the absolute percentage errors used by MAPE.	57
3.10	Box-and-whisker plot and kernel density estimates for the scaled percentage errors used by sMAPE.	58
3.11	Box-and-whisker plot and kernel density estimates for the bounded relative absolute errors used by UMBRAE (using the naïve errors as the benchmark).	58
4.1	An illustration of the DA algorithm showing the position of the switch point for an arbitrary example where N is 10.	72
4.2	A comparison of computational time costs for different algorithms based on generalised bell-shaped IT2 fuzzy sets described in Section 4.6.	83
4.3	A comparison of computational time costs for different algorithms based on generalised randomly-shaped IT2 fuzzy sets described in Section 4.6.	83
4.4	A comparison of optimised algorithms based on bell-shaped fuzzy sets in R.	92
4.5	A comparison of optimised algorithms based on random-shaped fuzzy sets in R.	92
4.6	A comparison of optimised algorithms (excluding EIASC) based on bell-shaped fuzzy sets in R.	93
4.7	A comparison of optimised algorithms (excluding EIASC) based on random-shaped fuzzy sets in R.	93

4.8	A comparison of optimised algorithms based on bell-shaped fuzzy sets in Matlab.	94
4.9	A comparison of optimised algorithms based on random-shaped fuzzy sets in Matlab.	94
5.1	An example of the extended ANFIS architecture.	99
5.2	An illustration of $F \cap A$ based on the <i>min</i> operator, where F denotes the fuzzy input generated by singleton fuzzification and A denotes the antecedent fuzzy set.	101
5.3	An illustration of $F \cap A$ based on the <i>min</i> operator, where F denotes the fuzzy input generated by non-singleton fuzzification and A denotes the antecedent fuzzy set.	101
5.4	An illustration of the LSE for its inappropriateness in IT2 ANFIS, compared to T1 ANFIS.	109
5.5	The estimation and training error curves for the training data during the pure LSE training epochs with IRIS data (batch mode). . .	111
5.6	The estimation and training error curves for the training data during the pure LSE training epochs with IRIS data (online mode). . .	112
5.7	The estimation and training error curves for the training data during the hybrid (BP-LSE) training epochs with IRIS data (batch mode).	113
5.8	The estimation and training error curves for the training data during the hybrid (BP-LSE) training epochs with IRIS data (online mode).	113
6.1	An illustration of the IT2 membership function where $\{\underline{a}' = 1, \bar{a}' = 3, b' = 2, c' = 5\}$	119

List of tables

2.1	Two passes in the hybrid learning procedure of ANFIS	29
3.1	Results on M3-Competition data at first six forecasting horizons. . .	52
3.2	Spearman's rank correlation coefficient of the rankings in Table 3.1.	53
3.3	Results with a 3% trimming level on M3-Competition data at first six forecasting horizons.	54
3.4	Spearman's rank correlation coefficient of the rankings in Table 3.3.	55
3.5	Ratings of Accuracy Measures.	62
4.1	The EKM algorithm for computing the centroid endpoints (c_l and c_r) of an IT2 Fuzzy Set.	69
4.2	The EIASC algorithm for computing the centroid endpoints (c_l and c_r) of an IT2 Fuzzy Set.	70
4.3	The switch points L and R obtained by three algorithms based on the example fuzzy sets.	81
4.4	The DAND algorithm for computing the centroid endpoints (c_l and c_r) of an IT2 Fuzzy Set.	87
4.5	The computational complexity of EKM based on the number of cal- culations and comparisons for the core part of its implementations for c_l	88
4.6	The computational complexity of EIASC based on the number of calculations and comparisons for the core part of its implementa- tions for c_l	89

4.7	The computational complexity of DA based on the number of calculations and comparisons for the core part of its implementations for c_l	89
4.8	The computational complexity of DAND based on the number of calculations and comparisons for the core part of its implementations for c_l	90
4.9	A summary of the computational complexity of different algorithms based on the number of calculations and comparisons for the core part of their implementations for c_l	90
6.1	Training/Testing RMSE by different Models for predicting the MG series with different noise.	123
6.2	Training/Testing MASE by different Models for predicting MG series with different level of noise.	124
6.3	Training/Testing MRAE by different Models for predicting MG series with different level of noise.	124
6.4	Training/Testing GMRAE by different Models for predicting MG series with different level of noise.	125
6.5	Training/Testing UMBRAE by different Models for predicting MG series with different level of noise.	125
6.6	Five example data samples for the problem of low-voltage electrical line length estimation.	127
6.7	Training/Testing RMSE on low-voltage electricity data (5-fold cross-validation).	128
6.8	Training/Testing errors on low-voltage electricity data (5-fold cross-validation) based on other accuracy measures.	128
6.9	Five example data samples for the problem of medium-voltage electrical line maintenance cost estimation.	129
6.10	Training/Testing RMSE on medium-voltage electricity data (5-fold cross-validation).	130

6.11	Training/Testing errors on medium-voltage electricity data (5-fold cross-validation) based on other accuracy measures.	130
6.12	Statistical comparisons of models (A and B) based on absolute errors of training data.	131
6.13	Statistical comparisons of models (A and B) based on absolute errors of testing data.	132

Nomenclature

Acronyms / Abbreviations

2D two-dimensional

3D three-dimensional

ANFIS Adaptive-Neural-Based Fuzzy Inference System

AvgRelMAE Average Relative Mean Absolute Error

DA Direct Approach

DAND Non-derivative based DA Algorithm

EIASC Enhanced Iterative Algorithm with Stopping Condition

EKM Enhanced Karnik-Mendel

FIS Fuzzy Inference System

sMAPE Symmetric Mean Absolute Percentage Error

FLC Fuzzy Logic Controller

FLS Fuzzy Logic System

FOU Footprint of Uncertainty

GMRAE Geometric Mean Relative Absolute Error

GT2	General Type-2
IASC	Iterative Algorithm with Stopping Condition
IT2	Interval Type-2
KM	Karnik-Mendel
LMF	Lower Membership Function
LR	Linear Regression
LSE	Least-Squares Estimate
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
MF	Membership Function
MG	Mackey Glass
MRAE	Mean Relative Absolute Error
MSE	Mean Squared Error
NT	Nie-Tan
RAE	Relative Absolute Error
RMSE	Root-Mean-Square Error
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machines
T1	Type-1

T2 Type-2

TSK Takagi-Sugeno-Kang

UMBRAE Unscaled Mean Bounded Relative Absolute Error

UMF Upper Membership Function

Chapter 1

Introduction

This thesis is concerned with the exploration of novel techniques involved in the optimisation of fuzzy inference systems by the adaptive-network-based fuzzy inference system (ANFIS), and the comparisons between type-1 (T1) and interval type-2 (IT2) ANFIS. Specifically, in this thesis:

- An extended ANFIS architecture which can be used for both T1 and IT2 fuzzy models is introduced.
- A direct approach and its enhancement for determining the switch points in the Karnik-Mendel (KM) algorithm are proposed to improve the computational efficiency of IT2 ANFIS.
- To enable fair comparisons of different models, a new accuracy measure, which combines the best features of various alternative measures, based on bounded relative error is introduced.
- Based on the above, experimental comparisons are made on both synthetic and real-world data to investigate the performance of T1 and IT2 ANFIS.

In this chapter, the background and motivation behind this research are described first, and later the aims and objectives are introduced. The organisation of this thesis is also outlined followed by highlighted contributions to knowledge.

1.1 Background and Motivation

Since its introduction by Zadeh in 1965 [1], fuzzy logic has become more and more popular. Fuzzy logic is a theory based on the notion of a fuzzy set, which is simple yet powerful to represent uncertainty. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition that an element either belongs or does not belong to the set. In contrast, fuzzy set theory provides a gradual assessment of the membership of elements in a set with the aid of a membership function valued in the real interval $[0, 1]$. Basically, there are two types of commonly used fuzzy sets: T1 and T2 (type-2). The difference between T1 and T2 fuzzy sets will be introduced in the background chapter later. Due to the capability in dealing with uncertainty, fuzzy logic systems (FLS), also known as fuzzy inference systems (FIS), have been shown to have many advantages in modelling applications in many areas such as engineering, natural science, and time series forecasting [2–4].

At the early stage, a common concern for constructing a FIS was that there are no standard methods for transforming human knowledge or experience into the rule base [5]. Also, effective methods are required to tune the membership functions for optimising the performance of FIS. The architecture of ANFIS was proposed by Jang [5] to serve as a basis for constructing a set of fuzzy rules with appropriate membership functions to generate the desired FIS. A hybrid learning algorithm based on the gradient method and the least-squares estimate (LSE) is applied to identify parameters of membership functions. The original ANFIS was designed for T1 Takagi-Sugeno-Kang (TSK) fuzzy inference models.

Up until the beginning of this century, there was a heavy emphasis on T1 FLS. During the past few years, there has been a steady increase of interest in type-2 FLS, and in particular IT2 FLS [6]. An increasing interest in T2 FLS has also led to an increase in research on IT2 ANFIS. For example, early studies on T2 ANFIS can be found in [7, 8], who reported an approach that uses an adaptive network to learn a T2 fuzzy system based on linguistic inputs and numeric output. Other examples of research on T2 ANFIS, based on crisp inputs, can be found in [9–14]. As an extension of the studies on ANFIS

based on singleton fuzzification, studies on non-singleton fuzzification based IT2 ANFIS can be found in [12, 15, 16].

Though some studies have been made on IT2 ANFIS, a clear architecture for IT2 ANFIS has not been presented in the literature compared to that of T1 ANFIS presented in [5]. Also, it has been indicated in Mendel [17] that least-squares based methods cannot be applied to a T2 FLS. It is because the *Fuzzy Basis Function* expansion for a T2 TSK FLS, which is the starting point for the LSE method, cannot be obtained without knowing the consequent parameters in advance. This issue was also mentioned in [10]. More investigations on this will be provided in Chapter 5, where an extended ANFIS architecture for both T1 and IT2 models is proposed.

It has been demonstrated that in principle T2 FLS can provide better (and certainly no worse) performance than T1 FLS [18]. This can be guaranteed by initialising the membership functions of an IT2 FLS with optimised membership functions of a T1 FLS. Thus, theoretically, IT2 ANFIS is expected to be better than T1 ANFIS as it has more model parameters. However, it is still an open question that whether IT2 ANFIS can always in practice provide better performance than T1 ANFIS, especially for the cases when both T1 and IT2 ANFIS models are optimised from scratch. To investigate this, comparisons of T1 and IT2 ANFIS are made in Chapter 5, and further explored in Chapter 6.

Forecasting has always been an attractive research area since it plays an important role in daily life. As one of the most popular research domains, time series forecasting has received particular interest from researchers [19–23]. Many studies have been conducted with the aim of identifying the most accurate methods for time series forecasting [24–28]. However, research findings indicate that the performance of forecasting methods varies according to the accuracy measure being used [29]. Various accuracy measures have been proposed as the best to use in the past decades [30]. However, many of these measures are not generally applicable due to issues such as being infinite or undefined under certain circumstances, which may produce misleading results. Extensive comparisons of commonly used measures will be made in Chapter 3.

Besides accuracy, computational efficiency is also an important factor considered while choosing a model. The KM algorithm [31] is an iterative procedure widely used in the fuzzy logic theory to compute the centroid of an IT2 fuzzy set. Specifically, the KM algorithm is used to determine the switch points needed for the lower and upper bounds of the centroid. Such process is also a key step to generate the output of ANFIS. Though the KM algorithm is known to converge monotonically and superexponentially fast, methods to reduce their computational cost are desired. Many enhanced algorithms have been proposed to improve the computational efficiency of the KM algorithm [32–35]. However, all of these algorithms are still based on iterative procedures. In Chapter 4, a direct approach for determining the switch points in the KM algorithm will be introduced.

1.2 Aims and Objectives

The aim of this thesis is to introduce a novel framework for implementing and evaluating T1 and IT2 ANFIS models. The remainder of this section discusses the objectives that must be achieved to attain the aim.

- *Make it easy to implement T1 and IT2 ANFIS models.*

To fulfil the aim, a fundamental requirement is the capability to implement T1 and IT2 ANFIS models. As mentioned in the background above, a clear architecture for IT2 ANFIS has not been presented in the literature. This somehow has been an obstacle to the research of IT2 ANFIS and its application to real-world problems. Given that the architecture of T1 ANFIS has been explicitly introduced in [5], one objective of this research is to extend this ANFIS architecture so that it can be used for both T1 and IT2 models.

- *Reduce the computational costs of IT2 ANFIS.*

With the same number of rules and membership functions, IT2 models are often more computationally expensive than T1 models. This is, in fact, a crucial obstacle to the use of IT2 systems. Thus, it would be quite useful to reduce the

computational costs for IT2 ANFIS. Note that a bottle-neck for IT2 ANFIS is to aggregate the output of each rule by the KM algorithm. Hence, our objective is to improve the efficiency of such aggregating process by enhancing the KM algorithm. Given that the KM algorithm and its current enhancements in the literature are all iterative algorithms, a possible direction is to explore a direct approach to determine the switch points in the KM algorithm without iterations.

- *Enable fair comparison for T1 and IT2 ANFIS models.*

Without a suitable measure, it is not possible to carry out an objective evaluation of methodologies. It is well known that many of existing measures suffer from one or more issues such as poor resistance to outliers and scale dependence. To make fair comparisons, it is necessary to have an accuracy measure which does not suffer from the common issues of existing accuracy measures. Hence, one of the objectives of this thesis is to propose a new accuracy measure which combines the best features of alternative measures without suffering their common issues. Specifically, the new accuracy measure is expected to have the following properties: (i) Informative: it can provide an informative result without the need to trim errors; (ii) Resistant to outliers: it can hardly be dominated by a single forecasting outlier; (iii) Symmetric: overestimates and underestimates are treated equally; (iv) Scale-independent: it can be applied to datasets on different scales; (v) Interpretability: it is easy to understand and can provide intuitive results.

- *Evaluate T1 and IT2 ANFIS models.*

It has been demonstrated in [18] that in principle IT2 FLS, with the same number of fuzzy rules and membership functions, can potentially provide better (and certainly no worse) performance than T1 FLS. This can be guaranteed by initialising the Type-2 membership functions with the optimised T1 membership function. However, it is still an open question that whether IT2 ANFIS can provide significantly better performance than T1 ANFIS in practice. Thus, another objective

of this thesis is to investigate the properties of T1 and IT2 ANFIS by making comparisons to answer the above question.

1.3 Contributions to Knowledge

This research has contributed to two journal papers, two peer-reviewed conference papers and one book chapter. These publications are listed below.

- [1] C. Chen, J. Twycross, and J. M. Garibaldi, “A new accuracy measure based on bounded relative error for time series forecasting,” *PLOS ONE*, vol. 12, no. 3, pp. 1–23, 2017.
- [2] C. Chen, R. John, J. Twycross, and J. Garibaldi, “A direct approach for determining the switch points in the Karnik-Mendel algorithm,” *IEEE Transactions on Fuzzy Systems*, vol. PP, no. 99, p. 1, 2017.
- [3] C. Chen, R. John, J. Twycross, and J. M. Garibaldi, “An extended ANFIS architecture and its learning properties for type-1 and interval type-2 models,” in *Proceedings IEEE International Conference on Fuzzy Systems*, 2016, pp. 602–609.
- [4] C. Chen, R. John, J. Twycross, and J. M. Garibaldi, “Type-1 and interval type-2 ANFIS: a comparison,” in *Proceedings IEEE International Conference on Fuzzy Systems*, 2017, pp. 1–6.
- [5] T. A. Runkler, S. Coupland, R. John, and C. Chen, “Interval type-2 defuzzification using uncertainty weights,” in *Frontiers in Computational Intelligence*, S. Mostaghim, A. Nürnberger, and C. Borgelt, Eds. Springer International Publishing, 2017.

1.4 Organisation of the thesis

In this Chapter, we have provided an overview of the thesis. Specifically, a brief review of background information was first given to identify research gaps which motivated the

aims and objectives of this work. Then, a list of publications was given to indicate the contributions to knowledge.

The remainder of this thesis is organised as follows:

- Chapter 2 provides an overview of background material and existing literature that are used and referred to in this thesis.
- Chapter 3 proposes a new accuracy measure as an alternative to commonly used measures which are not generally applicable for fair comparisons due to issues such as being infinite or undefined under certain circumstances. The new accuracy measure is able to combine the best features of various alternative measures without suffering from their issues.
- Chapter 4 presents a direct approach for determining the switch points in the KM algorithm. The objective of this chapter is to reduce the computational cost of IT2 ANFIS.
- Chapter 5 introduces an extended novel ANFIS architecture which makes it easy to implement T1 and IT2 ANFIS models. By making comparisons, this chapter also examines the question that whether the iterative process of estimating the consequent parameters by the LSE method can be successful for IT2 ANFIS.
- Chapter 6 further compares the performance of T1 and IT2 ANFIS based on both synthetic and real-world datasets. Results are given based on different accuracy measures including the one proposed in Chapter 3 to provide more insights.
- Chapter 7 concludes the thesis by summarising the main points and outcomes. Also, this chapter presents a discussion about limitations and future work.

Chapter 2

Background

2.1 Introduction

In this chapter, background material and a literature survey of research subjects on which this thesis is based are provided. The background of fuzzy sets and systems is firstly introduced in Section 2.2. It includes some basic definitions of fuzzy sets and the set-theoretical operations for fuzzy sets. Section 2.2.2 gives a brief review of fuzzy inference systems including an introduction of the key components of a typical fuzzy inference system. Section 2.2.3 focuses on ANFIS which is a commonly used architecture for optimising a fuzzy inference system. As it has been mentioned above that suitable accuracy measures are very important to conduct objective comparisons of different fuzzy models. Hence, in Section 2.3, a review of commonly used accuracy measures in the literature is made focusing on their advantages and disadvantages. Specifically, accuracy measures are summarised into three categories: scale-dependent measure, percentage-based measures, and relative based measures. A special review of the symmetric mean absolute percentage error (sMAPE) is made since sMAPE was used as a major measure in the M3-competition [29], but later it was widely criticised for its asymmetry which is contradict to its name.

2.2 Fuzzy Sets and Systems

Fuzzy sets were originally introduced by Zadeh [1] to handle the ambiguity inherent in assigning grades of membership to the elements of real-world sets such as the set of *young* men or the set of *small* cars. They were specifically designed to represent vagueness and uncertainty to deal with imprecision in real-world problems.

T1 fuzzy sets are standard fuzzy sets whose membership grades are crisp numbers from 0 to 1. Although many complex problems can be expressed by using just T1 fuzzy sets, there remain limitations. As it was pointed out by many researchers[see, for example, 36], T1 fuzzy sets may seem problematical since there is, in fact, no fuzziness in the standard T1 membership grade.

To address this issue, Zadeh [37] proposed the concept of type- n ($n = 2, 3, \dots$) fuzzy sets whose membership grades themselves are type- $(n-1)$ fuzzy sets. They are very useful in circumstances where it is difficult to determine an exact membership function (MF) for fuzzy sets. The use of T2 fuzzy sets was therefore advocated by many researchers [e.g. 38, 39]. However, T2 fuzzy sets are more difficult to use and understand than T1 fuzzy sets. Also, due to the significant computational requirements, their use in practice has been limited.

More recently, there has been a renewed interest in T2 fuzzy sets, mainly due to the efforts of Mendel and John [40], but also due to the significant increase of computational power. In [40], Mendel and John established a set of terms to help with the understanding and the use of T2 fuzzy sets. In particular, they introduced a concept known as *footprint of uncertainty* (FOU) which enables the visualisation of T2 fuzzy sets in a 2D sketch.

Particular interests have been paid by researchers to a specific class of T2 fuzzy sets called IT2 fuzzy sets. The secondary membership grades of an IT2 fuzzy set are all restricted to have the value 1. Thus, the calculations of IT2 fuzzy sets are easy to perform, and the computational costs have been significantly reduced.

2.2.1 Basic Definitions

By providing a brief review of fuzzy sets and fuzzy inference systems, this section lets us establish common notations and definitions for fuzzy sets and systems.

Fuzzy Sets

Definition 2.2.1. A *type-1 (T1) fuzzy set* A can be represented by a set of real-valued pairs as

$$A = \{(x, \mu_A(x)) \mid \forall x \in X\} \quad (2.1)$$

where x is a variable in the domain X (also called the *universe of discourse* of A), $\mu_A(x)$ is a *membership function* $\mu_A : X \rightarrow [0, 1]$, and the value of $\mu_A(x)$ is the *degree of membership*, or *membership grade*, of x in A .

When the domain X is continuous, the T1 fuzzy set A is also commonly expressed as

$$A = \int_{x \in X} \mu_A(x) / x \quad (2.2)$$

where the symbol $\int$ in this equation does not denote integration; instead, it denotes the set (or collection) of all $x \in X$ with the associated membership grade $\mu_A(x)$. When X is discrete, the fuzzy set A is then often represented by

$$A = \sum_{i=1}^N \mu_A(x_i) / x_i \quad (2.3)$$

where x_i is the i^{th} of N discrete points in the universe of discourse X . In this equation, the symbol $\sum$ does not denote summation; it denotes the set of all $x_i \in X$ with the associated membership grade $\mu_A(x_i)$.

For simplicity, the T1 fuzzy set A and its membership function $\mu_A(x)$ are often considered to be synonyms, that is,

$$A \Leftrightarrow \mu_A(x). \quad (2.4)$$

Example 2.2.1. Examples of two T1 fuzzy sets are presented in Fig. 2.1.

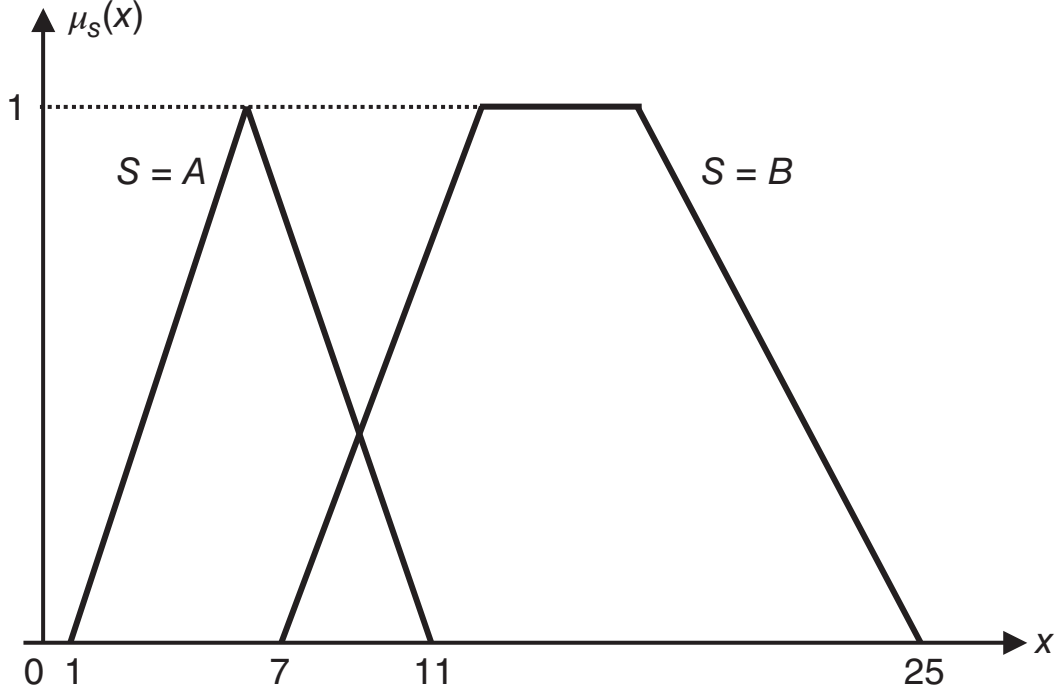


Fig. 2.1 **Examples of two T1 fuzzy sets, A (triangle) and B (trapezoidal).** This figure is taken from [6].

Definition 2.2.2. A *general type-2 (GT2) fuzzy set* $\tilde{A}$ can be considered as an extension of a T1 fuzzy set. It can be represented by a set of pairs as

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid \forall x \in X\} \quad (2.5)$$

where x is the *primary variable* of $\tilde{A}$ in the domain X . Unlike the real-valued membership grade in a T1 fuzzy set, the membership grade $\mu_{\tilde{A}}(x)$ of any x in the GT2 fuzzy set $\tilde{A}$ is a T1 fuzzy set A_x (also called a *vertical-slice* [17, 18]), that is,

$$\mu_{\tilde{A}}(x) \Leftrightarrow A_x = \{(u, \mu_{A_x}(u)) \mid \forall u \in J_x\} \quad (2.6)$$

where u is the *secondary variable* of $\tilde{A}$ in the domain $J_x \subseteq U = [0, 1]$ for any given x . J_x is also called the *primary membership grade(s)* of x in $\tilde{A}$. $\mu_{A_x}(u)$ is the *secondary*

membership function $\mu_{A_x} : J_x \rightarrow [0, 1]$ and the value of $\mu_{A_x}(u)$ is the *secondary membership grade*. Note that $\mu_{A_x}(u)$ is the same as notation $\mu_{\tilde{A}}(x, u)$ for any given value of x . $\mu_{\tilde{A}}(x, u)$ is normally used for 3D representations (see Fig. 2.2 for example).

When the domain X is continuous, the GT2 fuzzy set $\tilde{A}$ is also commonly written as

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x \quad (2.7)$$

where the symbol $\int$ in this equation does not denote integration; instead, it denotes the set (or collection) of all $x \in X$ with the associated secondary membership function $\mu_{\tilde{A}}(x)$. When X is discrete, $\tilde{A}$ is then often expressed as

$$\tilde{A} = \sum_{i=1}^N \mu_{\tilde{A}}(x_i)/x_i \quad (2.8)$$

where x_i is the i^{th} of N discrete points in the universe of discourse X . In this equation, the symbol $\sum$ does not denote summation; it denotes the set of all $x_i \in X$ with the associated secondary membership function $\mu_{\tilde{A}}(x_i)$.

There are also other representations for GT2 fuzzy sets. For example, two well-known ones are the α -Plane representation [41] and the zSlice representation [42]. Note that these two representations are essentially the same as the concept of level-sets initiated by Zadeh [37]. More recently, Hamrawi et al. [43] introduced an alpha-cut decomposition theorem for T2 fuzzy sets, which is a new representation analogous to the alpha-cut representation of T1 fuzzy sets and the extension principle. Details about these are not investigated here since it is beyond the scope of this thesis.

Definition 2.2.3. The *FOU* of a GT2 fuzzy set $\tilde{A}$ is the area covered by all the primary membership grades on the xu -plane.

In practice, the FOU of a GT2 fuzzy set is commonly characterised by upper and lower membership functions (see Fig. 2.3).

Definition 2.2.4. The *upper membership function (UMF)* $\bar{\mu}_{\tilde{A}}(x)$ and *lower membership function (LMF)* $\underline{\mu}_{\tilde{A}}(x)$ are two T1 fuzzy sets which respectively define the upper and

lower boundaries of the FOU of a GT2 fuzzy set $\tilde{A}$ in the domain X , where $x \in X$. $\bar{\mu}_{\tilde{A}}(x)$ and $\underline{\mu}_{\tilde{A}}(x)$ can be characterised as

$$\bar{\mu}_{\tilde{A}}(x) = \{(x, \max(J_x)) \mid \forall x \in X\} \quad (2.9)$$

$$\underline{\mu}_{\tilde{A}}(x) = \{(x, \min(J_x)) \mid \forall x \in X\} \quad (2.10)$$

where J_x denotes the *primary membership grade(s)* of $\tilde{A}$ at any specified x .

Definition 2.2.5. A GT2 fuzzy set $\tilde{A}$ is called an *interval type-2 (IT2) fuzzy set* when all the secondary membership grades are equal to 1, that is, $\mu_{A_x}(u) = 1$ for all $x \in X$ and $u \in J_x$. Since the secondary membership grades of an IT2 fuzzy set convey no new information, FOU can be considered as a complete description of an IT2 fuzzy set [6]. Hence, such IT2 fuzzy set $\tilde{A}$ can be expressed as

$$\tilde{A} = \{(x, [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]) \mid \forall x \in X\} \quad (2.11)$$

or simplified as

$$\tilde{A} \Leftrightarrow [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]. \quad (2.12)$$

Example 2.2.2. A 3D example of an IT2 fuzzy set is shown in Fig. 2.2.

Example 2.2.3. A 2D example of an IT2 fuzzy set is presented in Fig. 2.3.

Set-Theoretical Operations

Definition 2.2.6. The *union of two T1 fuzzy sets* A and B is still a T1 fuzzy set which can be expressed as

$$A \cup B = \{(x, \mu_{A \cup B}(x)) \mid \forall x \in X\} \quad (2.13)$$

where

$$\mu_{A \cup B}(x) = \perp(\mu_A(x), \mu_B(x)) \quad (2.14)$$

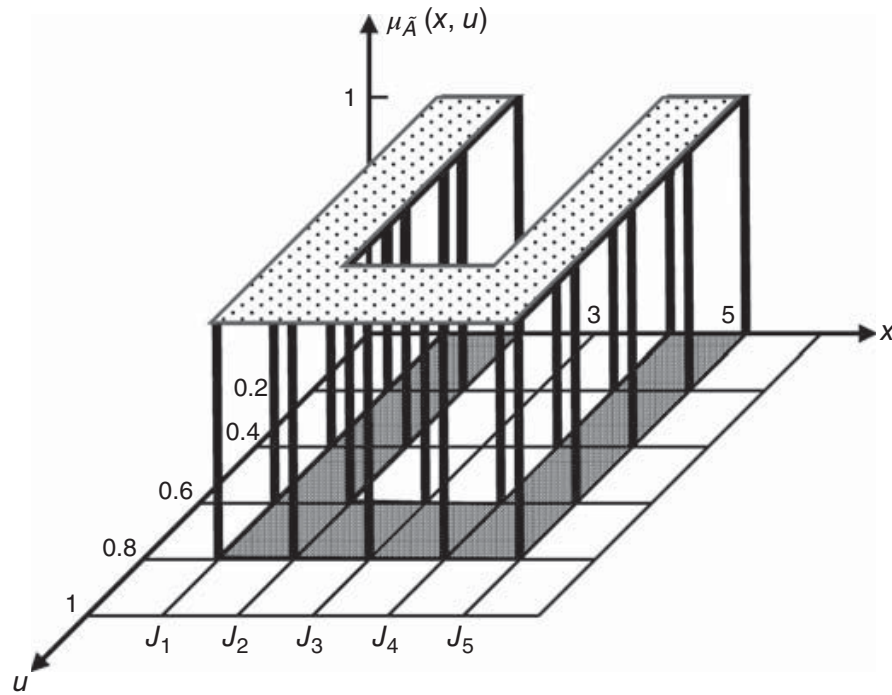


Fig. 2.2 A 3D example of an IT2 fuzzy set. The shaded area on the xu -plane is the FOU of this example IT2 fuzzy set. Note that J_x is the primary membership grades of x in $\tilde{A}$. For example, J_1 is an interval. This figure is taken from [44].

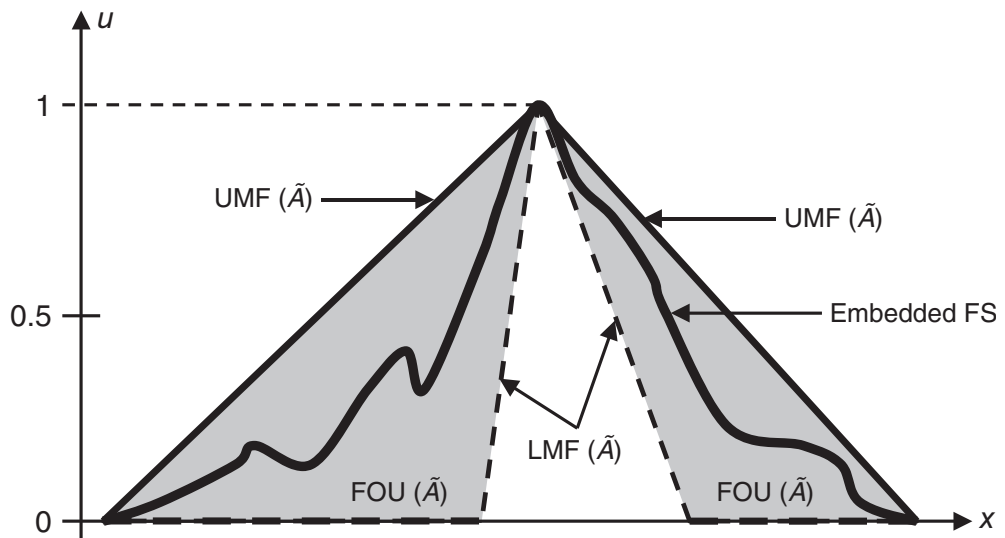


Fig. 2.3 A 2D example of an IT2 fuzzy set on the xu -plane. The shaded region is the FOU. UMF (solid line) and LMF (dotted line) are the upper and lower boundaries of the FOU respectively. This figure is taken from [44].

and $\perp$ is called *t-conorm* which is a function $\perp : [0, 1] \times [0, 1] \rightarrow [0, 1]$. Normally, the *t-conorm* function for fuzzy sets is *maximum*.

Definition 2.2.7. The *intersection of two T1 fuzzy sets* A and B is still a T1 fuzzy set which can be expressed as

$$A \cap B = \{(x, \mu_{A \cap B}(x)) \mid \forall x \in X\} \quad (2.15)$$

where

$$\mu_{A \cap B}(x) = \top (\mu_A(x), \mu_B(x)) \quad (2.16)$$

and $\top$ is called *t-norm* which is a function $\top : [0, 1] \times [0, 1] \rightarrow [0, 1]$. Normally, the *t-norm* function for fuzzy sets is *minimum* or *product*. In the literature, the *t-norm* operator is also commonly noted as $\star$.

Example 2.2.4. The union and intersection of two T1 fuzzy sets A and B are illustrated in Fig. 2.4.

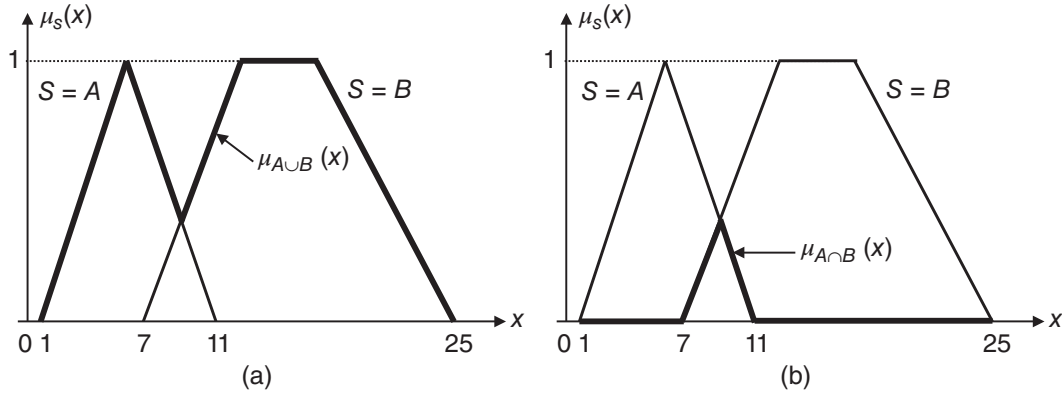


Fig. 2.4 An illustration of (a) union and (b) intersection of two T1 fuzzy sets **A** and **B**. The t-norm and t-conorm for the union and intersection here are based on the *maximum* and *minimum* respectively. Figures are taken from [6].

Definition 2.2.8. The *complement of a T1 fuzzy sets* A can be expressed as

$$\neg A = \{(x, \neg \mu_A(x)) \mid \forall x \in X\} \quad (2.17)$$

where

$$\neg \mu_A(x) = 1 - \mu_A(x). \quad (2.18)$$

There are different approaches (e.g. the extension principle [37]) to do the union, intersection and complement operations for IT2 fuzzy sets [6]. In this thesis, these set-theoretical operations for IT2 fuzzy sets are derived from such definitions above for T1 fuzzy sets.

Definition 2.2.9. The *union of two IT2 fuzzy sets* $\tilde{A}$ and $\tilde{B}$ is still an IT2 fuzzy set which can be characterised as

$$\tilde{A} \cup \tilde{B} \Leftrightarrow \left[\perp \left(\underline{\mu}_{\tilde{A}}(x), \underline{\mu}_{\tilde{B}}(x) \right), \perp \left(\bar{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{B}}(x) \right) \right]. \quad (2.19)$$

Definition 2.2.10. The *intersection of two IT2 fuzzy sets* $\tilde{A}$ and $\tilde{B}$ is still an IT2 fuzzy set which can be characterised as

$$\tilde{A} \cap \tilde{B} \Leftrightarrow \left[\top \left(\underline{\mu}_{\tilde{A}}(x), \underline{\mu}_{\tilde{B}}(x) \right), \top \left(\bar{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{B}}(x) \right) \right]. \quad (2.20)$$

Definition 2.2.11. The *complement of an IT2 fuzzy set* $\tilde{A}$ can be characterised as

$$\neg \tilde{A} \Leftrightarrow \left[\left(1 - \underline{\mu}_A(x) \right), \left(1 - \bar{\mu}_A(x) \right) \right]. \quad (2.21)$$

2.2.2 Fuzzy Inference System

A fuzzy inference system (FIS) is also known as a fuzzy logic system (FLS), or in some cases a fuzzy logic controller (FLC). It is a model that uses the fuzzy set theory to maps inputs into outputs (e.g. from features to classes in the case of classification). Due to its capability in dealing with uncertainty, FIS has been shown to have many advantages in modelling applications in many areas including engineering, natural science, and time series forecasting [2–4].

Based on T1 and T2 fuzzy sets, the FIS can also be classified as T1 and T2. A T1 FIS is a model which only involves T1 fuzzy sets. If there is any T2 fuzzy set involved

in a FIS model, it is a T2 FIS. Up until the beginning of this century, there was a heavy emphasis on T1 FIS. During the past few years, there has been a steady increase of interest in T2 FIS, especially in IT2 FIS [6], and a number of works have helped to make the implementation of T2 FIS straightforward [44, 45].

An important role of FIS is to mimic the process followed by the human mind when performing control actions [6]. For example (adapted from [6]), when a person drives (controls) a car, it is unlikely for him or her to think:

*If the **temperature** is 10 degrees Celsius and the **rainfall** is 70.5 mm and the **road** is 40% slippery and the **distance** between my car and the car in front of me is 3 meters, **then** I will depress the **acceleration** pedal only 10%.*

Instead, it is much more likely for him or her to think:

*If it (the **temperature**) is cold and the **rainfall** is heavy and the **road** is a bit slippery and the **distance** between my car and the car in front of me is quite close, **then** I will depress the **acceleration** pedal slightly.*

In a human-driven control process, the cycle begins with a person converting a physical quantity (e.g. distance) from numbers into words or perceptions (e.g. quite close). Then such words or perceptions as inputs, will trigger the output actions by words (e.g., depress the acceleration pedal slightly) based on the person's knowledge base. Finally, the person executes a particular action (e.g., depress the acceleration pedal only 10%).

With the aim of mimicking human control actions, the general structure of a T1 FIS is shown in Fig. 2.5. As can be observed from this figure, a typical T1 FIS is comprised of four main components: rules, fuzzifier, inference engine and defuzzifier. The structure of a T2 FIS, as shown in Fig. 2.6, is very similar to the T1 FIS with an extra component named type-reducer. Details of these components are described below.

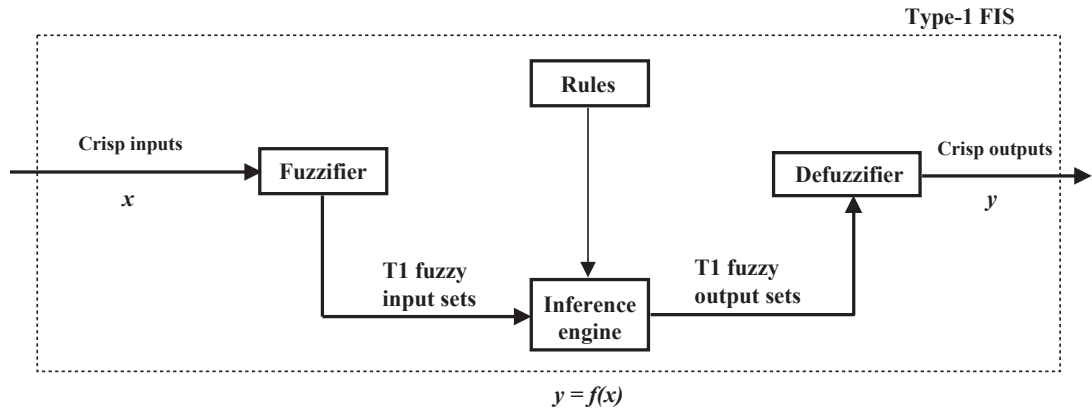


Fig. 2.5 **General structure of a T1 FIS.** The heavy lines with arrows indicate the path taken by signals during the actual operation of the FIS. Rules are predefined and activated by the inference engine during the actual operation of the FIS. This figure is adapted from [6].

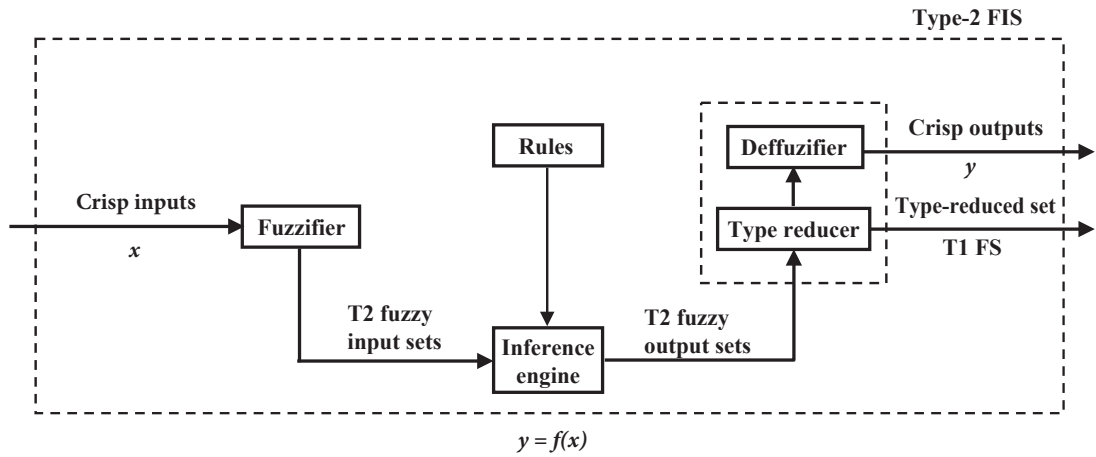


Fig. 2.6 **General structure of a T2 FIS.** This figure is adapted from [6].

Rules

A FIS is also called a rule-based FIS. The rule base, a set of fuzzy rules, is the core part of a FIS. The fuzzy rules are based on an if-then structure. For example, *If Temperature is Low and Pressure is High, then Fan Speed is Medium*. Or, more mathematically, the above rule can be expressed as

IF x_1 is A_1 **AND** x_2 is A_2 , **THEN** y is C_1

where x_1 and x_2 are inputs representing *Temperature* and *Pressure* respectively, and the output y represents *Fan Speed*. A_1 , A_2 and C_1 are membership functions representing *Low*, *High* and *Medium* respectively. The IF part of a rule is called antecedent, and the THEN part of a rule is called consequent.

There are two types of commonly used fuzzy rules. One attributed to Zadeh [46] where both antecedents and consequents are fuzzy sets. The first successful form of FIS based on this type of rules was introduced by Mamdani and Assilian [47] in 1975. Hence, a FIS based on this type of rules is also called a Mamdani type FIS. Another type of rules, which was known as the TSK type, was introduced by Takagi and Sugeno [48] and Sugeno and Kang [49]. In the TSK rules, the antecedents are also fuzzy sets, but the consequents are linear functions of the crisp inputs. For example, for the rule above, C_1 can be a first-order linear function f_1 of x_1 and x_2 , which can be expressed as

$$f_1 = p_1x_1 + q_1x_2 + r_1 \quad (2.22)$$

where p_1 , q_1 , and r_1 are constant parameters.

Fuzzifier

In a real-world system, the inputs are normally crisp numbers. A fuzzifier maps a crisp number into a fuzzy set, which then can be used by the inference engine. Such process of mapping a crisp number into a fuzzy set is called fuzzification. In general, there are two types of fuzzifiers: singleton and non-singleton [50]. Supposing we have a crisp number x' , a singleton fuzzifier maps x' into a fuzzy set (normally T1) which only has non-zero membership grades at x' . In contrast, a non-singleton fuzzifier maps the crisp number x' into a fuzzy set (either T1 or T2) which also has non-zero membership grades at points other than x' . While singleton fuzzification based FISs are the most common types of FISs, non-singleton fuzzification based FISs [51] were specifically designed for handling the uncertainties associated with the inputs to a FIS. It has been shown in some research that non-singleton fuzzification based FISs were able to provide better performance in many cases [4, 52].

Inference Engine

In the fuzzy inference engine, fuzzy logic principles (e.g. set-theoretical operations) are used to map input fuzzy sets (obtained from fuzzifier), based on the IF-THEN rule base, into output fuzzy sets. Details about this process can be found in Section 3.4 in [50].

Defuzzifier

The defuzzifier produces a crisp output based on the output fuzzy sets from the inference engine for a T1 FIS, or from the type reducer for a T2 FIS. The process in a type reducer is a mapping from one or more fuzzy sets into a real number. Such process is called defuzzification. Many types of defuzzifiers have been proposed in the literature. An important criterion for the choice of a defuzzifier is the computational simplicity. Based on this criterion, some well-known candidate defuzzifiers for a Mamdani type FIS are maximum, mean-of-maxima, centroid, centre-of-sums, height, modified height, and centre-of-sets [50]. In this thesis, we only talk about the centroid which is the most commonly used defuzzification method. Supposing the output of a FIS from the inference engine is T1 fuzzy set C which is equivalent to $\mu_C(y)$ where y is the variable in the domain Y . The centroid of C is defined as

$$c = \frac{\sum_{i=1}^N y_i \mu_C(y_i)}{\sum_{i=1}^N \mu_C(y_i)} \quad (2.23)$$

where y_i is the i^{th} of N discrete points in the domain Y .

The defuzzification method for a TSK type FIS is very similar to the centroid defuzzifier above. As mentioned in [50], there are two types of defuzzifiers for the TSK models: un-normalised and normalised. The un-normalised defuzzifier can be expressed as

$$y = \sum_{i=1}^M w_i f_i \quad (2.24)$$

where y is the defuzzified output, M is the number of rules, w_i is the firing strength of the i^{th} rule, and f_i is the consequent linear function of the i^{th} rule. The normalised

defuzzifier for a TSK FIS can be expressed as

$$y = \frac{\sum_{i=1}^M w_i f_i}{\sum_{i=1}^M w_i} \quad (2.25)$$

Type Reducer

A type reducer in Fig. 2.6 is used to map a T2 (IT2 in this thesis) fuzzy set into a T1 fuzzy set, which can then be defuzzified to a crisp output. Such process is called type reduction. During the past few years, type-reduction has been one of the most active research topics for T2 fuzzy sets and systems. In the literature, many type reduction methods have been proposed for IT2 fuzzy sets [50].

The Nie-Tan (NT) method [53] is one of the most well-known and easy to apply methods for IT2 fuzzy sets. After type reduction for a given IT2 fuzzy set $\tilde{A}$, which is represented by

$$\tilde{A} \Leftrightarrow [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)] \quad (2.26)$$

the reduced output is a T1 fuzzy set A which has the same domain as $\tilde{A}$. The membership grade of A at x is defined as

$$\mu_A(x) = \frac{\underline{\mu}_{\tilde{A}}(x) + \bar{\mu}_{\tilde{A}}(x)}{2} \quad (2.27)$$

Many other type reduction methods are based on the centroids of embedded T1 fuzzy sets of an IT2 fuzzy set, where an example of embedded T1 fuzzy set is the wavy line illustrated in Fig. 2.3. An exhaustive such type reduction method [54] is to discretised an IT2 fuzzy set and to calculate the centroids of all embedded T1 fuzzy sets. However, the exhaustive method is usually computationally expensive and hence impractical. Alternative approximations of the exhaustive method are proposed. For example, the sampling method, which uses some randomly selected samples of the embedded T1 fuzzy sets, is an efficient cut-down alternative to the exhaustive method [55]. It has been recently proved that the sampling method computes the accurate centre of gravity of an IT2 fuzzy set when the number of sampling is infinity [56].

Another well-known and probably the most commonly used type reduction method based on centroids is the KM algorithm, or its enhancement. The KM algorithm only calculates the left and right bounds of the centroids for all possible embedded T1 fuzzy sets. A more detailed review is given below.

Karnik-Mendel Algorithm

The KM algorithm was the first algorithm proposed to determine the switch points¹ when computing the centroids of IT2 fuzzy sets [31]. With its fast convergence, it is still the most widely used algorithm for computing the switch points [57, 6, 58–60]. It was originally shown that the maximum number of iterations of the KM algorithm is N , which is the number of discrete points in the universe of discourse for an IT2 fuzzy set. This was believed to be extremely conservative, and it was subsequently proved by Liu [61] that the maximum number of iterations of the KM algorithm is $(N + 1)/2$. A much smaller number $(N + 2)/4$ was given in [62] as the maximum number of iterations, without proof. Though these numbers are much smaller than N , they are still believed to be conservative [6]. Many studies by simulations show that the KM algorithm converges in from two to six iterations, regardless of N [34].

Many attempts have been made to improve the efficiency of the KM algorithm. The enhanced KM (EKM) algorithm introduces a better initialisation, which “on average ... can save about two iterations” [34]. Also, some algorithmic improvements to simplify computations were introduced to reduce the computational cost for each iteration. An improved iterative algorithm with stopping condition (IASC) was first proposed in [32] and further refined in [33]. Further improvements to the IASC have been made in the enhanced IASC (EIASC) in [35]. Both the IASC and EIASC algorithms have been reported to be superior to the KM and EKM algorithms when N is small (e.g. $N \leq 100$). However, their computational costs increase rapidly as N increases since many possible switch points have to be evaluated before finding the correct ones [6].

¹The mathematical explanation about the KM algorithm and the switch points will be provided in Section 4.2.

Rather than calculating the exact centroids, closed-form solutions, which are much more efficient than iterative algorithms, have been provided for approximations. For example, it has been demonstrated that the NT method can give a very good approximation to the KM algorithm [53, 56]. As an extension of the NT method, a better approximation with Taylor-series is provided in [63]. Closed-form formulae for calculating the centroids of a general T2 fuzzy set are proposed in [64], where linear connections between the centroid endpoints of any α plane and that of $\alpha = 0$ and $\alpha = 1$ planes have been introduced. However, the calculations of the centroid endpoints of the $\alpha = 0$ and $\alpha = 1$ planes are still based on the iterative KM algorithm. Other algorithms to improve efficiency have also been proposed recently, such as [65] and [66], although these too are iterative.

To the best of our knowledge, all existing algorithms, such as the EKM and the EIASC, for determining the switch points are still iterative-based methods. Since type reduction is considered to be a bottleneck of a FIS, it is still a challenge to improve the efficiency of these algorithms. For example, it would be interesting to introduce a direct approach which can compute the switch points without multiple iterations. We use the term ‘iteration’ here to mean the repeated calculation of the position of the switch points. Within all approaches, there are common calculations that require looping (e.g. cumulative sum), which we do not consider as iterations of the main algorithm.

2.2.3 The Adaptive Network-based Fuzzy Inference System

A common concern for constructing a FIS was that there were no standard methods for transforming human knowledge or experience into the rule base. Also, effective methods are required to tune the membership functions for optimising the performance of FISs. The architecture of *Adaptive-Network-based Fuzzy Inference System* (ANFIS) was proposed by Jang [5] in 1993 to serve as a basis for constructing a set of fuzzy rules with appropriate membership functions to generate the desired FIS to fit the training data. Note that there are also some other algorithms proposed for optimising FISs, such as simulated annealing [67–70], particle swarm optimisation [71–73] and quantum particle

swarm optimisation [74–76]. Many of these have already been applied to IT2 FISs. Compared to these algorithms, T1 ANFIS has been shown to be very competitive. In many studies, ANFIS has been shown to be superior to models using other techniques such as autoregression, genetic algorithms and artificial neural networks [77–79]. In this thesis, we focus on ANFIS because its competitive capability in optimising FIS. Also, though there have been some studies on IT2 ANFIS, its properties have not been fully investigated.

ANFIS Architecture

In the original paper [5] of ANFIS, three types of fuzzy reasoning (inferencing) mechanisms, as shown in Fig. 2.7, were discussed. Among these inferencing mechanisms, the first two can be considered as the Mamdani type, and the third is the TSK type. For simplicity, the original ANFIS architecture was designed based on the TSK type reasoning where the antecedents are T1 fuzzy sets, and the consequents are first order linear functions of the crisp inputs.

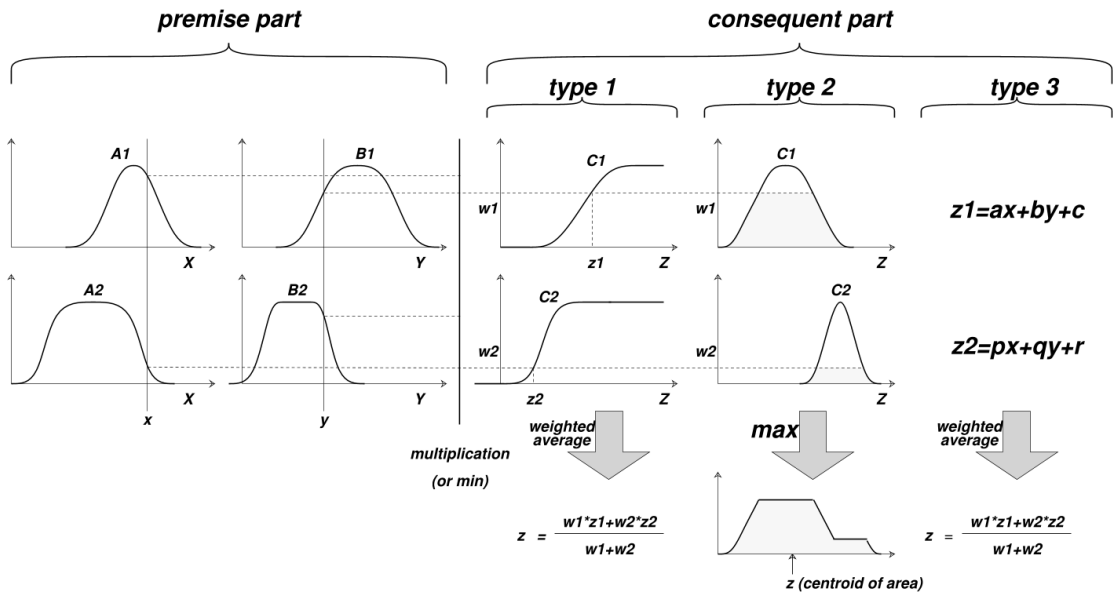


Fig. 2.7 Three types of fuzzy reasoning mechanisms in Fig. 2.7. This figure is taken from [5].

Fig. 2.8 shows an example structure of the original ANFIS for a FIS model which has two inputs x and y and one output f . It has two TSK type fuzzy rules, which are

IF x is A_1 **AND** y is B_1 , **THEN** $f_1 = p_1x + q_1y + r_1$

IF x is A_2 **AND** y is B_2 , **THEN** $f_2 = p_2x + q_2y + r_2$

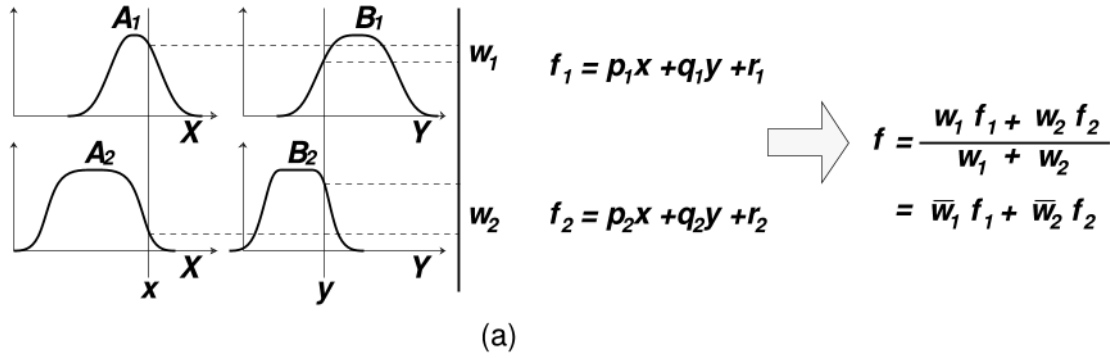
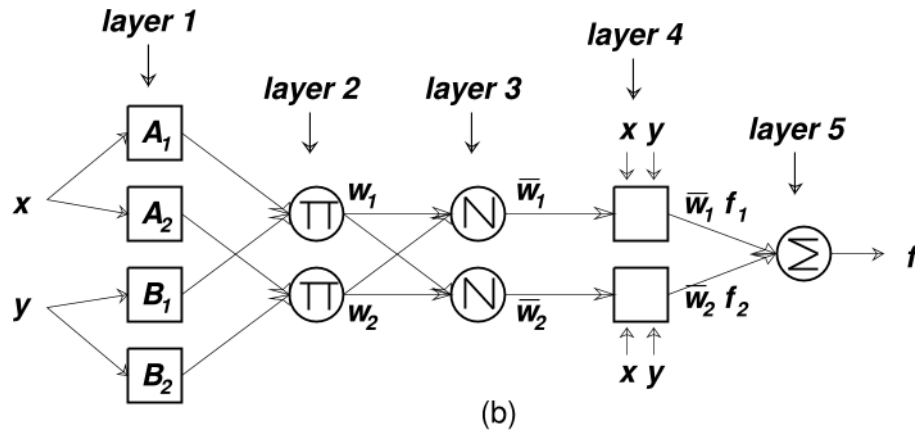


Fig. 2.8 **The typical structure of ANFIS.** In this model, there are two inputs and two TSK type fuzzy inference rules. For each input, there are two antecedent membership functions. This figure is taken from [5].

The original ANFIS is a five-layer network, where nodes in the same layer have the same function. Based on the structure in Fig. 2.8, we summarise the functions of nodes in each layer below:

Layer1: Every square node i in this layer has a node function like

$$O_i^1 = \mu_{A_i}(x)$$

, where the variable x denotes the crisp input to node i and A_i is antecedent fuzzy set with membership function $\mu_{A_i}(x)$. Supposing x' is a specific value of x , O_i^1 , which is then $\mu_{A_i}(x')$, indicates how x' satisfy the rule description ' x is A_1 '. Note that there is no fuzzification process described in this layer. It is because the original ANFIS is based on singleton fuzzification, and including a singleton fuzzification process will not change the results of O_i^1 . More discussion about this will be made in Section 5.2.

Layer2: Every node labelled Π in this layer calculates the firing strength of a fuzzy rule. The firing strength is simply the product of incoming signals for each node. For instance,

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

Layer3: The nodes labelled N in this layer produce the normalised firing strengths by calculating the ratio of each firing strength of rule i to the sum of firing strengths of all rules.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2.$$

Layer4: The output of the i th node in this layer is the product of the normalised firing strength and the consequent polynomial f_i of the i th fuzzy rule.

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), \quad i = 1, 2.$$

Layer5: The single node labelled Σ computes the summation of all incoming signals which is the overall output of the inference system.

$$O_i^5 = \sum_i \bar{w}_i f_i$$

As it has been mentioned above, the original ANFIS was based on T1 TSK FIS. More recently, an increasing interest in T2 fuzzy systems has led to an increased research interest in T2 (IT2) ANFIS. For example, early studies on T2 ANFIS can be found

in [7, 8], which described an approach that uses an adaptive network to learn a T2 fuzzy system based on linguistic inputs and numeric output. Other examples of research on T2 ANFIS, based on crisp inputs, can be found in [9–14]. As an extension of the studies on ANFIS based on singleton fuzzification, studies on non-singleton fuzzification based IT2 ANFIS can be found in [12, 15, 16]. Though these studies have been made on IT2 ANFIS, a clear architecture for IT2 ANFIS has not been presented in the literature compared to that of T1 ANFIS presented in [5].

Learning Algorithm

As can be seen in Fig. 2.8, there are two shapes of nodes: square and circle. A square node is called an adaptive node which has parameters, while a circle node is a fixed node without parameters. More specifically, the parameters in Layer 1 are for antecedent membership functions, and the parameters in Layer 4 are for consequent linear functions. The learning algorithm of ANFIS is used to tune antecedent and consequent parameters to fit the given input-output data pairs with least square errors in a limit number of learning epochs. In [5], four types of learning algorithms were discussed as follows:

1. *Gradient Descent Only*: All parameters in Layer 1 and 4 are updated by the gradient decent method. Note that this pure gradient decent method is generally slow in terms of convergence and likely to be trapped in local minima.
2. *Gradient Descent and One Pass of LSE*: The LSE method is used in the first training epoch to initialise the consequent parameters in Layer 4. Then all parameters in Layer 1 and 4 are updated by the gradient decent method in further epochs.
3. *Gradient Descent and LSE*: This is the most commonly used learning algorithm for ANFIS in the literature. It is called a hybrid learning algorithm (summarised in Table. 2.1) by which in each training epoch: The consequent parameters in Layer 4 are updated by LSE in the forward pass, and the premise parameters in Layer 1 are updated by the gradient decent method in the backward pass.
4. *Sequential (Approximate) LSE Only*: While all parameters have been linearised, the extended Kalman filter algorithm can be used to update all parameters in Layer

Table 2.1 **Two passes in the hybrid learning procedure of ANFIS**

	Forward Pass	Backward Pass
Premise Parameters	Fixed	Gradient Descent
Consequent Parameters	LSE	Fixed

1 and 4. This method was proposed in the neural network literature and later successfully applied in fuzzy neural network models [13, 14].

In this thesis, we focus on the third type of learning algorithm above, which is the most popular in the literature.

ANFIS: T1 vs IT2

It has been demonstrated that in principle T2 FIS should be able to provide better (and certainly no worse) performance than T1 FIS when they have the same number of membership functions and rules [18]. This can be guaranteed by initialising IT2 membership functions from an optimised T1 FIS. However, in practice, it should be noted that IT2 FIS does not always provide a significant improvement compared to T1 FIS. In fact, IT2 FIS can in some cases be worse than T1 FIS when both of T1 and IT2 membership functions were initialised from scratch [80].

While there has been an increased interest in the research on IT2 ANFIS as mentioned above, very few comparisons have been made between the performance of T1 and IT2 ANFIS models, especially on accessible or reproducible datasets. In [81], it was shown that all IT2 fuzzy-neural-network models were able to produce more accurate predictions than T1 ANFIS models on a Mackey-Glass (MG) problem. However, it is difficult to reproduce such results. For example, based on the same model settings and data, the root-mean-square error (RMSE) of T1 ANFIS in [81] is much larger than what we reproduced, which is the same as that in the original ANFIS paper [5].

Note that it was mentioned in Mendel [17] that least-squares based methods cannot be applied to an IT2 FIS. This is because, although the output of an IT2 FIS can be expressed as a fuzzy basis function term like $\vec{y} = X\vec{\theta}$, the fuzzy basis function expansion X , which is the starting point for the LSE method, cannot be obtained without knowing the consequent parameters $\vec{\theta}$ in advance. A possible solution would be to choose some initial values for $\vec{\theta}$ [10], then the LSE method could be used to estimate $\vec{\theta}$. However, after an update of $\vec{\theta}$, the fuzzy basis function expansion X would need to be recomputed. This iterative process would be continued until it converges in some sense. It is still an open question that whether such an iterative process can be successful.

2.3 Accuracy Measure

Forecasting has always been an attractive research area since it plays an important role in daily life. As one of the most popular research domains, time series forecasting has received particular concern from researchers [19–23]. Many comparative studies have been conducted with the aim of identifying the most accurate methods for time series forecasting [24]. However, research findings indicate that the performance of forecasting methods varies according to the accuracy measure being used [29]. Various accuracy measures have been proposed as the best to use in the past decades. However, many of these measures are not generally applicable due to issues such as being infinite or undefined under certain circumstances, which may produce misleading results. The criteria required for accuracy measures have been explicitly addressed by Armstrong and Collopy [24] and further discussed by Fildes [82] and Clements and Hendry [83]. As discussed, a good accuracy measure should provide an informative and clear summary of the error distribution. The criteria should also include reliability, construct validity, computational complexity, outlier protection, scale-independency, sensitivity to changes and interpretability. It has been suggested by many researchers that no single measure can be superior to all others in these criteria [24, 25, 84].

The evolution of accuracy measures can be seen through the measures used in the major comparative studies of forecasting methods. RMSE and the mean absolute percentage error (MAPE) can be considered as the very early and most popular accuracy measures. They were the primary measures used in the original M-Competition [85]. Despite well-known issues such as their high sensitivity to outliers, they are still widely used [86–88]. When using these accuracy measures, errors which are small and appear to be good, such as 0.1 by RMSE and 1% by MAPE, can often be obtained. Wei et al.[78] employed RMSE as the performance indicator in their research on stock price forecasting. The average error obtained was 84, and it was claimed to be superior to some other previous models. However, without comparison, the error 84 as a number is not easy to interpret. In fact, the average fluctuation of stock indices used was 83 which is smaller than the error of their proposed model. A similar case can be found regarding MAPE. Esfahanipour and Aghamiri [89] proposed a model with an error of 1.3%, which appears to be good. However, this error was larger than the average daily fluctuation of the stock price, which was approximately 1.2%. The poor interpretation here is mainly due to the lack of comparable benchmark used by the accuracy measure.

Armstrong and Collopy [24] recommended the use of relative absolute errors as a potential solution to the above issue. Accuracy measures based on relative errors, such as the mean relative absolute error (MRAE), can provide a better interpretation of how good the evaluated forecasting method performs compared to the benchmark method. However, when the benchmark error is small or equal to zero, the relative error could become extremely large or infinite. This may lead to an undefined mean or at least a distortion of the result. Thus, Armstrong and Collopy suggested a method named ‘winsorizing’ to overcome this problem by trimming extreme values. However, this process will also add some complexity to the calculation, and an appropriate trimming level has to be specified [30].

Similarly, MAPE also has the issue of being infinite or undefined due to zeros in the denominator [90]. The symmetric mean absolute percentage error (sMAPE) was first proposed by Armstrong [91] as a modified MAPE which could be an easy way to

fix the issue of MAPE mentioned above. It was then used in the M3-Competition as an alternative primary measure to MAPE [29]. However, Goodwin and Lawton [26] pointed out that sMAPE is not as symmetric as its name suggested. In fact, it gave more penalties to under-estimates more than to over-estimates. Thus, the use of sMAPE in the M3-Competition was widely criticised by researchers later [92]. In an unpublished working paper, Chen and Yang [93] defined a modified sMAPE, called msMAPE, by adding an additional component to the denominator of sMAPE. The added component can efficiently avoid the inflation of sMAPE caused by zero-valued observations. However, this does not address the issue of asymmetry for sMAPE.

Hyndman and Koehler [30] proposed the mean absolute scaled error (MASE) as a generally applicable measurement of forecasting accuracy without the problems seen in the other accuracy measures. However, this measure can still be dominated by a single large error, though infinite and undefined values have been well avoided for most cases [28]. Davydenko and Fildes [28] proposed an altered version of MASE, the average relative MAE (AvgRelMAE), which uses the geometric mean to average the relative efficiencies of adjustments across time series. Although the geometric mean is appropriate for averaging benchmark ratios [94], the appropriateness of AvgRelMAE still depends on its component measure RelMAE for each time series.

In the rest of this Section, a more detailed review of the accuracy measures that are related to this thesis is provided. This will also let us establish common notations and definitions for these accuracy measures.

For a time series with n observations, let Y_t denote the observation at time t and F_t denote the forecasts of Y_t . Then the forecasting error e_t can be defined as $(Y_t - F_t)$. Let e_t^* denote the forecasting error at time t obtained by some benchmark method. That means $e_t^* = (Y_t - F_t^*)$, where F_t^* is the forecast at time t by the benchmark method.

2.3.1 Scale-dependent Measures

The measures based on absolute or squared errors are also known as scale-dependent measures since their scale depends on the scale of the data. They are useful in comparing

forecasting methods on the same set of data. However, they should not be used across data sets that are on different scales. The most commonly used scale-dependent measures are Mean Absolute Error (MAE), Mean Squared Error (MSE) and RMSE:

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t| \quad (2.28)$$

$$MSE = \frac{1}{n} \sum_{t=1}^n e_t^2 \quad (2.29)$$

$$RMSE = \sqrt{MSE} \quad (2.30)$$

MAE had been cited in the very early forecasting literature as a primary measure of performance for forecasting models [95]. As shown in Equation 2.28, MAE directly calculates the arithmetic mean of absolute errors. Hence, it is very easy to compute and to understand. However, it may produce biased results when extremely large outliers exist in data sets. Specifically, even a single large error can sometimes dominate the result of MAE.

MSE, which calculates the arithmetic mean of squared errors, was used in the first M-Competition [85]. However, its use was widely criticised later as inappropriate [96, 24]. MSE is more vulnerable to outliers since it gives extra weight to large errors. Also, the squared errors are on a different scale from the original data. Thus, RMSE, which is the square root of MSE, is often preferred to MSE as it is on the same scale as the data. However, RMSE is also sensitive to forecasting outliers [97].

2.3.2 Percentage-based Measures

To be scale-independent, a common approach is to use percentage errors based on observed values. Two example measures based on percentage errors are MAPE and sMAPE defined as:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|e_t|}{|Y_t|} \quad (2.31)$$

$$sMAPE = \frac{1}{n} \sum_{t=1}^n \frac{2 \cdot |e_t|}{|Y_t| + |F_t|} \quad (2.32)$$

It should be noted that absolute values are used in the denominator of sMAPE defined in this paper. This definition is different but equivalent to the definition in Makridakis [25] and Makridakis and Hibon [29] when forecasts and actual values are all non-negative. The absolute values in the denominator can avoid negative sMAPE as pointed out by Hyndman and Koehler [30].

MAPE was used as one of the major accuracy measures in the original M-Competition [85]. However, the percentage errors could be excessively large or undefined when the target time series has values close to or equal to zero [90]. Moreover, Armstrong [91] pointed out that MAPE has a bias favouring estimates that are below the actual values. This was illustrated by extremes: *“a forecast of 0 can never be off by more than 100%, but there is no limit to errors on the high side”*. Makridakis [25] discussed the asymmetric issue of MAPE with another example which involves two forecasts on different actual values. However, we believe that the example by Makridakis is beyond the idea of Armstrong in 1985. To our understanding, we believe that the assumption concerning the asymmetric issue of MAPE described by Armstrong [91] is: i), the estimates are non-negative while the actual value is positive; ii) the forecasting range is asymmetric that 0 is the lower bound for lower estimates while there is no upper bound for upper estimates; iii), errors for lower estimates and upper estimates should be symmetric (an extreme case: 0 as the worst lower estimate should have the same absolute error as the worst upper estimate which is infinite).

sMAPE can produce symmetric errors in the asymmetric forecasting range as stated in the above assumption. However, it is more natural to consider the symmetric property in a symmetric forecasting range for lower and upper estimates. Thus, sMAPE was widely criticised as an asymmetric measure [26, 92, 30]. Regardless of the asymmetric

issue, an advantage of sMAPE is that it does not have the issue of MAPE from being excessively large or infinite. Also, due to the error bounds defined, sMAPE is more resistant to outliers since it gives less significance to outliers compared to other measures which do not have bounds for errors.

2.3.3 Relative-based Measures

Another approach for accuracy measures to be scale-independent is to use relative errors based on the errors produced by a benchmark method (e.g. the naïve method). The most commonly used such measures are MRAE and the geometric mean relative absolute error (GMRAE):

$$MRAE = \frac{1}{n} \sum_{t=1}^n \left| \frac{e_t}{e_t^*} \right| \quad (2.33)$$

$$GMRAE = \left(\prod_{t=1}^n \left| \frac{e_t}{e_t^*} \right| \right)^{\frac{1}{n}} = \exp \left(\frac{1}{n} \sum_{t=1}^n \ln \left(\frac{e_t}{e_t^*} \right) \right) \quad (2.34)$$

MRAE can provide a clearer intuition of the performance improvement compared to the benchmark method. However, MRAE has a similar limitation as MAPE, in that it can also be excessively large or undefined, when e_t^* is close to or equal to zero.

GMRAE is favoured since it is generally acknowledged that the geometric mean is more appropriate for averaging relative quantities than the arithmetic mean [24, 82]. According to an alternative representation of GMRAE shown above in Equation 2.34, a key step for calculating GMRAE is to make an arithmetic mean of log-scaled error ratios. This makes GMRAE more resistant to outliers compared to MRAE which uses the arithmetic mean of original error ratios. However, GMRAE is still sensitive to outliers. More specifically, GMRAE can be dominated by not only a single large outlier, but also an extremely small error close to zero. This is because there is neither upper bound nor lower bound for the log-scaled error ratios used by GMRAE. Also, it should also be

noticed that zero errors, both in e_t and e_t^* , have to be excluded from the analysis. Thus, GMRAE may not be sufficiently informative.

Rather than use the average of relative errors, one can also use the relative of average errors obtained by a base measure. For example, when the base measure is RMSE, then relative RMSE (RelRMSE) is defined as:

$$RelRMSE = \frac{RMSE}{RMSE^*} \quad (2.35)$$

RelRMSE is a commonly used measure proposed by Armstrong and Collopy [24] where $RMSE^*$ denotes the $RMSE$ produced by a benchmark method. Similar measures, such as RelMAE and RelMAPE, can be easily defined. They are also called relative measures. An advantage of relative measures is their interpretability [30]. However, the performance of relative measures is restricted by the component measure. For example, RelMAPE is also undefined when MAPE is undefined. Further, RelMAPE can also be easily dominated by extreme large outliers since MAPE is not resistant to outliers. Thus, it makes no sense to compute RelMAPE if MAPE, as the component, is skewed.

Another disadvantage of relative measures is that they are only available when there are several forecasts on the same series [30]. As a related idea of relative measures, MASE does not have the above issue. It is defined as:

$$MASE = \frac{1}{n} \sum_{t=1}^n \frac{|e_t|}{MAE^{**}} \quad (2.36)$$

In MASE, the absolute error $|e_t|$ for each observation is scaled by the average *in-sample* error MAE^{**} produced a benchmark method (e.g. one-step naïve method, or seasonal naïve method for seasonal data). Thus, MASE will not produce infinite or undefined values except in the irrelevant case where all historical data are equal. However, MASE is still vulnerable to outliers [28]. Moreover, it has to be assumed that

the period-to-period difference of the time series is stationary so that the scaling factor is a consistent estimator of the scale of the series.

For comparisons of forecasting methods on multiple time series, MASE is equivalent to the weighted arithmetic mean of relative MAEs [28]:

$$MASE = \frac{1}{N} \sum_{i=1}^m n_i r_i, \quad r_i = \frac{MAE_i}{MAE_i^{**}} \quad (2.37)$$

where m denotes the number of time series, n_i denotes the number of observations for the i^{th} time series and $N = \sum_{i=1}^m n_i$. As pointed out by Davydenko and Fildes [28], using the arithmetic mean of MAE ratios introduces a bias towards overrating the accuracy of a benchmark method. They proposed the measure AvgRelMAE as an alternative to MASE, based on the geometric mean to average the scaled quantities.

$$AvgRelMAE = \left(\prod_{i=1}^m r_i^{n_i} \right)^{\frac{1}{N}}, \quad r_i = \frac{MAE_i}{MAE_i^*} \quad (2.38)$$

It should be noticed that AvgRelMAE uses *out-of-sample* MAE_i^* as the scaling factor while MASE uses *in-sample* MAE_i^{**} . Though AvgRelMAE was shown to have many advantages such as interpretability and robustness [28], it still has the same issue with MASE since they are based on RelMAE. As mentioned above, the accuracy of RelMAE is constrained by the accuracy of MAE. Since MAE can be dominated by extreme outliers, the MAE ratio r_i does not necessarily represent an advisable comparison of forecasting methods based on the errors of the majority of forecasts for the i^{th} time series.

2.4 Summary

This chapter has provided an overview of the background materials and existing literature that are used and referred to in this thesis. By providing a brief review of fuzzy sets

and FISs, this chapter has allowed us to establish common notations and definitions for fuzzy sets and systems. Specifically, the definitions for T1, GT2 and IT2 fuzzy sets have been provided. Note that IT2 fuzzy sets are special cases of GT2 fuzzy sets. In terms of computational simplicity, GT2 fuzzy sets and systems are not investigated in this thesis. Hence, the set-theoretical operations have only been introduced for T1 and IT2 fuzzy sets. A brief review of T1 and IT2 FISs has been made. A more detailed introduction has been presented for the key components of the typical T1 and T2 fuzzy inference systems illustrated in Fig. 2.5 and 2.6. Since type reduction has been considered to be a bottleneck for the efficiency of a FIS, a review has been made on the well-known KM algorithm and its enhancements which are the most commonly used algorithms for type reduction. To the best of our knowledge, all existing KM related algorithms are iterative-based methods. It is still a challenge to improve the efficiency of these algorithms.

A review has been made on ANFIS which is a commonly used architecture for optimising the performance of a FIS. In many studies, ANFIS has been shown to be superior to models using other techniques such as autoregression, genetic algorithms and artificial neural network. Though some studies have been made on IT2 ANFIS, a clear architecture for IT2 ANFIS has not been presented in the literature. Also, very few comparisons have been made between the performance of T1 and IT2 ANFIS models, especially on accessible or reproducible datasets. Though the hybrid learning algorithm used for T1 ANFIS can also be easily applied to IT2 ANFIS, it is still an open question that whether such an iterative learning algorithm can be successful for IT2 ANFIS.

Since proper accuracy measures are very important to conduct objective comparisons of different models, a review of commonly used accuracy measures in the literature has been made focusing on their advantages and disadvantages (a summary is provided in Chapter 3 after further comparisons). Though sMAPE has been widely criticised for its asymmetric issue, it has many advantages. It does not have the issue of many measures such as MAPE for being excessively large or infinite. Also, due to the error bounds

defined, sMAPE is more resistant to outliers since it gives less significance to outliers compared to other measures which do not have bounds for errors.

In the next chapter, based on bounded relative error, we propose a new accuracy measure which combines the best features of various alternative measures without suffering from their issues.

Chapter 3

A New Accuracy Measure based on Bounded Relative Error

3.1 Introduction

As declared in Section 1.2, one of the objectives of this thesis is to make fair comparisons. This chapter proposes a new accuracy measure to address the issues of commonly used measures reviewed in Section 2.3. Specifically, by introducing a novel bounded relative absolute error, the newly proposed measure can address the asymmetric issue of sMAPE while maintaining its other properties, such as scale-independence and outlier resistance. Further, compared to sMAPE which uses the percentage errors based on the observation values, the new measure improves the interpretability based on relative errors with a selectable benchmark. Given that [24] claimed that measures based on relative errors are the most reliable, we believe our measure is reliable in this sense.

The remainder of this chapter is organised as follows. The newly proposed accuracy measure is then introduced in Section 3.2. The next two Sections 3.3 and 3.4 show the results of an evaluation of the proposed measure and discuss its relationship to other measures. Finally, a summary is presented in Section 3.5.

3.2 A New Accuracy Measure

The criteria for a useful accuracy measure have been explicitly addressed in the literature [24, 82–84]. As reviewed in Section 2.3, many measures have been proposed with various advantages and disadvantages. However, most of these measures suffer from one or more issues. In this section, we propose a new accuracy measure which adopts the advantages of other measures such as sMAPE and MRAE without having their common issues. Specifically, the proposed measure is designed to have the following properties: (i) Informative: it can provide an informative result without the need to trim errors; (ii) Resistant to outliers: it would not be easily dominated by a single forecasting outlier; (iii) Symmetric: overestimates and underestimates are treated equally; (iv) Scale-independent: it can be applied to datasets on different scales; (v) Interpretability: it is easy to understand and can provide intuitive results.

It has been discussed in Section 2.3.2 that sMAPE is resistant to outliers due to bounded error defined. We would like to propose a new measure in a similar fashion to sMAPE without its issues. Since relative errors are more general than percentage errors, we use the Relative Absolute Error (RAE) as the base to derive our new measure.

$$RAE = \frac{|e_t|}{|e_t^*|} \quad (3.1)$$

Since RAE has no upper bound, it can be excessively large or undefined when $|e_t^*|$ is small or equal to zero. This issue can be easily addressed by adding a $|e_t|$ to the denominator of RAE, which introduces a bounded RAE (BRAE):

$$BRAE = \frac{|e_t|}{|e_t| + |e_t^*|} \quad (3.2)$$

In BRAE, the added $|e_t|$ can ensure that the denominator will be no less than the numerator. It means BRAE will have a maximum error of 1 while the minimum error is 0 when $|e_t|$ is equal to zero. Due to the upper bound of BRAE, an accuracy measure based on BRAE will be more resistant to forecasting outliers. It can be noticed that the asymmetric issue of sMAPE has also been addressed in BRAE by adding a $|e_t|$ rather

than a $|F_t|$ to the denominator. Also, a measure based on BRAE is more appropriate than sMAPE for intermittent demand data which have many zero-valued observations. To avoid the issue of being undefined, BRAE is defined to be 0.5 for the special case when $|e_t|$ and $|e_t^*|$ are both equal to zero.

In practice, the one-step naïve method is a commonly used benchmark method where $e_t^* = Y_{t-1} - Y_t$. However, it should be noticed that the naïve method is not necessarily an effective benchmark method. For example, when most forecasting methods can generally produce much smaller errors than the naïve method, BRAE will have the same issue as percentage error based measure stated above. Thus, it is preferable to use a properly competitive method as a benchmark, such that a value of around 0.5 is obtained by BRAE.

Based on BRAE, a measure called Mean Bounded Relative Absolute Error (MBRAE) can be defined as:

$$MBRAE = \text{mean}(BRAE) = \frac{1}{n} \sum_{t=1}^n \frac{|e_t|}{|e_t| + |e_t^*|} \quad (3.3)$$

Though MBRAE is adequate to compare forecasting methods, it is a scaled error that cannot be directly interpreted as a normal error ratio reflecting the error size. In fact, the process of calculating GMRAE also contains a mean of log-scaled error ratio which is not easily interpretable. But this issue is addressed by converting the log-scaled error to a normal ratio with the exponential function. Similarly, a transformation can be made to MBRAE to obtain a more interpretable measure which is termed the unscaled MBRAE (UMBRAE):

$$UMBRAE = \frac{MBRAE}{1 - MBRAE} \quad (3.4)$$

With UMBRAE, the performance of a proposed forecasting method can be interpreted as follows: when $UMBRAE$ is equal to 1, the proposed method performs approximately the same as the benchmark method; when $UMBRAE < 1$, the proposed method performs about $(1 - UMBRAE) * 100\%$ better than the benchmark method; when

$UMBRAE > 1$, the proposed method is approximately $(UMBRAE - 1) * 100\%$ worse than the benchmark method.

In general, UMBRAE is informative without the need to trim extreme errors. At the same time, based on the bounded errors, UMBRAE is resistant to outliers. It is also symmetric and obviously scale-independent. The benchmark used by UMBRAE is selectable where the naïve method can be easily applied. To the best of our knowledge, UMBRAE has not been proposed before. We suggest it as a generally applicable accuracy measure for time series forecasting. UMBRAE would be particularly useful for the cases where results are not supposed to be dominated by forecasting outliers.

3.3 Evaluation and Results

In this section, the performance of UMBRAE is evaluated. The naïve method is used as the benchmark for UMBRAE. Properties such as reliability and sensitivity have been well investigated in the study by Armstrong and Collopy [24]. In their study, MAPE and MRAE have been assessed to be acceptable in terms of reliability and good in terms of sensitivity. In fact, these properties, especially reliability, cannot be easily examined. For example, in the reliability tests, if forecasting methods are expected to have the same rankings when they are evaluated by a reliable accuracy measure, these forecasting methods themselves have to perform stably on different time series. It is difficult to find such forecasting methods in the real world. Thus, these properties are not examined in our study. Instead, it is assumed that UMBRAE, based on relative errors, will also be reliable and sensitive to error changes. Consequently, our evaluation will be mainly focused on the properties of outlier resistance, symmetry, and scale-independence. To make comparisons, other commonly used measures (MAE, RMSE, MASE, MRAE, GMRAE, MAPE and sMAPE) reviewed in Section 2.3 are also examined in our evaluation. Comparisons are firstly made with synthetic time series to specifically examine the required properties. Then the M3-Competition data

with 3003 time series [29] are used to demonstrate how these measures perform with real-world data.

3.3.1 Evaluation with Synthetic Data

Three groups of synthetic time series data are used in the comparative study. These synthetic data are not designed to be representative of real-world data. Rather, they are selected to clearly show the drawbacks of accuracy measures in terms of the required properties. In the synthetic evaluations, the average one-step naïve error is used to scale errors for MASE.

One of the most desired properties of an accuracy measure is the ability to resist outliers. Thus, the first group of synthetic data is made to examine whether the accuracy measure is resistant to a single forecasting outlier. As shown in Fig 3.1, Y_t is the objective time series with 10 observations, which are randomly generated under the normal distribution (mean = 300, sd = 100). F_t^n is the forecasting series of Y_t . Specifically, F_t^1 does not have obvious forecasting outliers, and it has an error of approximately 10% measured by MAPE. The other three forecasts are the same as F_t^1 except that they all have a forecasting outlier for the eighth observation. Though occasionally occurring large errors should also be considered in evaluating the performance of a forecasting method, it is assumed that a single large outlier should not affect the whole performance significantly. However, the results in Fig 3.1 shows that the errors reported by some accuracy measures have been significantly dominated by the single forecasting outlier. The worst is RMSE where its error for F_t^4 has become approximately 36 times larger than its error for F_t^1 . Though MASE has been scaled from MAE, it in fact performs the same as MAE in dealing with the forecasting outlier. The errors given by MAE and MASE for F_t^4 have both been distorted to be about 15 times larger than for F_t^1 . In contrast, sMAPE, GMRAE and UMBRAE are less sensitive to this single forecasting outlier. UMBRAE reports the smallest difference between four time series.

The second group of time series data is created to evaluate whether over-estimates and under-estimates are treated equally by the accuracy measures. As presented in

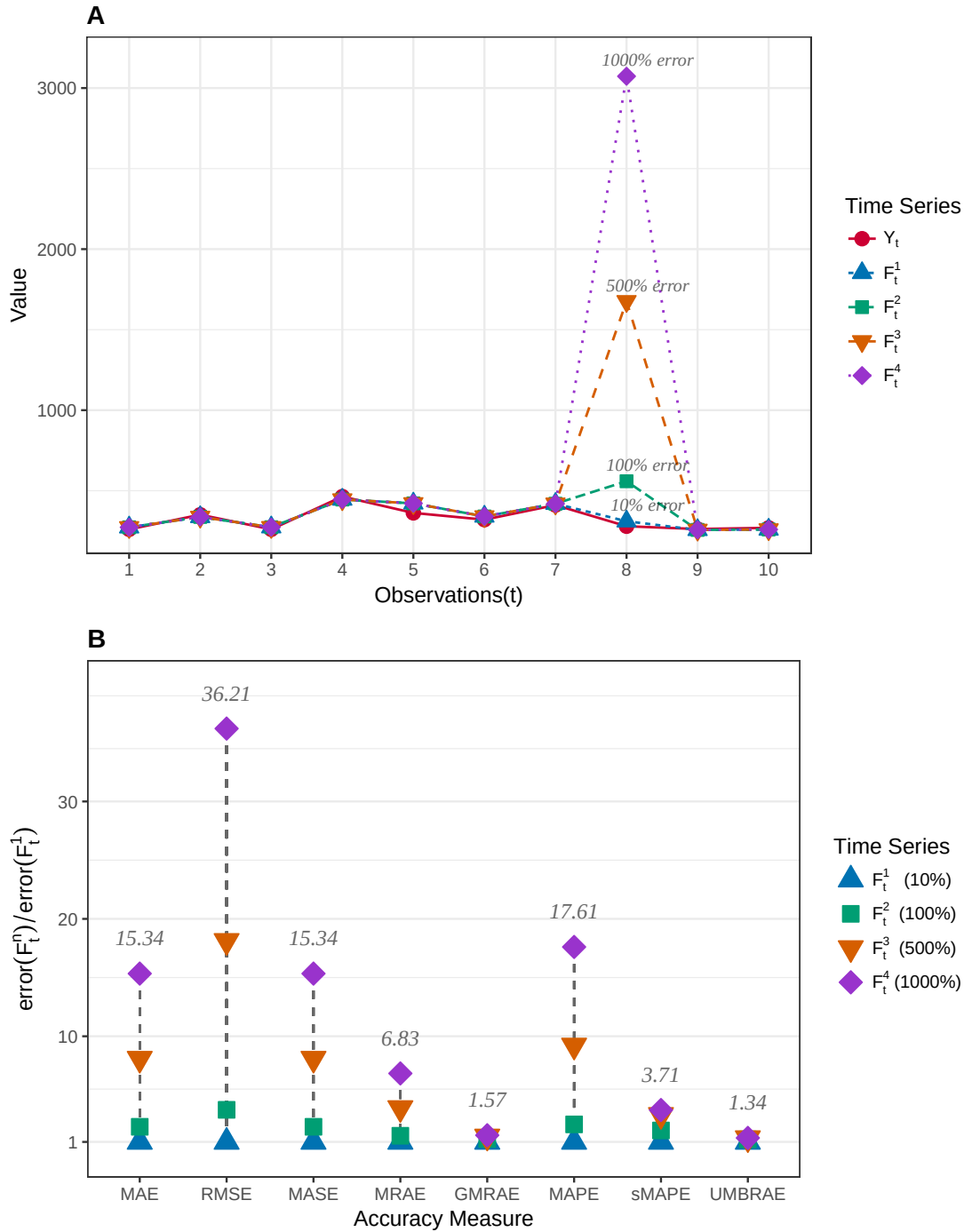


Fig. 3.1 **Evaluation on the resistance of accuracy measures to a single forecasting outlier.** A: Synthetic time series data where Y_t is the target series and F_t^n are forecasts. The only difference between F_t^n is their forecasts on the observation Y_8 . B: Results of single forecasting outlier evaluation, which shows UMBRAE is less sensitive than other measures to a single forecasting outlier.

Fig 3.2, Y_t is the same time series as which was used in the single forecasting outlier resistance evaluation. In this scenario, F_t^1 makes a 10% over-estimate error to all observations in Y_t while F_t^2 makes a 10% under-estimate. The results in Fig 3.2 show that all the accuracy measures except sMAPE have given the same error for F_t^1 and F_t^2 . sMAPE produces a larger error for F_t^2 which indicates it puts a heavier penalty on under-estimates than on over-estimates.

Davydenko and Fildes [28] suggested another scenario to examine the property of symmetry for measures. In this scenario, the reward given for improving the benchmark is expected to balance the penalty given for reducing the benchmark by the same quantity. We also use this to examine our measure UMBRAE. Suppose that a time series has only two observations (y) and there is one forecasting method to be compared with another benchmark method. For the benchmark method, it makes the forecasts f with errors ($y - f$) of 1 and 2 respectively. In contrast, the forecasting method produces errors of 2 and 1 respectively. As an expected result, the forecasting method has an error of 1 measured by UMBRAE based on the benchmark method. Thus, UMBRAE is also symmetric for this case.

Normally, the scale-dependent issue of accuracy measures is related to their capability of evaluating forecasting performance across data series on different scales. Accuracy measures based on percentages or relative ratios are suited to perform such evaluations, and no synthetic data are made for this. However, the scale-dependent issue also exists within a time series. Thus, the third group of synthetic data shown in Fig 3.3 is made to evaluate the property of accuracy measures dealing with data on different scales within a single time series. In this dataset, Y_t is a time series generated by the Fibonacci sequence from 2 to 144. As the forecasts to Y_t , all forecasting values of F_t^1 are set to have a 20% over-estimate error of the relevant observation of Y_t . In contrast, F_t^2 has the same mean absolute error as F_t^1 , but its errors are on different percentage scales from 1440% to 0.2%. Specifically, F_n^2 has the same absolute error as F_{11-n}^1 . For instance, F_1^2 has the same absolute error as F_{10}^1 which is 28.8. As presented in Fig 3.3, MAE, RMSE, MASE and even GMRAE do not show any difference between the two forecasts. MRAE and

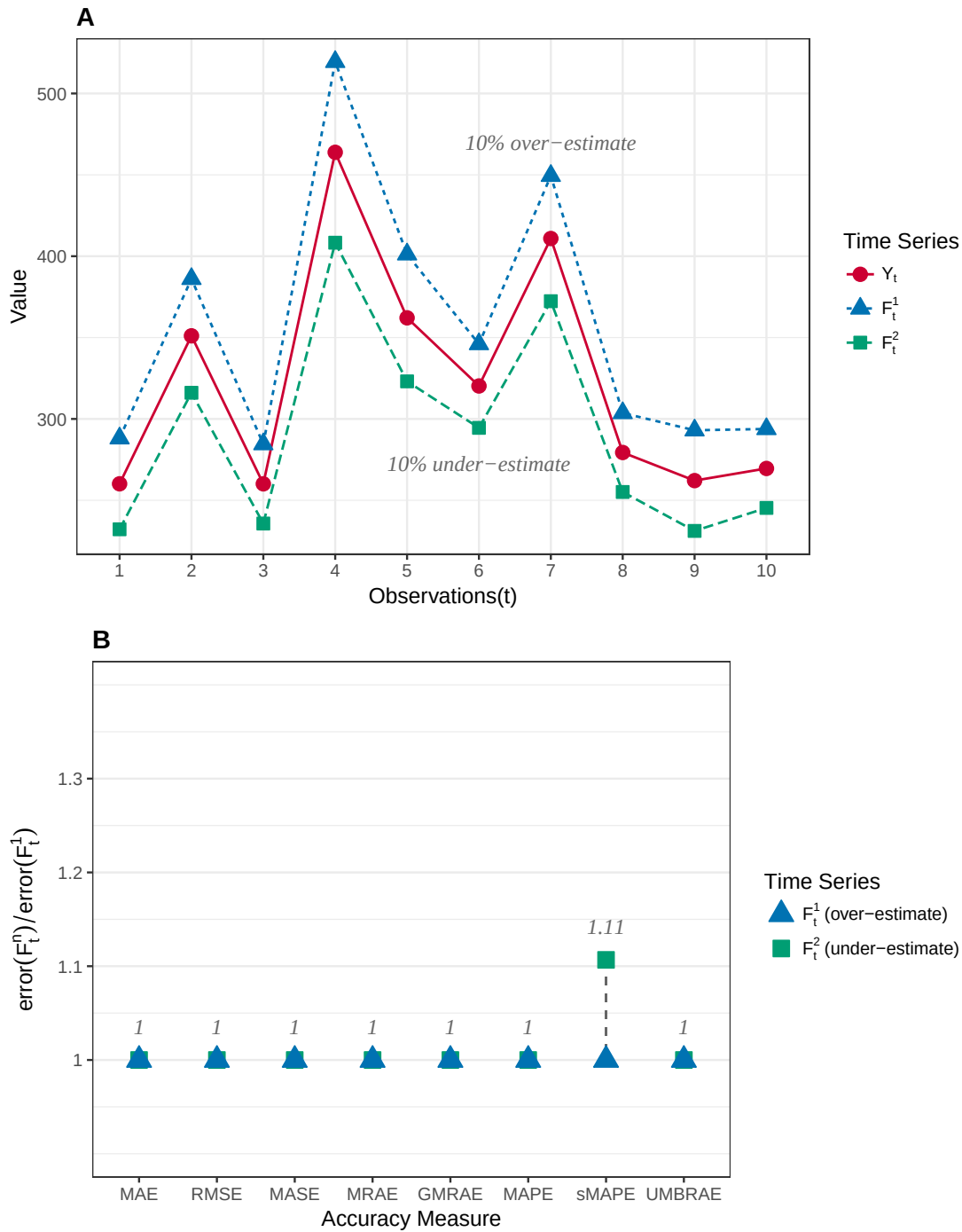
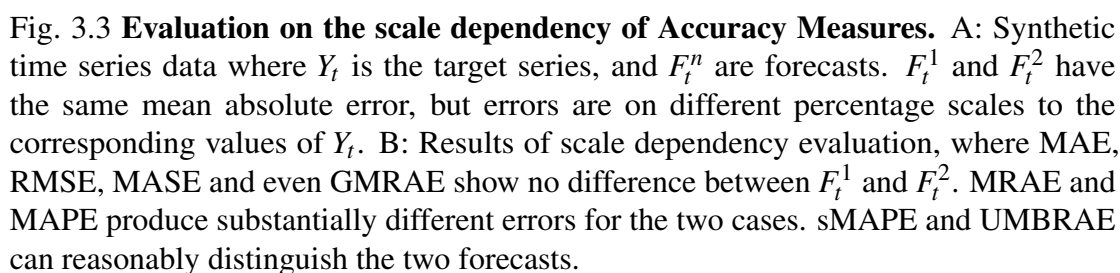


Fig. 3.2 Evaluation on the symmetry of accuracy measures to over-estimates and under-estimates. A: Synthetic time series data where Y_t is the target series, and F_t^n are forecasts. F_t^1 makes a 10% over-estimate to all observations of Y_t , while F_t^2 makes a 10% under-estimate. B: Results of symmetric evaluation, which shows UMBRAE and all other accuracy measures except SMAPE are symmetric.



MAPE, however, have produced substantially different results for the two cases. The errors measured by them for F_t^2 are approximately ten times larger than for F_t^1 . In contrast, UMBRAE and sMAPE give a moderate difference between the two forecasts.

3.3.2 Evaluation with the M3-Competition Data

The M-Competitions are well-known empirical studies which employ various real-world time series data in comparing the performance of forecasting methods. In this study, we use the M3-Competition data [29] which contains 3003 time series to evaluate our proposed measure. The forecasting data are available with R package ‘Mcomp’ maintained by Hyndman. The ‘Mcomp’ package for R is available from webpage: <https://CRAN.R-project.org/package=Mcomp>. Among the 24 forecasting methods in the M3-Competition, 22 are used in our evaluation since their forecasts are available for all the 3003 time series. Since the one-step naïve method is used by many accuracy measures as the benchmark, it is also listed in the results as a forecasting method. As an alternative version of MASE, AvgRelMAE which use geometric mean to average errors across time series is also included in this evaluation. To simplify the comparisons, errors are measured at the first six forecasting horizons across the 3003 time series, which is available from all of the 22 forecasting methods.

The results are listed in Table 3.1. It can be noticed that errors by MAE and RMSE are relatively large numbers which are meaningless without comparisons. UMBRAE is able to give interpretable results where a forecasting method with an $error < 1$ can be considered to be better than the benchmark method in terms of the average relative absolute error based on BRAE. As shown in the results, the naïve method, which is the benchmark used by UMBRAE, has an error of 1. Errors of other forecasting methods measured by UMBRAE are all less than 1. This indicates that these forecasting methods are better than the naïve method. However, MRAE gives the opposite result in which the naïve method is ranked as the best. It has to be mentioned that all the errors excluding that for the naïve method measured by AvgRelMAE are smaller than 1, whereas all the errors measured by MASE are much larger than 1. The rank correlation coefficient

of different measures is shown in Table 3.2 (see next page). The correlation between RMSE (or MRAE) and other measures is extremely low. In contrast, UMBRAE shows substantially high agreement with most of other measures, where the average Spearman rank correlation is 0.516. Particularly, UMBRAE has remarkably high correlations with GMRAE and AvgRelMAE which are 0.995 and 0.990 respectively.

— intentionally left blank —

Table 3.1 Results on M3-Competition data at first six forecasting horizons.

Forecasting Method	Accuracy Measure																	
	MAE		RMSE		MASE		AvgRelMAE		MRAE		GMRAE		MAPE		sMAPE		UMBRAE	
	Error	Rank	Error	Rank	Error	Rank	Error	Rank	Error	Rank	Error	Rank	Error(%)	Rank	Error(%)	Rank	Error	Rank
THETA	589.2	2	1452	11	1.742	1	0.7527	1	2.06	5	0.7186	1	14.85	2	11.49	1	0.7882	1
ForecastPro	612.3	6	1720	15	1.821	6	0.7556	2	2.62	16	0.7212	2	15.28	4	11.72	2	0.7948	2
ForcX	577.0	1	1269	5	1.768	2	0.7560	3	2.22	9	0.7274	3	14.81	1	11.72	2	0.8028	3
COMB S-H-D	609.1	5	1477	12	1.775	3	0.7784	5	1.86	4	0.7508	5	15.70	8	12.02	4	0.8106	4
DAMPEN	630.2	10	1779	16	1.817	5	0.7767	4	2.20	8	0.7473	4	15.85	10	12.05	5	0.8132	5
AutoBox2	599.7	3	1190	3	2.151	23	0.7887	6	2.48	13	0.7557	6	15.86	11	12.47	10	0.8216	6
PP-Autocast	640.9	13	1795	19	1.934	13	0.7941	7	2.37	10	0.7633	8	15.78	9	12.28	7	0.8266	7
HOLT	667.9	18	1822	20	1.899	10	0.7953	8	2.77	17	0.7616	7	16.87	17	13.16	17	0.8270	8
B-J auto	642.1	14	1789	17	1.881	8	0.7957	9	2.40	11	0.7711	10	16.34	13	12.39	9	0.8295	9
WINTER	724.5	21	7326	23	1.997	17	0.8062	10	2.54	14	0.7719	11	18.20	22	13.26	18	0.8340	10
Auto-ANN	601.5	4	1180	2	1.896	9	0.8075	12	2.58	15	0.7687	9	15.15	3	12.47	10	0.8386	11
ARARMA	746.4	23	4022	22	1.967	16	0.8138	14	3.03	22	0.7751	12	17.47	20	12.91	16	0.8408	12
Flors-Pearc1	656.6	15	1790	18	1.877	7	0.8124	13	2.43	12	0.7816	13	16.75	15	12.78	14	0.8412	13
ROBUST-Trend	669.7	19	1651	14	1.806	4	0.8074	11	2.16	7	0.7866	14	18.01	21	13.49	19	0.8500	14
SMARTFCS	631.2	11	1496	13	1.920	12	0.8161	15	2.80	18	0.7936	15	15.70	8	12.32	8	0.8552	15
AutoBox3	664.3	17	1437	10	1.953	14	0.8335	16	3.01	21	0.8121	16	16.60	14	13.62	21	0.8668	16
THETasm	620.0	9	1264	4	1.901	11	0.8407	17	2.14	6	0.8147	17	15.37	5	12.80	15	0.8689	17
AutoBox1	728.7	22	2506	21	2.093	21	0.8465	18	3.21	23	0.8151	18	17.40	19	13.64	22	0.8721	18
RBF	615.7	8	1329	8	2.016	19	0.8496	20	2.96	20	0.8351	19	15.67	6	12.26	6	0.8817	19
Flors-Pearc2	635.1	12	1287	6	1.964	15	0.8465	18	2.87	19	0.8361	20	16.82	16	12.76	13	0.8831	20
SINGLE	615.3	7	1157	1	2.000	18	0.8831	21	1.47	3	0.8713	21	16.09	12	12.68	12	0.9098	21
naïve2	659.8	16	1306	7	2.035	20	0.9054	22	1.41	2	0.9037	22	17.20	18	13.55	20	0.9314	22
naïve	719.1	20	1412	9	2.134	22	1.0000	23	1.00	1	1.0000	23	18.91	23	14.69	23	1.0000	23

The bold numbers highlight the best performance and the italic numbers show the worst. The calculation of MRAE and GMRAE does not include the observations where the naïve error is zero. Zero relative absolute errors are trimmed for GMRAE.

Table 3.2 **Spearman's rank correlation coefficient of the rankings in Table 3.1.**

Accuracy Measure	MAE	RMSE	MASE	AvgRelMAE	MRAE	GMRAE	MAPE	sMAPE	UMBRAE
MAE	—	0.650	0.409	0.463	0.278	0.493	0.895	0.821	0.476
RMSE	0.650	—	−0.110	−0.254	0.357	−0.229	0.449	0.168	−0.260
MASE	0.409	−0.110	—	0.711	0.220	0.687	0.510	0.590	0.679
AvgRelMAE	0.463	−0.254	0.711	—	0.054	0.985	0.481	0.687	0.990
MRAE	0.278	0.357	0.220	0.054	—	0.010	0.079	0.110	0.026
GMRAE	0.493	−0.229	0.687	0.985	0.010	—	0.532	0.702	0.995
MAPE	0.895	0.449	0.510	0.481	0.079	0.532	—	0.841	0.513
sMAPE	0.821	0.168	0.590	0.687	0.110	0.702	0.841	—	0.706
UMBRAE	0.476	−0.260	0.679	0.990	0.026	0.995	0.513	0.706	—
Average	0.561	0.096	0.462	0.515	0.142	0.522	0.538	0.578	0.516

To eliminate the influence of outliers and extreme errors, we also use trimmed means to evaluate the accuracy measures. A 3% trimming level is used in our study. As shown in Table 3.3, most errors measured by MAE, RMSE, MASE, MRAE and MAPE have significant differences compared to that without trimming shown in Table 3.1. The rankings of forecasting methods made by these measures also have significant changes. In contrast, errors and rankings measured by other measures have fewer changes. Particularly, the value of UMBRAE is quite invariant to trimming, where differences appear only after the third decimal point for most of the forecasting methods. It can also be noticed that the rankings made by UMBRAE in Table 3.3 keep the same as that in Table 3.1. In general, all the measures except MRAE have similar rankings. As shown in Table 3.3, the rank correlations between UMBRAE and other measures are much higher on average as shown in Table 3.4.

Table 3.3 Results with a 3% trimming level on M3-Competition data at first six forecasting horizons.

Accuracy Measure																								
Forecasting Method	MAE		RMSE		MASE		AvgRelMAE		MRAE		GMRAE		MAPE		sMAPE		UMBRAE							
	Error	Rank	Error	Rank	Error	Rank	Error	Rank	Error	Rank	Error	Rank	Error(%)	Rank	Error(%)	Rank	Error	Rank						
THETA	489.8	1	805.5	2	1.549	1	0.7410	1	1.38	8	0.7360	1	10.49	1	10.18	1	0.7825	1						
ForecastPro	496.6	3	820.6	3	1.582	3	0.7424	2	1.40	10	0.7395	2	10.79	2	10.37	2	0.7893	2						
ForcX	494.5	2	802.1	1	1.577	2	0.7456	3	1.33	5	0.7477	3	10.80	3	10.46	3	0.7977	3						
COMB S-H-D	508.7	4	831.0	4	1.583	4	0.7662	5	1.28	4	0.7628	5	11.19	4	10.70	5	0.8055	4						
DAMPEN	509.8	5	840.3	5	1.592	5	0.7653	4	1.35	6	0.7662	7	11.28	6	10.69	4	0.8081	5						
AutoBox2	520.8	7	842.8	7	1.599	6	0.7776	6	1.50	14	0.7605	4	11.62	11	11.06	11	0.8166	6						
PP-Autocast	518.6	6	856.6	9	1.629	7	0.7796	7	1.44	12	0.7716	8	11.38	10	10.86	6	0.8218	7						
HOLT	541.7	14	905.5	16	1.674	12	0.7803	8	1.67	18	0.7656	6	11.97	14	11.42	14	0.8221	8						
B-J auto	524.3	9	860.7	10	1.653	9	0.7834	9	1.36	7	0.7736	9	11.67	12	11.04	10	0.8248	9						
WINTER	544.4	16	909.0	17	1.682	14	0.7903	10	1.66	16	0.7786	10	12.06	15	11.48	18	0.8293	10						
Auto-ANN	523.6	8	841.0	6	1.681	13	0.7947	11	1.54	15	0.7909	11	11.21	5	11.01	8	0.8342	11						
ARARMA	547.5	17	913.4	18	1.673	11	0.7986	13	1.74	20	0.7943	13	12.17	18	11.42	14	0.8362	12						
Flors-Pearl	539.3	13	889.8	15	1.661	10	0.8002	14	1.40	10	0.8002	14	12.15	17	11.40	13	0.8367	13						
ROBUST-Trend	567.6	20	955.3	21	1.635	8	0.7971	12	1.47	13	0.8080	15	12.62	21	11.87	19	0.8458	14						
SMARTFCS	529.9	10	862.3	11	1.715	16	0.8047	15	1.67	18	0.7937	12	11.29	8	11.01	8	0.8511	15						
AutoBox3	566.2	19	945.7	20	1.749	17	0.8215	16	1.81	21	0.8167	16	12.28	19	11.94	20	0.8628	16						
THETasm	538.8	12	870.7	13	1.713	15	0.8287	17	1.40	10	0.8359	18	11.31	9	11.37	12	0.8652	17						
AutoBox1	576.8	22	972.0	22	1.812	20	0.8320	18	1.85	23	0.8349	17	12.59	20	12.07	21	0.8682	18						
RBF	533.2	11	863.7	12	1.827	21	0.8369	19	1.83	22	0.8550	20	11.28	6	10.98	7	0.8781	19						
Flors-Pearce2	555.1	18	887.2	14	1.785	18	0.8374	20	1.69	19	0.8370	19	12.12	16	11.45	17	0.8796	20						
SINGLE	543.6	15	854.5	8	1.808	19	0.8670	21	1.06	3	0.8947	21	11.83	13	11.43	16	0.9089	21						
naïve2	576.1	21	924.0	19	1.844	22	0.8870	22	1.02	2	0.9256	22	12.63	22	12.14	22	0.9323	22						
naïve	633.9	23	1001.9	23	1.944	23	1.0000	23	1.00	1	1.0000	23	14.02	23	13.26	23	1.0000	23						

The bold numbers highlight the best performance and the italic numbers show the worst.

Table 3.4 Spearman's rank correlation coefficient of the rankings in Table 3.3.

Accuracy Measure	MAE	RMSE	MASE	AvgRelMAE	MRAE	GMRAE	MAPE	sMAPE	UMBRAE
MAE	—	0.951	0.805	0.828	0.256	0.820	0.940	0.970	0.839
RMSE	0.951	—	0.707	0.710	0.345	0.708	0.929	0.909	0.720
MASE	0.805	0.707	—	0.952	0.239	0.912	0.631	0.751	0.948
AvgRelMAE	0.828	0.710	0.952	—	0.135	0.980	0.688	0.771	0.996
MRAE	0.256	0.345	0.239	0.135	—	0.077	0.143	0.156	0.133
GMRAE	0.820	0.708	0.912	0.980	0.077	—	0.684	0.753	0.985
MAPE	0.940	0.929	0.631	0.688	0.143	0.684	—	0.950	0.697
sMAPE	0.970	0.909	0.751	0.771	0.156	0.753	0.950	—	0.781
UMBRAE	0.839	0.720	0.948	0.996	0.133	0.985	0.697	0.781	—
Average	0.801	0.747	0.743	0.758	0.186	0.740	0.708	0.755	0.762

To show the error distributions in a similar manner to that in [28], we use the errors produced by the forecasting method ForecastPro as an example. Figs 3.4 to 3.11 show the distributions of the eight underlying error measurements used in the nine accuracy measures discussed above. In each figure, the top plot shows the kernel density estimate of the errors illustrating its distribution, while the bottom shows a box-and-whisker plot which more clearly highlights the outliers. From these figures, it can be observed that the distribution of error measurements used in UMBRAE is more evenly distributed, with fewer outliers than in the other measures.

3.4 Discussion

Fig 3.1 shows that MRAE and MAPE can be easily dominated by a single forecasting outlier. This is because they are based on the arithmetic mean, and there is no upper bound defined for the single error. In practice, the poor resistance to forecasting outliers may produce misleading results. This can be illustrated by our evaluation on the M3-Competition data. As shown in Table 3.1, MRAE gives significantly different rankings from other measures. It suggests the naïve method performs the best while almost all the other accuracy measures indicate that the naïve method is the worst. By examining the forecasting data, we can find that the results measured by MRAE are seriously distorted by the extreme large relative absolute errors where the naïve errors are small.

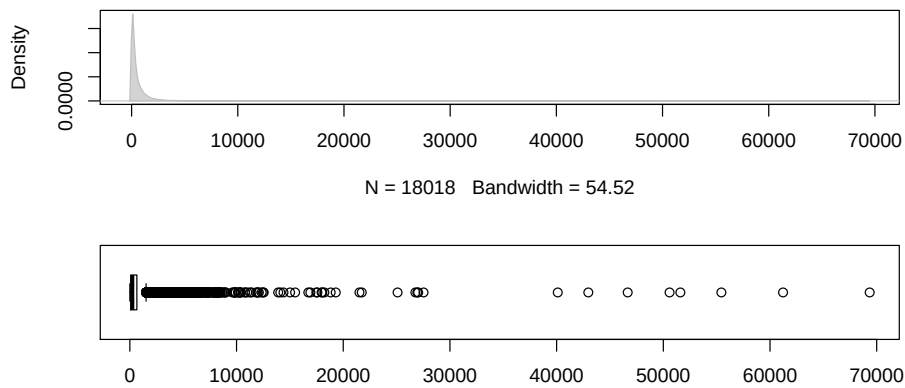


Fig. 3.4 Box-and-whisker plot and kernel density estimates for the absolute errors used by MAE.

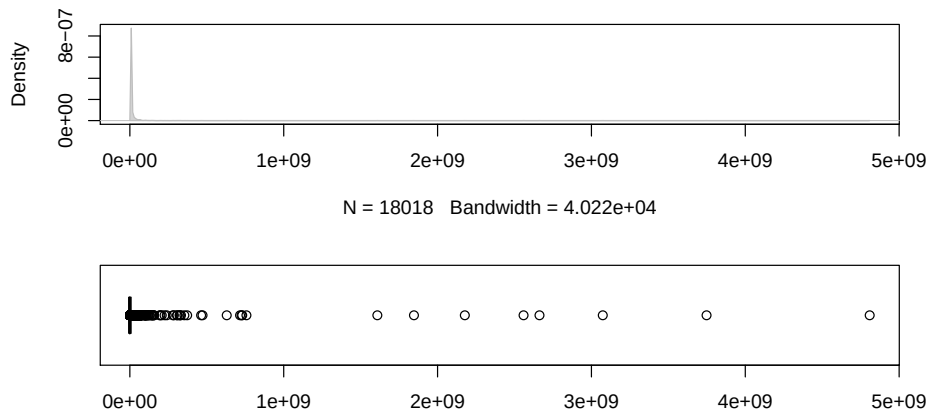


Fig. 3.5 Box-and-whisker plot and kernel density estimates for the squared errors used by RMSE.

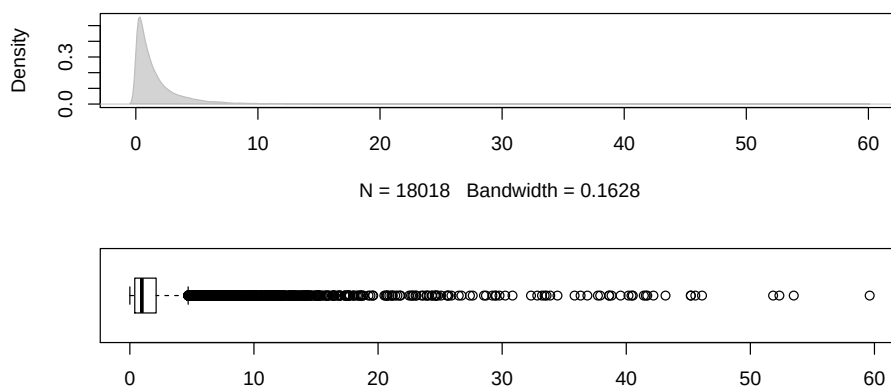


Fig. 3.6 Box-and-whisker plot and kernel density estimates for the absolute scaled errors used by MASE.

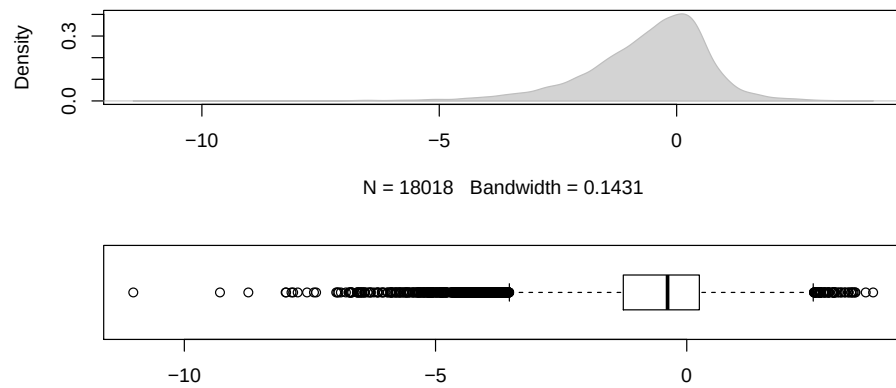


Fig. 3.7 Box-and-whisker plot and kernel density estimates for the absolute scaled errors used by AvgRelMAE (log-scale).

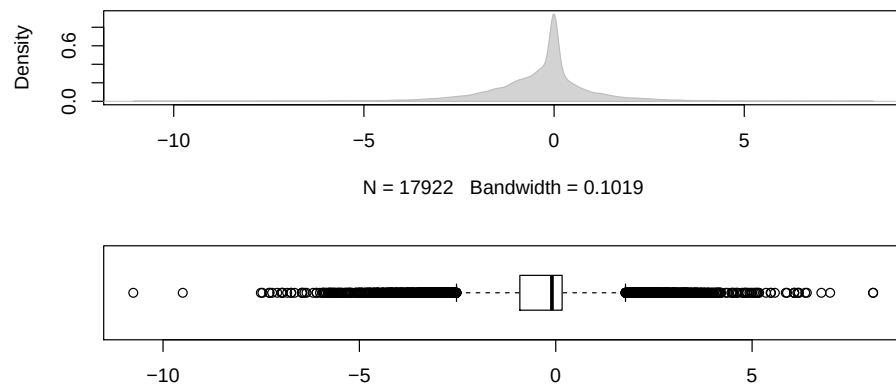


Fig. 3.8 Box-and-whisker plot and kernel density estimates for the relative absolute errors used by MRAE and GMRAE (log-scale, forecasts with zero or undefined error excluded).

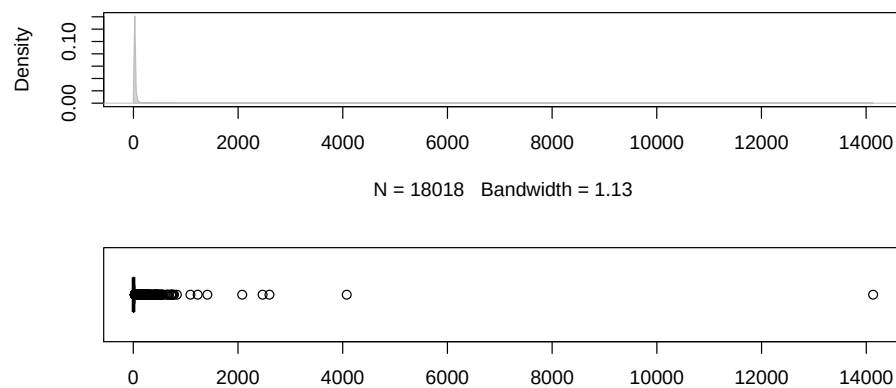


Fig. 3.9 Box-and-whisker plot and kernel density estimates for the absolute percentage errors used by MAPE.

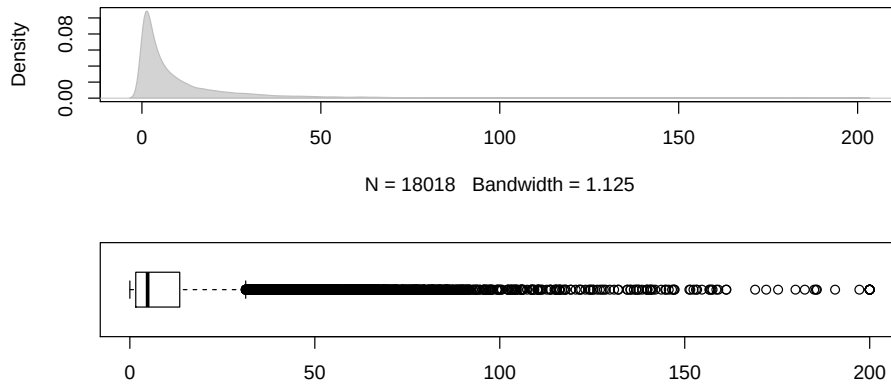


Fig. 3.10 Box-and-whisker plot and kernel density estimates for the scaled percentage errors used by sMAPE.

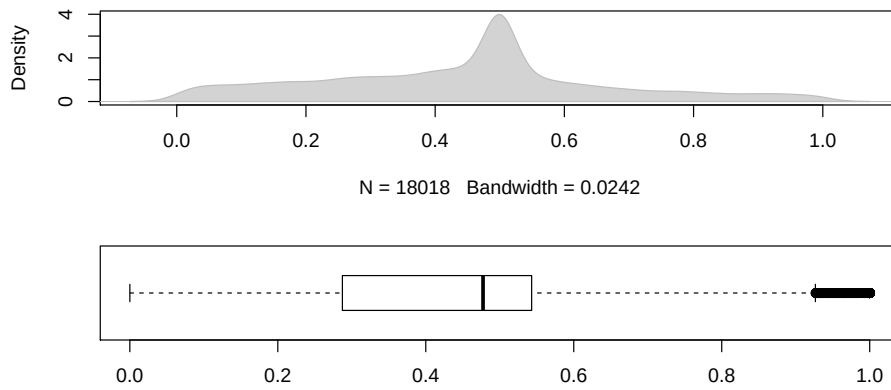


Fig. 3.11 Box-and-whisker plot and kernel density estimates for the bounded relative absolute errors used by UMBRAE (using the naïve errors as the benchmark).

With the geometric mean, GMRAE has shown remarkable resistance to the forecasting outliers. However, one disadvantage of measures based on the geometric mean is that zero-error forecasts have to be excluded. Thus, these measures may not be sufficiently informative. In contrast, due to the use of bounded errors, UMBRAE can perform as well as GMRAE in resisting forecasting outliers. In fact, the errors and rankings given by UMBRAE are highly correlated to which measured by GMRAE, especially in situations (Tables 3.3 and 3.4) where extreme errors are trimmed. Thus, for the cases where measures such as GMRAE are preferred, UMBRAE could be an alternative measure since it is much easier to use without the need to trim errors.

It can also be noticed in Figs 3.4 to 3.11 that all the accuracy measures except AvgRelMAE (see Fig 3.7), GMRAE (see Fig 3.8) and UMBRAE (see Fig 3.11) have highly skewed distributions with long tails including extremely large forecasting outliers. Although undefined and zero errors (0.5%) have been trimmed, GMRAE still contains about 10.2% forecasting outliers including some large log-transformed errors such as -10.76 and 8.08. On the other hand, UMBRAE follows a symmetric distribution, and it only produces about 3% outliers which will not affect the result significantly.

It has to be noted that UMBRAE does not necessarily always provide the same information as GMRAE. For example, given a time series with a million observations, if the forecasting method and the benchmark method produces errors $(y - f)$ which are e and e^* following the standard normal distribution, UMBRAE and GMRAE will both be approximately 1. However, if the forecasting method produces errors of $2e$, the value of GMRAE will be approximately 2 as one may expected. But, UMBRAE will give an error of approximately 1.67 which is less than 2. This is because the bounded error $\frac{|e|}{|e|+|e^*|}$ used by UMBRAE will not be increased too much when error e is doubled for the cases where $|e|$ is much larger than $|e^*|$. In other words, a twice as worse forecast will not be given an error of twice the significance by UMBRAE when the forecast is much worse than most of other forecasts. In fact, this is the key strategy of UMBRAE for resisting outliers. Also, the above expectation of error 2 is based on the estimation by ‘relative average error’. However, it is arguable that the ‘average relative error’ is not

necessarily the same as the ‘relative average error’. This can be more or less reflected by the synthetic test shown in Fig 3.3. More discussions about this will be given later in this section about the scale-independence.

One of the common concerns about an accuracy measure is whether it is symmetric. Two different cases were used to evaluate the property of symmetry for accuracy measures. In our point of view, the first case is about the symmetry in the absolute quantity which concerns whether the same over-estimates and under-estimates can be treated fairly by a measure. As shown in Fig 3.2, only sMAPE is not symmetric in the absolute quantity (due to the asymmetric bounded errors used). This issue has been addressed by UMBRAE with symmetric bounded errors defined. The second case is in fact about the symmetry in the relative quantity where measures are expected to give a result of 1 for averaging two relative errors N and $\frac{1}{N}$. Normally, a measure which uses the arithmetic mean should not be symmetric in such relative quantity. However, UMBRAE, which uses the arithmetic mean for part of its calculations, shows a symmetric result. This is because UMBRAE does not work directly on the original error ratios. The original relative errors have been converted to bounded relative errors for UMBRAE before calculating the arithmetic mean. In fact, this is quite similar to the process of calculating GMRAE which is based on the geometric mean. As a result, it is not an issue for UMBRAE to use the arithmetic mean.

It is necessary (or, at least, highly desirable) for an accuracy measure to be scale-independent when assessing forecasting methods across data on different scales. Normally, measures based on percentages or ratios in the same range are considered to be scale-independent. However, we argue that it is not enough for these percentages or ratios to be in the same range. To be truly scale-independent, these error percentages or ratios should also be closely related to the scale of data for specific observations. Otherwise, they may lead to misleading results. For example, in Table 3.1, the error of MASE for the naïve method is 2.134. This is somewhat a confusing result which may be intuitively interpreted as an indication that the naïve method performs worse than the naïve method itself! In fact, it means the naïve method gives smaller errors on average

for the forecasting data than its errors for the *in-sample* data. In contrast, AvgRelMAE does not have this issue of MASE since it uses the average error on *out-of-sample* as the scaling factor. Fig 3.3 shows that MASE fails to distinguish the difference between the two forecasts which are clearly different considering the error percentages at different observations. This is because every single error used by MASE at different observations is scaled by the same scaling factor. GMRAE also fails in this evaluation. We notice that this is because GMRAE, in fact, has the same issue as MASE. Every single error of GMRAE can also be considered to be a scaled error based on a consistent scaling factor $GMAE^*$, which is the geometric mean of the benchmark errors e^* . According to the above, we conclude that MASE, AvgRelMAE and GMRAE are relatively scale-independent because they assume that the scaling factor is a consistent estimator. In contrast, UMBRAE is scale-independent, and it is closely related to the error ratios at observations. Thus, it can reasonably show the difference between the two forecasts with respect to error percentages.

Another important property of an accuracy measure is its interpretability. As Table 3.1 shows, the numerical errors measured by MAE and RMSE have little intuitive meaning without comparisons and have therefore been scored as ‘fair’. Comparatively, measures which produce errors in percentages or ratios based on a benchmark are more interpretable. The benchmark used by an accuracy measure is also important for its interpretability. In Table 3.1, errors measured by MAPE are all small errors around 10%. However, these small errors are less meaningful without comparisons. This is because these small percentages are based on the original values of observations. Thus, they do not necessarily indicate a good performance. In contrast, errors measured by UMBRAE are more interpretable. An error of 0.77 indicates that the forecasting method performs roughly 23% better than the benchmark method.

As shown in Table 3.5, the accuracy measures are rated by the key criteria concerned in Section 3.2. Measures are considered to be less (‘fair’) informative if undefined or zero errors have to be excluded. The property of symmetry is rated in both absolute quantity and relative quantity as discussed above. Measures are rated as relatively

scale-independent because they assume that the scaling factor is a consistent estimator. Relative-based accuracy measures are considered to be more interpretable than other measures since they can provide more intuitive results in terms of performance without extra comparisons. sMAPE is rated as poor in interpretability since its error, which has a range of $[0, 200]$, is not as easy as MAPE to understand.

Table 3.5 Ratings of Accuracy Measures.

Accuracy Measure	Informative	Resistant to Outliers	Symmetric (absolute,relative)	Scale-independent	Interpretability
MAE	good	fair	yes,no	no	fair
RMSE	good	poor	yes,no	no	fair
MASE	good	fair	yes,no	relatively	good
AvgRelMAE	good	good	yes,yes	relatively	good
MRAE	fair	fair	yes,no	yes	good
GMRAE	fair	good	yes,yes	relatively	good
MAPE	fair	fair	yes,no	yes	fair
sMAPE	good	good	no,no	yes	poor
UMBRAE	good	good	yes,yes	yes	good

In summary, we show that UMBRAE (i) is informative and uses all available errors; (ii) can perform as well as GMRAE in resisting forecasting outliers without the need to trim zero-error forecasts; (iii) is symmetric in both absolute quantity and relative quantity; (iv) is scale-independent; (v) is interpretable and can provide intuitive results. As such, UMBRAE combines the best features of various alternative measures into a single new measure. Thus, we believe UMBRAE is an interesting new measure because it constitutes a simple, flexible, easy to use and understand measure that is resistant to outliers. Also, the forecasting benchmark for calculating UMBRAE is selectable, and the ideal choice should be a forecasting method to be outperformed. As a well-known benchmark, the naïve method can be easily applied as a default to show whether a forecasting method is generally good or not.

3.5 Summary

This chapter proposed a new accuracy measure UMBRAE based on bounded relative errors. As discussed in the review of sMAPE, one advantage of the bounded error is that

it gives less significance to outliers since it does not have the issue of being excessively large or infinite. Evaluation on the proposed measure along with related measures has been made on both synthetic and real-world data. It has been illustrated that UMBRAE combines the best features of various alternative measures without having their common drawbacks. UMBRAE, with selectable benchmark, can provide an informative and interpretable result based on bounded relative error. It is less sensitive to forecasting outliers than other measures. It is also symmetric and scale-independent. Though it has been commonly accepted that there cannot be any single best accuracy measure, we suggest that UMBRAE is a good choice for general use when evaluating the performance of forecasting methods. Since UMBRAE, in our study, performs similar to GMRAE without the need to trim zero-error forecasts, we particularly recommend UMBRAE as an alternative measure for the cases where GMRAE is preferred.

Although we have shown that UMBRAE has many advantages as described above, its statistical properties have not been well studied. For example, the way how UMBRAE reflects the properties of the distributions of errors is unclear. Moreover, one possible underlying drawback for UMBRAE is that the bounded error used by UMBRAE will reach the maximum value 1.0 when the benchmark error ($Y_t - F_t^*$) is equal to zero even if the forecast is good. This may produce a biased estimate especially when the benchmark method produces a large number of zero errors. Although this drawback may not be relevant for the majority of real-world data, we would like to address this issue in the future.

The next chapter presents a direct approach for determining the switch points in the KM algorithm, as an accomplishment of the objective to reduce the computational costs of IT2 ANFIS.

Chapter 4

A Direct Approach for Determining the Switch Points in the Karnik-Mendel Algorithm

4.1 Introduction

The objective of this chapter is to reduce the computational cost of IT2 ANFIS. As mentioned in Section 2.2.2, the KM algorithm is used to compute the centroid of IT2 fuzzy sets, determining the switch points needed for the lower and upper bounds of the centroid, through an iterative process. It is commonly acknowledged that there is no closed-form solution for determining such switch points. Many enhanced algorithms have been proposed to improve the computational efficiency of the KM algorithm. However, all of these algorithms are still based on iterative procedures.

In this chapter, a direct approach based on derivatives for determining the switch points without multiple iterations is proposed, together with mathematical proof to show that these switch points are correct for the lower and upper bounds of the centroid. Experimental simulations show that the direct approach obtains the same switch points, but is more computationally efficient than any of the existing (iterative) algorithms in R.

The rest of this chapter is organised as follows. The mathematical formulation of the KM algorithm is introduced in Section 4.2. Section 4.3 briefly reviews some of the well-known iterative algorithms for the switch points in the KM algorithm. The new direct approach is introduced in Section 4.4, followed by a proof in Section 4.5 to show that the switch points calculated by the direct approach are the same as obtained via the KM algorithm. Experimental results for comparisons are shown and discussed in Sections 4.6 and 4.7. Finally, a summary is provided in Section 4.11.

4.2 The KM algorithm

In this section, we briefly review the KM algorithm.

Let an IT2 fuzzy set $\tilde{A}$ be based on

$$\begin{aligned} x_i &\in X, \quad i = 1, 2, \dots, N \\ J_i &\equiv [\underline{u}_i, \bar{u}_i], \quad 0 \leq \underline{u}_i \leq \bar{u}_i \leq 1 \end{aligned}$$

where x_i is the primary variable in the discrete universe of discourse X (note that x_i is in ascending order for i from 1 to N), J_i represents the membership grade interval for the primary variable x_i , and N is the number of discrete points in the universe of discourse.

For any given embedded T1 fuzzy set, with membership grades $u_i \in J_i$ for all i , of such an IT2 fuzzy set $\tilde{A}$, the centroid is defined as:

$$c = \frac{\sum_{i=1}^N x_i u_i}{\sum_{i=1}^N u_i}.$$

If the above centroid c is computed for all embedded T1 fuzzy sets, a centroid interval $C = [c_l, c_r]$ can be obtained where

$$\begin{aligned} c_l &= \inf_{\forall u_i \in J_i} \frac{\sum_{i=1}^N x_i u_i}{\sum_{i=1}^N u_i} \\ c_r &= \sup_{\forall u_i \in J_i} \frac{\sum_{i=1}^N x_i u_i}{\sum_{i=1}^N u_i} \end{aligned}$$

In fact, there is no need to compute the centroids for all embedded T1 fuzzy sets to get the centroid interval. It is well known that the endpoints c_l and c_r of the centroid interval can also be expressed as (the derivation can be found in [34]):

$$c_l = \frac{\sum_{i=1}^L x_i \bar{u}_i + \sum_{i=L+1}^N x_i \underline{u}_i}{\sum_{i=1}^L \bar{u}_i + \sum_{i=L+1}^N \underline{u}_i} \quad (4.1)$$

$$c_r = \frac{\sum_{i=1}^R x_i \underline{u}_i + \sum_{i=R+1}^N x_i \bar{u}_i}{\sum_{i=1}^R \underline{u}_i + \sum_{i=R+1}^N \bar{u}_i} \quad (4.2)$$

where L and R , which are integer indices in the range of $[1, N-1]$, are the switch points for selecting $\underline{u}_i$ or $\bar{u}_i$.

As stated by Mendel [6], there is no closed-form solution for such switch points L and R , and hence, for c_l and c_r . By utilising the properties of the switch points such that

$$x_L \leq c_l \leq x_{L+1}$$

$$x_R \leq c_r \leq x_{R+1}$$

the KM algorithm can be used to find them iteratively [31].

4.3 The iterative algorithms

In this section, two commonly used iterative algorithms are briefly introduced to establish terminology and notation. Rather than introduce the original KM and IASC algorithms, their enhancements (the EKM and EIASC algorithms) are reviewed as they are more efficient than the original algorithms.

4.3.1 The EKM algorithm

The EKM algorithm is summarised in Table 4.1. Compared to the original KM algorithm, the EKM algorithm introduces a better initialisation (Step 2) for the starting position and a change of the termination condition (Step 4) to remove an unnecessary iteration [34].

The EKM algorithm can save, on average, about two iterations. Simplified calculations (Step 5) are also introduced to save computational costs for each iteration.

4.3.2 The EIASC algorithm

The EIASC algorithm is summarised in Table 4.2. Compared to the original IASC algorithm, the EIASC algorithm introduced a new stopping criterion (Step 4) [35]. Also, the starting point for computing c_r has been changed to N (Step 2) since the switch point R for c_r has been shown to be generally greater than $N/2$ [34].

— intentionally left blank —

Table 4.1 The EKM algorithm for computing the centroid endpoints (c_l and c_r) of an IT2 Fuzzy Set.

Step	The EKM algorithm for computing c_l	The EKM algorithm for computing c_r
1	Sort x_i ($i = 1, 2, \dots, N$) in ascending order and match $\underline{u}_i$ and $\bar{u}_i$ accordingly with their respective x_i .	Sort x_i ($i = 1, 2, \dots, N$) in ascending order and match $\underline{u}_i$ and $\bar{u}_i$ accordingly with their respective x_i .
2	Set $k = \lfloor N/2.4 \rfloor$, which is the nearest integer to $N/2.4$, and compute	Set $k = \lfloor N/1.7 \rfloor$, which is the nearest integer to $N/1.7$, and compute
	$a = \sum_{i=1}^k x_i \bar{u}_i + \sum_{i=k+1}^N x_i \underline{u}_i$ $b = \sum_{i=1}^k \bar{u}_i + \sum_{i=k+1}^N \underline{u}_i$	$a = \sum_{i=1}^k x_i \underline{u}_i + \sum_{i=k+1}^N x_i \bar{u}_i$ $b = \sum_{i=1}^k \underline{u}_i + \sum_{i=k+1}^N \bar{u}_i$
	$c = a/b$	
3	Find $k' \in [1, N-1]$ such that $x_{k'} < c \leq x_{k'+1}$	
4	If $k' = k$, set $c_l = c$ and stop;	If $k' = k$, set $c_r = c$ and stop;
5	Otherwise, go to Step 5; Compute $s = \text{sign}(k' - k)$, and	Otherwise, go to Step 5; Compute $s = \text{sign}(k' - k)$, and
	$a' = a + s \sum_{i=\min(k,k')+1}^{\max(k,k')} x_i (\bar{u}_i - \underline{u}_i)$ $b' = b + s \sum_{i=\min(k,k')+1}^{\max(k,k')} (\bar{u}_i - \underline{u}_i)$	$a' = a + s \sum_{i=\min(k,k')+1}^{\max(k,k')} x_i (\underline{u}_i - \bar{u}_i)$ $b' = b + s \sum_{i=\min(k,k')+1}^{\max(k,k')} (\underline{u}_i - \bar{u}_i)$
	$c' = a'/b'$	
6	Set $c = c'$, $a = a'$, $b = b'$ and $k = k'$. Go to Step 3.	

Note that for the case described in Section 4.2, Step 1 is not necessary since x_i has already been defined in ascending order. Table 4.1 is adapted from [6, 34].

Table 4.2 The EIASC algorithm for computing the centroid endpoints (c_l and c_r) of an IT2 Fuzzy Set.

Step	The EIASC algorithm for computing c_l	The EIASC algorithm for computing c_r
1	Sort x_i ($i = 1, 2, \dots, N$) in ascending order and match $\underline{u}_i$ and $\bar{u}_i$ accordingly with their respective x_i .	
2	Initialise $k = 0$ and	Initialise $k = N$ and
		$a = \sum_{i=1}^N x_i \underline{u}_i$ $b = \sum_{i=1}^N \underline{u}_i$
3	Compute $k = k + 1$	Compute
		$a = a + x_k(\bar{u}_k - \underline{u}_k)$ $b = b + \bar{u}_k - \underline{u}_k$ $c = a/b$
4	If $c \leq x_{k+1}$, set $c_l = c$ and stop;	$k = k - 1$ If $c \geq x_k$, set $c_r = c$ and stop; Otherwise, go to Step 3;

Note that for the case described in Section 4.2, Step 1 is not necessary since x_i has already been defined in ascending order. Table 4.2 is adapted from [6, 35].

4.4 The Direct Approach (DA) algorithm

This section introduces the direct approach for determining the switch points.

An arbitrary c in the centroid interval C , represented as Equation 4.3, can be expanded to Equation 4.4. Equation 4.4 can then be transformed to Equation 4.5 by substituting $x_1 + \sum_{j=2}^i \delta x_j$ for all corresponding $x_i (i > 1)$, where

$$\delta x_j = x_j - x_{j-1}.$$

For example, x_3 in Equation 4.4 is represented as $(x_1 + \delta x_2 + \delta x_3)$ in Equation 4.5. Note that δx_j is always positive since x_i (defined in Section 4.2), and hence x_j , is in ascending order. This transformation is made to allow the sign of the partial derivatives to be easily identified, as shown in Equations 4.7 to 4.11. After the transformation, by rearranging and aggregating all the items with x_1 and δx_j for $j = 2, 3, \dots, N$ in Equation 4.5, c can finally be represented as Equation 4.6. We can then compute the partial derivative of c with respect to u_j in the pattern of Equations 4.7 to 4.8. Then, this partial derivative $\frac{\partial c}{\partial u_j}$ can be represented as Equation 4.9 for specific $j \in [1, N]$.

It is noted by Karnik and Mendel [31] and Wu and Mendel [34] that equating the partial derivative of c with respect to u_j to zero does not give us any information about the value of u_j that maximises or minimises c . However, it can be observed that the sign of the partial derivative of c with respect to u_j does not depend on the value of u_j . Hence, it is clear that, when the partial derivative is negative, it is necessary to take the largest possible value of u_j to minimise c . When the partial derivative is positive, it is necessary to take the smallest possible value of u_j in order to minimise c . That is, to obtain c_l (the minimum centroid), one must set u_j to $\bar{u}_j$ when the partial derivative is negative, and to $\underline{u}_j$ when the partial derivative is positive. We observe that for any given value of u_j , this partial derivative is monotonically increasing with j (from 1 to N). Hence, for c_l , there must be a switch point $k \in [1, N]$ for which $\frac{\partial c}{\partial u_k} \leq 0$ and $\frac{\partial c}{\partial u_{k+1}} \geq 0$. Based on the above, it is possible to find the switch point directly by locating the value of k where the sign of the partial derivative changes. The same principles can be used to find the switch point k

for the maximum centroid c_r , swapping $\bar{u}_j$ and $\underline{u}_j$ in u_j . The DA algorithm is such an algorithm, deploying the derivatives of c with respect to each u_j , and hence providing a method for directly obtaining the switch points. Figure 4.1 illustrates the monotonically increasing trend of the partial derivative $\frac{\partial c}{\partial u_j}$ and the location of the switch point.

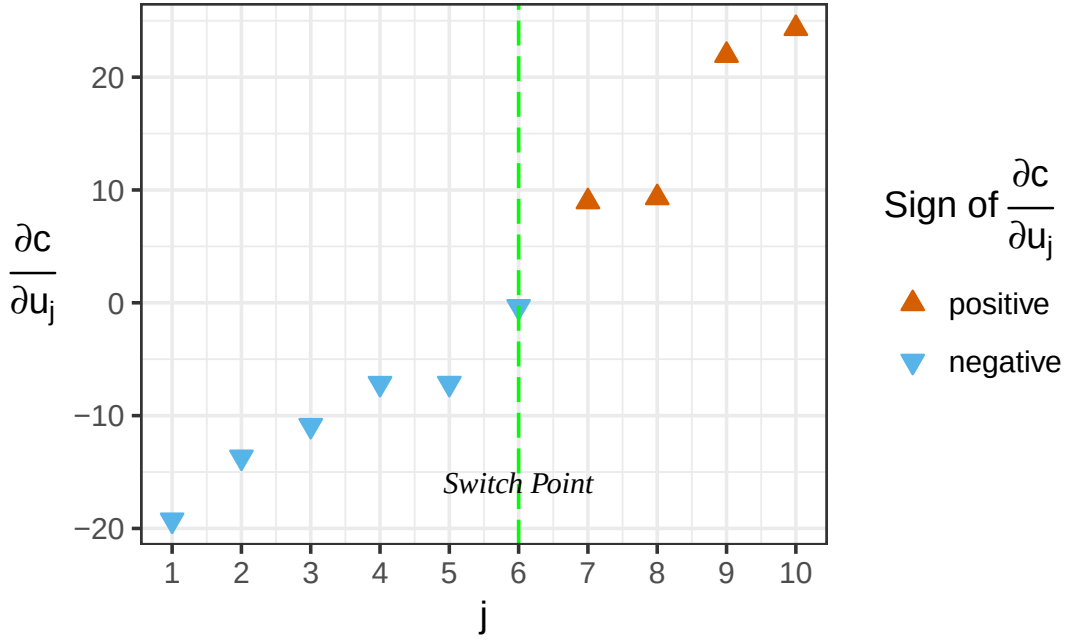


Fig. 4.1 An illustration of the DA algorithm showing the position of the switch point for an arbitrary example where N is 10.

$$c = \frac{\sum_{i=1}^N x_i u_i}{\sum_{i=1}^N u_i} \quad (4.3)$$

$$= \frac{x_1 u_1 + x_2 u_2 + x_3 u_3 \cdots + x_N u_N}{\sum_{i=1}^N u_i} \quad (4.4)$$

$$= \frac{x_1 u_1 + (x_1 + \delta x_2) u_2 + (x_1 + \delta x_2 + \delta x_3) u_3 \cdots + (x_1 + \delta x_2 \cdots + \delta x_N) u_N}{\sum_{i=1}^N u_i} \quad (4.5)$$

$$= x_1 + \frac{(\sum_{i=2}^N u_i) \delta x_2}{\sum_{i=1}^N u_i} + \frac{(\sum_{i=3}^N u_i) \delta x_3}{\sum_{i=1}^N u_i} \cdots + \frac{(\sum_{i=N}^N u_i) \delta x_N}{\sum_{i=1}^N u_i} \quad (4.6)$$

$$\begin{aligned}
\frac{\partial c}{\partial u_j} &= \frac{\partial}{\partial u_j} \left(x_1 + \frac{(\sum_{i=2}^N u_i) \delta x_2}{\sum_{i=1}^N u_i} + \frac{(\sum_{i=3}^N u_i) \delta x_3}{\sum_{i=1}^N u_i} \dots + \frac{(\sum_{i=N}^N u_i) \delta x_N}{\sum_{i=1}^N u_i} \right) \quad (4.7) \\
&= \frac{\partial}{\partial u_j} (x_1) + \frac{\partial}{\partial u_j} \left(\frac{(\sum_{i=2}^N u_i) \delta x_2}{\sum_{i=1}^N u_i} \right) + \frac{\partial}{\partial u_j} \left(\frac{(\sum_{i=3}^N u_i) \delta x_3}{\sum_{i=1}^N u_i} \right) \dots + \frac{\partial}{\partial u_j} \left(\frac{(\sum_{i=N}^N u_i) \delta x_N}{\sum_{i=1}^N u_i} \right) \quad (4.8) \\
&= \left\{ \begin{aligned} &\left(0 - \frac{(\sum_{i=2}^N u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=3}^N u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} - \dots - \frac{(\sum_{i=N-2}^N u_i) \delta x_{N-2}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N-1}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} \right) |j=1 \\ &\left(0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=3}^N u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} - \dots - \frac{(\sum_{i=N-2}^N u_i) \delta x_{N-2}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N-1}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} \right) |j=2 \\ &\left(0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=2}^2 u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=4}^N u_i) \delta x_4}{(\sum_{i=1}^N u_i)^2} - \dots - \frac{(\sum_{i=N-2}^N u_i) \delta x_{N-2}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N-1}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} \right) |j=3 \\ &\vdots \\ &\left(0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=2}^2 u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=3}^3 u_i) \delta x_4}{(\sum_{i=1}^N u_i)^2} - \dots + \frac{(\sum_{i=N-3}^N u_i) \delta x_{N-2}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N-1}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} \right) |j=N-2 \\ &\left(0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=2}^2 u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=3}^3 u_i) \delta x_4}{(\sum_{i=1}^N u_i)^2} - \dots + \frac{(\sum_{i=N-3}^N u_i) \delta x_{N-2}}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=N-2}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} \right) |j=N-1 \\ &\left(0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=2}^2 u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=3}^3 u_i) \delta x_4}{(\sum_{i=1}^N u_i)^2} - \dots + \frac{(\sum_{i=N-3}^N u_i) \delta x_{N-2}}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=N-2}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=N-1}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} \right) |j=N \end{aligned} \right\} \quad (4.9)
\end{aligned}$$

$$pos_j = \begin{cases} 0 & | j = 1 \\ 0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} & | j = 2 \\ 0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=1}^2 u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} & | j = 3 \\ \vdots & \\ 0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=1}^2 u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} \dots + \frac{(\sum_{i=1}^{N-3} u_i) \delta x_{N-2}}{(\sum_{i=1}^N u_i)^2} & | j = N-2 \\ 0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=1}^2 u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} \dots + \frac{(\sum_{i=1}^{N-2} u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} & | j = N-1 \\ 0 + \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} + \frac{(\sum_{i=1}^2 u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} \dots + \frac{(\sum_{i=1}^{N-1} u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} & | j = N \end{cases} \quad (4.10)$$

$$neg_j = \begin{cases} -\frac{(\sum_{i=2}^N u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} \dots - \frac{(\sum_{i=N-1}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} & | j = 1 \\ -\frac{(\sum_{i=3}^N u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} \dots - \frac{(\sum_{i=N-1}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} & | j = 2 \\ -\frac{(\sum_{i=4}^N u_i) \delta x_4}{(\sum_{i=1}^N u_i)^2} \dots - \frac{(\sum_{i=N-1}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} & | j = 3 \\ \vdots & \\ -\frac{(\sum_{i=N-1}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} & | j = N-2 \\ -\frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} & | j = N-1 \\ 0 & | j = N \end{cases} \quad (4.11)$$

Since δx_j is always positive, it can be observed in Equation 4.9 that the partial derivative consists of two parts, which are the positive part pos_j and the negative part neg_j :

$$\frac{\partial c}{\partial u_j} = pos_j + neg_j$$

where pos_j and neg_j are presented in Equations 4.10 and 4.11, which can be summarised as Equations 4.12 and 4.13.

$$pos_j = \begin{cases} 0 & |j=1 \\ \frac{(\sum_{i=1}^1 u_i) \delta x_2}{(\sum_{i=1}^N u_i)^2} \left[+ \frac{(\sum_{i=1}^2 u_i) \delta x_3}{(\sum_{i=1}^N u_i)^2} \dots + \frac{(\sum_{i=1}^{j-1} u_i) \delta x_j}{(\sum_{i=1}^N u_i)^2} \right] & |j \in [2, N] \end{cases} \quad (4.12)$$

$$neg_j = \begin{cases} \left[- \frac{(\sum_{i=j+1}^N u_i) \delta x_{j+1}}{(\sum_{i=1}^N u_i)^2} \dots - \frac{(\sum_{i=N-1}^N u_i) \delta x_{N-1}}{(\sum_{i=1}^N u_i)^2} \right] - \frac{(\sum_{i=N}^N u_i) \delta x_N}{(\sum_{i=1}^N u_i)^2} & |j \in [1, N-1] \\ 0 & |j=N \end{cases} \quad (4.13)$$

$$pos_j = \begin{cases} 0 & |j=1 \\ \left(\sum_{i=1}^1 u_i \right) \delta x_2 \left[+ \left(\sum_{i=1}^2 u_i \right) \delta x_3 \dots + \left(\sum_{i=1}^{j-1} u_i \right) \delta x_j \right] & |j \in [2, N] \end{cases} \quad (4.14)$$

$$neg_j = \begin{cases} \left[- \left(\sum_{i=j+1}^N u_i \right) \delta x_{j+1} \dots - \left(\sum_{i=N-1}^N u_i \right) \delta x_{N-1} \right] - \left(\sum_{i=N}^N u_i \right) \delta x_N & |j \in [1, N-1] \\ 0 & |j=N \end{cases} \quad (4.15)$$

Having all the partial derivatives, the switch points L for c_l and R for c_r can be obtained as follows.

For L and hence c_l :

1. Calculate pos_j by setting all the u_i to be $\bar{u}_i$ and calculate neg_j by setting all the u_i to be $\underline{u}_i$, for all $j \in [1, N]$.
2. Calculate all the partial derivatives $\frac{\partial c}{\partial u_j}$ as $pos_j + neg_j$.
3. Find the smallest $k \in [1, N - 1]$ such that $\frac{\partial c}{\partial u_{k+1}} \geq 0$.
4. If k exists, set $L = k$; otherwise, set $L = N - 1$.
5. Compute c_l by Equation 4.1.

For R and hence c_r :

1. Calculate pos_j by setting all the u_i to be $\underline{u}_i$ and calculate neg_j by setting all the u_i to be $\bar{u}_i$, for all $j \in [1, N]$.
2. Calculate all the partial derivatives $\frac{\partial c}{\partial u_j}$ as $pos_j + neg_j$.
3. Find the largest $k \in [1, N - 1]$ such that $\frac{\partial c}{\partial u_k} \leq 0$.
4. If k exists, set $R = k$; otherwise, set $R = 1$.
5. Compute c_r by Equation 4.2.

It should be noted that the denominator $(\sum_{i=1}^N u_i)^2$ in Equations 4.12 and 4.13, which must be positive, can be neglected in calculations to save computational costs. In other words, without the risk of changing the sign of $\frac{\partial c}{\partial u_j}$, pos_j and neg_j can be simplified as presented in Equations 4.14 and 4.15. Also, in practice, there is no need to calculate the partial derivatives one by one, as cumulative sums and vectorised operations can be used. The pseudo-code for obtaining L and c_l is shown in Algorithm 1. R and c_r are calculated in a similar manner, making the substitutions as described above.

input : $X, \bar{U}, \underline{U}$, vectors of the primary variable, the upper membership grades, and the lower membership grades, respectively; $x_i, \bar{u}_i$, and $\underline{u}_i$ denote elements of the respective vectors;

output : L , the switch point; c_l , the lower bound of the centroid;

- 1 $X' \leftarrow \{x_i - x_{i-1} \mid i = 2, 3, \dots, N\}$, a vector of consecutive differences of elements of X ;
- 2 $S^l \leftarrow \{\sum_{i=1}^j \bar{u}_i \mid j = 1, 2, \dots, N-1\}$, a vector of the cumulative sum of the first $N-1$ elements of $\bar{U}$;
- 3 $S^2 \leftarrow \{\sum_{i=N}^j \underline{u}_i \mid j = N, \dots, 3, 2\}$ a vector of the cumulative sum of the last $N-1$ elements of $\underline{U}$ in the reverse order ;
- 4 $T^P \leftarrow \{x'_i \cdot s_i^l \mid i = 1, 2, \dots, N-1\}$, where x'_i and s_i^l are the i^{th} element of vectors X' and S^l respectively ;
- 5 $T^N \leftarrow \{x'_{N-i} \cdot s_i^2 \mid i = 1, 2, \dots, N-1\}$, where s_i^2 is the i^{th} element of the vector S^2 ;
- 6 $D^P \leftarrow \{0, \sum_{i=1}^j t_i^P \mid j = 1, 2, \dots, N-1\}$, where t_i^P is the i^{th} element of the vector T^P ;
- 7 $D^N \leftarrow \{\sum_{i=1}^j -t_i^N, 0 \mid j = N-1, \dots, 2, 1\}$, where t_i^N is the i^{th} element of the vector T^N ;
- 8 $D \leftarrow \{d_i^P + d_i^N \mid i = 1, 2, \dots, N\}$, where d_i^P and d_i^N are the i^{th} element of vectors D^P and D^N respectively ;
- 9 Find the smallest $k \in 1, 2, \dots, N-1$ such that $d_{k+1} \geq 0$, where d_{k+1} , which represents $\frac{\partial c_l}{\partial u_{k+1}}$, is the $(k+1)^{th}$ element of the vector D ;
- 10 **if** k **exists** **then** $L \leftarrow k$ **else** $L \leftarrow N-1$;
- 11 Compute c_l by Equation 4.1;

Algorithm 1: Pseudo code for obtaining L , the switch point; and c_l , the lower bound of the centroid.

4.5 Proof of the DA algorithm

This section proves that the DA algorithm is correct for determining the switch points.

Proposition 1. It is clear that when $\frac{\partial c}{\partial u_j} < 0$, u_j should be its maximum value $\bar{u}_j$ to minimise c , and it should be its minimum value $\underline{u}_j$ to maximise c . Conversely, when $\frac{\partial c}{\partial u_j} > 0$, u_j should be $\underline{u}_j$ to minimise c and it should be $\bar{u}_j$ to maximise c . When $\frac{\partial c}{\partial u_j} = 0$, c is not dependent on the value of u_j .

Theorem 1. The index L obtained by the DA algorithm, as described in Section 4.4, is the correct switch point for calculating the minimum centroid c_l .

Proof. Given that δx_i is always positive, it is clear (from Equations 4.14 and 4.15) that for any $j \in [1, N]$, pos_j attains its maximum value when u_i is $\bar{u}_i$ and neg_j attains its maximum value when u_i is $\underline{u}_i$; hence $\frac{\partial c}{\partial u_j}$ is also maximal. It can be observed from Equation 4.8 that $\frac{\partial c}{\partial u_j}$ is monotonically increasing in j . This is because the only difference of $\frac{\partial c}{\partial u_j}$ for $j = n - 1$ and $j = n$ is the n^{th} term ($n \in [2, N]$), which is negative for $j = n - 1$ and positive for $j = n$.

If there exists a smallest $k \in [1, N - 1]$ such that $\frac{\partial c}{\partial u_{k+1}} \geq 0$, then k is the correct switch point L (for calculating c_l) on the basis that:

1. L cannot be less than k , for the following reason. For every $j \in [1, k]$, it must be the case that $\frac{\partial c}{\partial u_j} < 0$. Therefore, by Proposition 1, u_j should be its maximum value $\bar{u}_j$ to minimise c . Now suppose L is less than k . Then, as L is the switch point, every $u_{j \in [L+1, k]}$ should be its minimum value $\underline{u}_j$, which by contradiction is not possible.
2. L cannot be greater than k , for the following reason. Assume L is greater than k . Then, as L is the switch point every $u_{j \in [k+1, L]}$ should be its maximum value $\bar{u}_j$. By Proposition 1, it must be the case that $\frac{\partial c}{\partial u_{k+1}} < 0$, in which case u_{k+2} must be changed from $\underline{u}_{k+2}$ to $\bar{u}_{k+2}$. This means $\frac{\partial c}{\partial u_{k+2}}$, if it exists, must also be negative. By induction, it can be deduced that $\frac{\partial c}{\partial u_N}$ must be negative. This is in contradiction

to the fact that $\frac{\partial c}{\partial u_N} \geq 0$. Thus, the assumption is incorrect. Hence, L cannot be greater than k .

If there does not exist a $k \in [1, N-1]$ such that $\frac{\partial c}{\partial u_{k+1}} > 0$, then for every $j \in [1, N-1]$, again it must be the case that $\frac{\partial c}{\partial u_j} < 0$. Also, $\frac{\partial c}{\partial u_N}$ must be 0. Thus, there is no contradiction for the switch point L to be $N-1$.

□

Theorem 2. The index R obtained by the DA algorithm, as described in Section 4.4, is the correct switch point for calculating the maximum centroid c_r .

Proof. The proof is similar to above.

□

4.6 Experimental comparison

To further investigate the performance of the new DA, comparisons in terms of time efficiency between the new approach and two of the most widely used algorithms (the EKM and the EIASC algorithms) were conducted. The platform was a laptop with Intel Core i7-3720QM Quad-core CPU @ 2.60GHz and 8GB memory, running Windows 7 Professional 64bit Service Pack 1. The programming language and software environment is R x64 version 3.2.3. Computational costs were measured by the user time returned by the built-in function *proc.time()* in the R environment.

4.6.1 Examples of IT2 fuzzy sets

In this section, three example IT2 fuzzy sets are used to verify the DA algorithm by comparing the switch points with the EKM algorithm and the EIASC algorithm. Specifically, the vector X , containing x_i , has 101 discrete values from 0 to 10 by a step size of 0.1. $\bar{u}_i$ and $\underline{u}_i$ are obtained by the following membership functions for each type of example IT2 fuzzy sets.

Symmetric Gaussian membership functions with uncertain deviation

$$\begin{aligned}\bar{u}_i &= \exp\left(-0.5\left(\frac{x_i-5}{1.75}\right)^2\right) \\ \underline{u}_i &= \exp\left(-0.5\left(\frac{x_i-5}{0.25}\right)^2\right)\end{aligned}$$

Gaussian upper membership function and triangular lower membership function

$$\begin{aligned}\bar{u}_i &= \begin{cases} \exp\left(-0.5\left(\frac{x_i-2}{5}\right)^2\right) & | 0 \leq x_i \leq 7.185 \\ \exp\left(-0.5\left(\frac{x_i-9}{1.75}\right)^2\right) & | 7.185 < x_i \leq 10 \end{cases} \\ \underline{u}_i &= \begin{cases} \frac{0.6(x_i+5)}{19} & | 0 \leq x_i \leq 2.6 \\ \frac{0.4(14-x_i)}{19} & | 2.6 < x_i \leq 10 \end{cases}\end{aligned}$$

Piecewise Gaussian membership functions

$$\begin{aligned}\bar{u}_i &= \max \begin{cases} \exp\left(-\left(\frac{x_i-3}{\sqrt{8}}\right)^2\right) \\ 0.8 \exp\left(-\left(\frac{x_i-6}{\sqrt{8}}\right)^2\right) \end{cases} \\ \underline{u}_i &= \max \begin{cases} 0.5 \exp\left(-\left(\frac{x_i-3}{\sqrt{2}}\right)^2\right) \\ 0.4 \exp\left(-\left(\frac{x_i-6}{\sqrt{2}}\right)^2\right) \end{cases}\end{aligned}$$

As shown in Table 4.3, the switch points L and R obtained with the examples by three algorithms are all the same.

Table 4.3 **The switch points L and R obtained by three algorithms based on the example fuzzy sets.**

	Example 1		Example 2		Example 3	
	L	R	L	R	L	R
DA	36	65	27	69	32	57
EKM	36	65	27	69	32	57
EIASC	36	65	27	69	32	57

4.6.2 Generalised IT2 fuzzy sets

In this section, two types of generalised IT2 fuzzy sets are used for the comparisons of time efficiency of DA, EKM and EIASC.

Generalised bell-shaped IT2 fuzzy sets

It was assumed that the vector X , containing x_i , is uniformly distributed from 0 to 10. $\bar{u}_i$ and $\underline{u}_i$ are defined by generalised bell-shaped function:

$$\bar{u}_i = \frac{1}{1 + \left(\left(\frac{x_i - c}{\bar{a}} \right)^2 \right)^b}$$

$$\underline{u}_i = \frac{1}{1 + \left(\left(\frac{x_i - c}{\underline{a}} \right)^2 \right)^b}$$

where $\underline{a}$ and b are randomly selected between 1 and 2; $\bar{a}$ is the multiplication of $\underline{a}$ with a random number between 1 and 2; c is a random number between 0 and 10.

Generalised randomly-shaped IT2 fuzzy sets

This experimental comparison was designed to be similar to the first comparison in [35]. It was assumed that vectors X and $\bar{U}$, containing x_i and $\bar{u}_i$ respectively, are uniformly

distributed from 0 to 1. $\underline{u}_i$ is the multiplication of $\bar{u}_i$ with a random number between 0 and 1.

Comparisons are made for the above two types of IT2 fuzzy sets separately. In each comparison, N , which is the length of X , is set to be between 10 and 2000 with a step size of 10 (giving 200 different values of N). For each value of N , 5000 Monte Carlo simulations were made and the computational time costs were aggregated to be compared for each algorithm.

In all the 2×10^6 simulations, the DA algorithm gave the same switch points as those given by the KM and the EIASC algorithms. Computational time comparisons are shown in Figures 4.2 and 4.3. It can be observed that the three algorithms compared are similar when N is very small. It should be noted that the EIASC algorithm is shown to be better than the EKM algorithm when N is smaller than 1000 [35]. However, in our experiments, the EIASC algorithm is only more efficient than other algorithms when N is around 10. In contrast, DA clearly outperforms other algorithms when N is larger than 100 regardless of the shape of fuzzy sets.

4.7 Discussion

As can be observed in Figures 4.2 and 4.3, the computational time of the EIASC algorithm increases most rapidly among the three algorithms. As has already been discussed, this is because the number of switch points to be evaluated for the EIASC algorithm increases along with the increase of N . In other words, the number of iterations for the EIASC algorithm increases rapidly. In contrast, the EKM algorithm can achieve its final result in two to six iterations, regardless of N . Thus, the computational time for the EKM algorithm increases less significantly than the EIASC algorithm. However, the computational time does increase linearly because the size of the vectors involved in the computing process increases along with the increase of N .

Regardless of the shape of fuzzy sets, our newly proposed DA algorithm performs the best among the three algorithms, although it can be considered as a brute-force approach

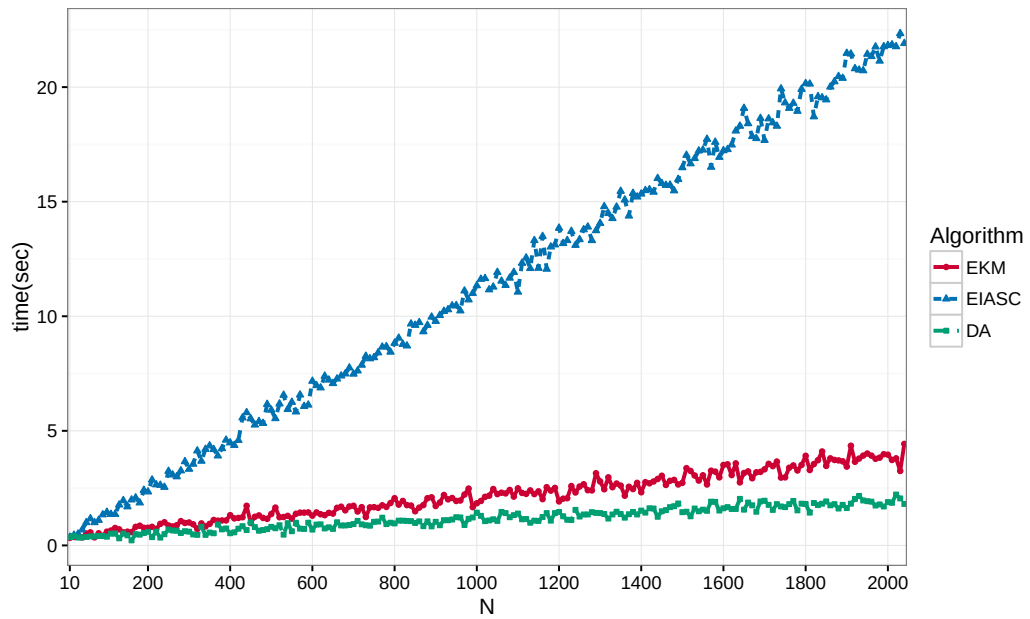


Fig. 4.2 A comparison of computational time costs for different algorithms based on generalised bell-shaped IT2 fuzzy sets described in Section 4.6.

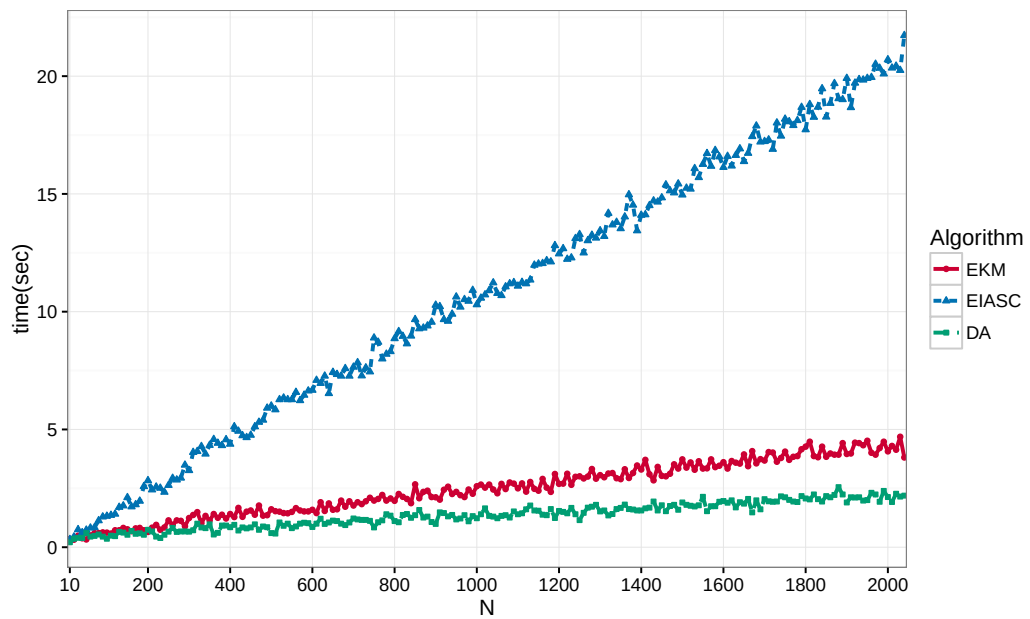


Fig. 4.3 A comparison of computational time costs for different algorithms based on generalised randomly-shaped IT2 fuzzy sets described in Section 4.6.

since all the partial derivatives have to be computed before locating the switch points. However, vectorised operations and the use of cumulative sum make the computing of

partial derivatives quite simple. Thus, in R, the DA algorithm is still competitive for small values of N and clearly outperforms EKM and EIASC for $N \gtrsim 100$.

Note that the above results were made in R. In the following sections, more comparisons are made in Matlab. Also, a platform-free comparison based on the number of computations of the algorithms will be provided.

4.8 Further Investigation

Thanks to Prof. Jerry Mendel and Dr. Dongrui Wu for pointing out the ranking difference of the EIASC algorithm in [98] and [35]. An investigation is made into this, and it can be confirmed that both the results in R and Matlab are valid for the EIASC algorithm. During the investigation, it is found that based on current implementations of the algorithms (EIASC, EKM, and DA) in these papers, it is true that EIASC performs the best in Matlab. Moreover, it is also true that EIASC performs the worst in R. This is mainly due to the efficiency difference of functions or operations in R and Matlab. For example, loop operations (iterations), on which the EIASC algorithm heavily relies, are much less efficient in R than they are in Matlab. It is also found that, in both R and Matlab, operations on vectors with indices are much slower than operations on the same vectors without indices. Note that the DA algorithm in [98] has many vector operations with indices. These operations have made the DA algorithm slower even though DA has been shown to be the fastest in R. It has also to be mentioned that the implementation of the EKM algorithm in [98] has some extra operations with the aim to avoid the issue of infinite loops. Those extra operations have made the EKM algorithm less efficient than it should have been. According to the above, we have learned a lesson that the computational efficiency of an algorithm is closely related to the platform, and how it is implemented.

4.9 Further Optimisation

In this section, we make our best effort to optimise the implementation of each algorithm by trying to avoid or reduce the number of inefficient operations. We also introduce a new direct approach without using the derivatives that used in the DA algorithm.

4.9.1 The EKM algorithm

The EKM algorithm is summarised in Table 4.1. As mentioned in [35], there could be numerical issues or potential infinite loops for the EKM algorithm. Such issue can be prevented by preprocessing steps (see Appendix A in [35]) or extra checks. However, it should be careful that these extra steps, especially when they are not well designed, may make the EKM significantly slower. In the following comparisons, we have taken out the unnecessary checks that were added for the implementation of EKM in [98].

4.9.2 An enhanced DA algorithm

In this section, we introduce an enhanced DA algorithms, called DA*. For the original DA algorithm in Section 4.4, there are many vector operations based on indices. It is found that most of these operations can be optimised by eliminating the use of indices. For example, calculating a vector of $\sum_{i=j}^N u_i$ for j from 2 to N can be transformed to calculating $\sum_{i=1}^N u_i - \sum_{i=1}^j u_i$ for j from 1 to $N - 1$. In R, the above transformation can be represented as from Equation 4.16 to 4.17. In practice, Equation 4.17 can be further simplified as Equation 4.18 where there is an extra zero at the tail of the desired vector. This can eliminate or at least postpone the use of indices until it is necessary.

$$\text{cumsum}(U[N : 2])[(N - 1) : 1] \quad (4.16)$$

$$\text{sum}(U) - \text{cumsum}(U)[1 : (N - 1)] \quad (4.17)$$

$$\text{sum}(U) - \text{cumsum}(U) \quad (4.18)$$

DA can be further optimised by eliminating unnecessary calculations. Note that the partial derivative $\frac{\partial c}{\partial u_j}$ can be represented as Equation 4.19. By transformation, c can be expressed as Equation 4.20. This means, after getting the derivatives, there is no need to calculate c by Equation 4.1 or 4.2 in Step 5 of DA. The computational complexity can be reduced by obtaining c with Equation 4.20.

$$\frac{\partial c}{\partial u_j} = \frac{x_j - c}{\sum_{i=1}^N u_i} \quad (4.19)$$

$$c = x_j - \frac{\partial c}{\partial u_j} \sum_{i=1}^N u_i \quad (4.20)$$

4.9.3 A non-derivative based DA algorithm

The non-derivative based DA algorithm (DAND), in essence, is a brute force method to calculate all possible values of c_l or c_r in Equations 4.1 and 4.2 for all possible values of L and R from 1 to $N - 1$.

Note that Equations 4.1 and 4.2 are equivalent to Equations 4.21 and 4.22, which does not contain vector operations with indices. By the cumulative summation, the computational cost can be saved to calculate c_l and c_r for all L and R from 1 to $N - 1$.

$$c_l = \frac{\sum_{i=1}^N x_i \underline{u}_i + \sum_{i=1}^L x_i (\bar{u}_i - \underline{u}_i)}{\sum_{i=1}^N \underline{u}_i + \sum_{i=1}^L (\bar{u}_i - \underline{u}_i)} \quad (4.21)$$

$$c_r = \frac{\sum_{i=1}^N x_i \bar{u}_i - \sum_{i=1}^R x_i (\bar{u}_i - \underline{u}_i)}{\sum_{i=1}^N \bar{u}_i - \sum_{i=1}^R (\bar{u}_i - \underline{u}_i)} \quad (4.22)$$

The DAND algorithm can be summarised in Table 4.4.

Table 4.4 The DAND algorithm for computing the centroid endpoints (c_l and c_r) of an IT2 Fuzzy Set.

Step	The DAND algorithm for computing c_l	The DAND algorithm for computing c_r
1	Sort x_i ($i = 1, 2, \dots, N$) in ascending order and match $\underline{u}_i$ and $\bar{u}_i$ accordingly with their respective x_i .	
2	Calculate	Calculate
	$a = \sum_{i=1}^N x_i \underline{u}_i$	$a = \sum_{i=1}^N x_i \bar{u}_i$
	$b = \sum_{i=1}^N \underline{u}_i$	$b = \sum_{i=1}^N \bar{u}_i$
3	Calculate the vector U , that is	
	$U = \{\bar{u}_i - \underline{u}_i\}$ for i from 1 to N	$U = \{\underline{u}_i - \bar{u}_i\}$ for i from 1 to N
4	Calculate vectors A , B and C , that are,	Calculate vectors A , B and C , that are,
	$A = a + \text{cumsum}(XU)$	$A = a + \text{cumsum}(XU)$
	$B = b + \text{cumsum}(U)$	$B = b + \text{cumsum}(U)$
	$C = A/B$	$C = A/B$
5	$c_l = \min(C)$	$c_r = \max(C)$

Note that for the case described in Section 4.2, Step 1 is not necessary since x_i has already been defined in ascending order.

Table 4.5 **The computational complexity of EKM based on the number of calculations and comparisons for the core part of its implementations for c_l .**

Step	Pseudo Code	Calculations per Iteration	Comparisons per Iteration	Iterations
1	$k = \lceil N/2.4 \rceil$	1	0	1
2	$a = \sum_{i=1}^k x_i \bar{u}_i + \sum_{i=k+1}^N x_i \underline{u}_i$	$2N$	0	1
3	$b = \sum_{i=1}^k \bar{u}_i + \sum_{i=k+1}^N \underline{u}_i$	N	0	1
4	$c = a/b$	1	0	1
5	Find $k' \in [1, N-1]$ such that $x_{k'} < c \leq x_{k'+1}$	0	N	$m+1$
6	If $k' = k$, set $c_l = c$ and stop;	0	1	$m+1$
7	$s = \text{sign}(k' - k)$	1	0	m
8	$a' = a + s \sum_{i=\min(k,k')+1}^{\max(k,k')} x_i (\bar{u}_i - \underline{u}_i)$	$3 k - k' + 1$	0	m
9	$b' = b + s \sum_{i=\min(k,k')+1}^{\max(k,k')} (\bar{u}_i - \underline{u}_i)$	$ k - k' + 1$	0	m
10	$c' = a'/b'$	1	0	m
11	Set $c = c'$, $a = a'$, $b = b'$ and $k = k'$. Go to Step 5;	0	0	m

Note that the total number of calculations from Steps 6 to 9 is approximately $4 \lfloor \lceil N/2.4 \rceil - L \rfloor$, regardless of m . Hence, the total number of calculations for EKM is $3N + 4 \lfloor \lceil N/2.4 \rceil - L \rfloor$, and the total number of comparisons is $(1 + m)N$.

4.9.4 Computational complexity

The computational complexity of each algorithm in this chapter is listed from Tables 4.5 to 4.8, and summarised in Table 4.9. The computational complexity is calculated based on the number of operations including floating point operations and comparisons. Note that the total numbers are approximations. Constant values are omitted when calculating the totals. For example, $2N - 1$ is considered to be $2N$. As can be seen from the results in Table 4.9, the asymptotic time complexity of all these algorithms is linear $O(N)$ in terms of the Big-O complexity.

Table 4.6 **The computational complexity of EIASC based on the number of calculations and comparisons for the core part of its implementations for c_l .**

Step	Pseudo Code	Calculations per Iteration	Comparisons per Iteration	Iterations
1	$k = 0$	0	0	1
2	$a = \sum_{i=1}^N x_i u_i$	$2N$	0	1
3	$b = \sum_{i=1}^N u_i$	N	0	1
4	$k = k + 1$	1	0	L
5	$u_k = \bar{u}_k - u_k$	1	0	L
6	$a = a + x_k u_k$	2	0	L
7	$b = b + u_k$	1	0	L
8	$c = a/b$	1	0	L
9	If $c \leq x_{k+1}$, set $c_l = c$, $L = k$ and stop; Otherwise, go to Step 4;	0	1	L

The total number of calculations is $3N + 6L$, and the total number of comparisons is L .

Table 4.7 **The computational complexity of DA based on the number of calculations and comparisons for the core part of its implementations for c_l .**

Step	Pseudo Code	Calculations per Iteration	Comparisons per Iteration	Iterations
1	$X' \leftarrow \{x_i - x_{i-1} \mid i = 2, 3, \dots, N\}$	N	0	1
2	$S^1 \leftarrow \{\sum_{i=1}^j \bar{u}_i \mid j = 1, 2, \dots, N-1\}$	N	0	1
3	$S^2 \leftarrow \{\sum_{i=N}^j u_i \mid j = N, \dots, 3, 2\}$	N	0	1
4	$T^P \leftarrow \{x'_i \cdot s_i^1 \mid i = 1, 2, \dots, N-1\}$	N	0	1
5	$T^N \leftarrow \{x'_{N-i} \cdot s_i^2 \mid i = 1, 2, \dots, N-1\}$	N	0	1
6	$D^P \leftarrow \{0, \sum_{i=1}^j t_i^P \mid j = 1, 2, \dots, N-1\}$	N	0	1
7	$D^N \leftarrow \{\sum_{i=1}^j -t_i^N, 0 \mid j = N-1, \dots, 2, 1\}$	N	0	1
8	$D \leftarrow \{d_i^P + d_i^N \mid i = 1, 2, \dots, N\}$	N	0	1
9	Find the smallest $k \in 1, 2, \dots, N-1$ such that $d_{k+1} \geq 0$ if k exists then $L \leftarrow k$ else $L \leftarrow N-1$;	0	N	1
10	Compute c_l by Equation 4.1	$3N$	0	1

The total number of calculations and comparisons for DA are $11N$ and N respectively. Pseudo code is taken from Algorithm 1, where detailed explanations can be found. For DA*, $2N$ calculations can be saved when computing c_l by Equation 4.20 in Step 10. But, in Step 2, the use of Equation 4.18 (for efficiency in R and Matlab) adds an extra N calculations. Hence, The total number of calculations for DA* is $10N$, and it can be simplified to $9N$.

Table 4.8 The computational complexity of DAND based on the number of calculations and comparisons for the core part of its implementations for c_l .

Step	Pseudo Code	Calculations per Iteration	Comparisons per Iteration	Iterations
1	$a = \sum_{i=1}^N x_i \underline{u}_i$	$2N$	0	1
2	$b = \sum_{i=1}^N \underline{u}_i$	N	0	1
3	$U \leftarrow \{\bar{u}_i - \underline{u}_i \mid i = 1, 2, \dots, N\}$	N	0	1
4	$A = a + \text{cumsum}(XU)$	$3N$	0	1
5	$B = b + \text{cumsum}(U)$	$2N$	0	1
6	$C = A/B$	N	0	1
7	$c_l = \min(C)$	0	N	1

Note that the calculations for Steps 4 and 5 can be simplified to $2N$ and N respectively. This can be achieved by adding a and b to the first element of XU and U respectively before calculating the cumulative sum. Hence, the total number of calculations is $10N$, or simplified to $8N$. The total number of comparisons is N .

Table 4.9 A summary of the computational complexity of different algorithms based on the number of calculations and comparisons for the core part of their implementations for c_l .

	Calculations	Comparisons
EKM	$3N + 4 \lfloor [N/2.4] - L \rfloor$	$(1 + m)N$
EIASC	$3N + 6L$	L
DA	$11N$	$1N$
DA*	$10N$ or $9N$	$1N$
DAND	$10N$ or $8N$	$1N$

L is the index of the switch point for c_l , and m is the number of iterations for the EKM algorithm. Note that the calculation complexity for DA* and DAND can be simplified to $9N$ and $8N$ respectively. However, it does not reduce the computational time clearly since operations with indices are required for the simplification.

4.9.5 Computational time

Computational time comparisons of optimised algorithms were made in both R and Matlab, based on two generalised fuzzy sets in 4.6.2. The platform was a laptop with Intel Core i7-3720QM CPU @ 2.60GHz and 8GB memory, running Windows 7 Professional 64bit Service Pack 1. The programming languages and software environment are R x64 version 3.3.3 and Matlab R2017a. Computational costs were measured by the user time returned by the built-in function *proc.time()* in R, and by functions *tic* and *toc* in Matlab.

Results of the comparisons in R can be seen in in Figs. 4.4 to 4.7. As shown in Figs. 4.4 and 4.5, EIASC still performs worst in R among all algorithms. It is interesting that the brute force approach DAND is the most efficient among all in R. The rankings in Matlab are similar except EIASC, as can be observed in Figs. 4.8 and 4.9.

4.10 Further Discussion

A key difference between EIASC and EKM is that how to select next potential solution to the switch point for each new iteration. For EKM, it requires a search which obviously costs more time. In EIASC, it is quite straightforward to the neighbour index. Note that such strategy in EIASC is in fact a ‘double-edged sword’. It reduces the complexity in finding the next solution, but makes EIASC an algorithm which heavily relies on loops. In fact, the use of loops is commonly not the first choice for efficient programming. We found that loops are much less efficient in R than they are in Matlab. This is the key reason for the unsatisfactory performance of EISAC in [98].

As mentioned in Table 4.9, the number of calculations for DA* and DAND can be simplified to $9N$ and $8N$ respectively. However, it does not reduce the computational time clearly since operations with indices are required for the simplification. In practice, the computational complexity is not necessarily the only factor for the efficiency of algorithms, especially when they are on the same level of computational complexity (e.g. $O(N)$). For example, it should be noted that the efficiency of an algorithm is closely related to its implementation in practice. Also, the execution platform is also

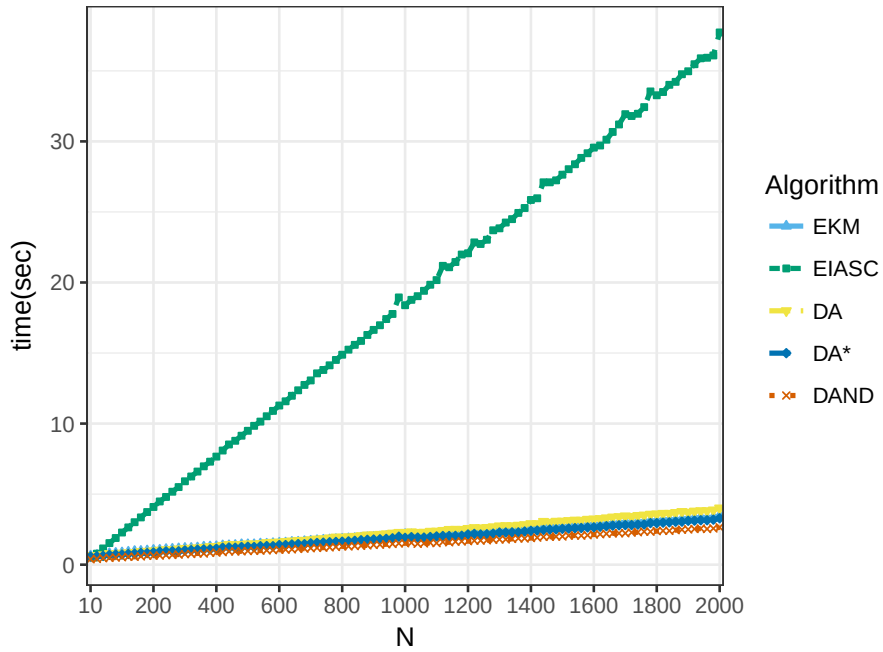


Fig. 4.4 A comparison of optimised algorithms based on bell-shaped fuzzy sets in R.

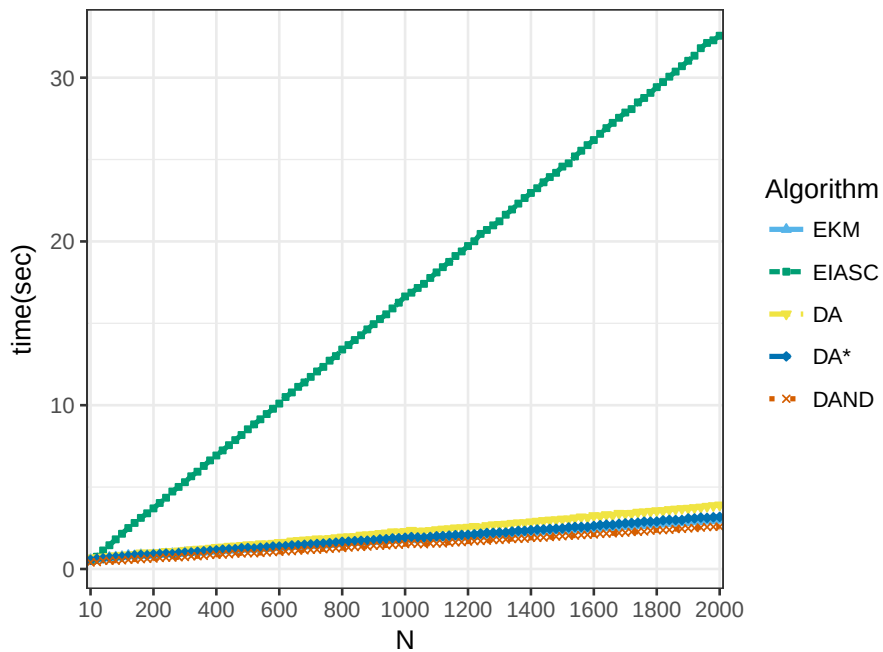


Fig. 4.5 A comparison of optimised algorithms based on random-shaped fuzzy sets in R.

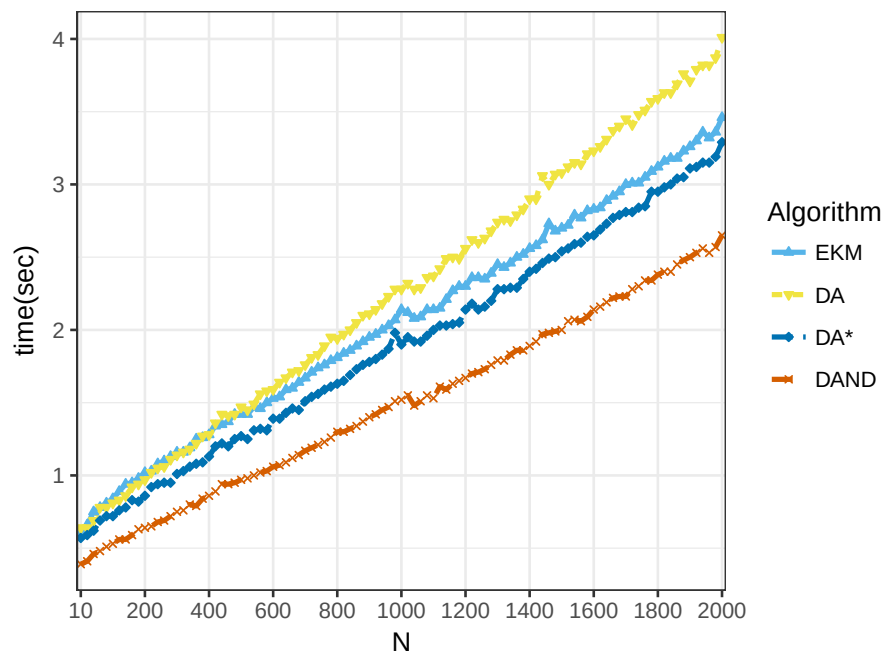


Fig. 4.6 A comparison of optimised algorithms (excluding EIASC) based on bell-shaped fuzzy sets in R.

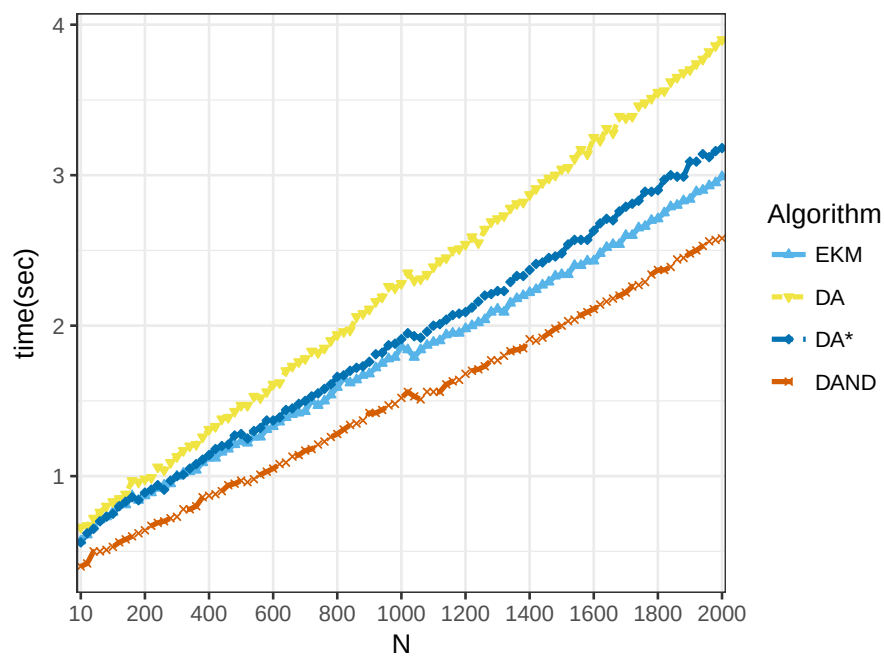


Fig. 4.7 A comparison of optimised algorithms (excluding EIASC) based on random-shaped fuzzy sets in R.

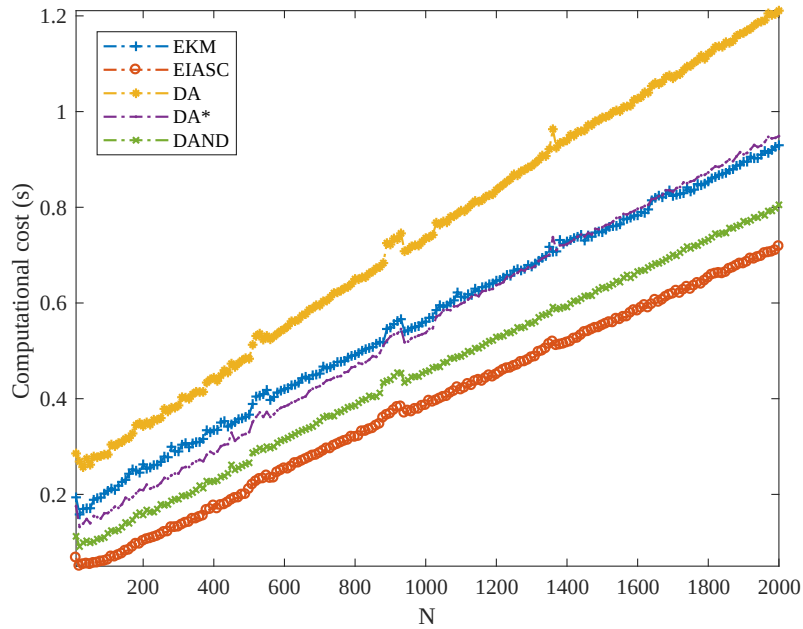


Fig. 4.8 A comparison of optimised algorithms based on bell-shaped fuzzy sets in Matlab.

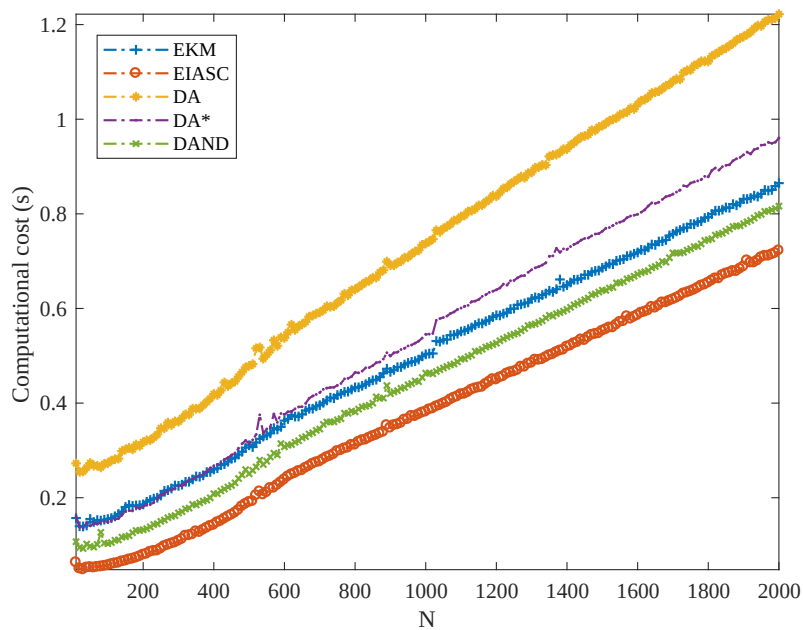


Fig. 4.9 A comparison of optimised algorithms based on random-shaped fuzzy sets in Matlab.

important for the efficiency of algorithms. As can be observed in the results above, EIASC performs the best in Matlab, but the worst in R.

Based on the above, we have learned a lesson that one should be very careful to make a claim of best algorithm to use in terms of computational efficiency since there are many factors which may cause significant different results.

4.11 Summary

In this chapter, a direct approach based on derivatives for determining the switch points for calculating the centroid of an IT2 fuzzy set has been introduced. A derivation of the algorithm, psuedo-code for calculating the switch points, and a mathematical proof of correctness of the switch points have been given for the proposed approach. By empirical simulations, it has been shown that DA is superior to all other iterative algorithms (including the EKM and the EIASC) in terms of computational efficiency in R. It should be noted that the DA algorithm is, in fact, a brute force method which requires the calculations of all partial derivatives. While it can be noted that the partial derivatives are in ascending order and the switch points are located where the sign of partial derivatives changes, it would be interesting for an approach to find the switch points without calculating all the partial derivatives.

Since it has been pointed out by Prof Jerry Mendel that the performance of EIASC in Section 4.6 is contradicted to the results in [35], where the EIASC algorithm was the best in Matlab. A further investigation has been made into this, and it can be confirmed that both the results in R and Matlab are valid for the EIASC algorithm. It has been found that many operations behave differently in R and Matlab. It should be noted that the efficiency of an algorithm is closely related to its implementation in practice.

In summary, we have contributed a DA algorithm based on derivatives for determining the switch points for calculating the lower and upper bounds of the centroids of an IT2 fuzzy set. We have also introduced a new direct approach DAND, which is not based on derivatives. Compared to other algorithms, DAND ranked first in R and

second only to EIASC in Matlab in terms of the computational time based on Monte Carlo simulations. We would cautiously suggest EIASC and DAND in Matlab and R respectively for the use of calculating the centroid of an IT2 fuzzy set.

In the next chapter, based on the direct approaches proposed above, an extended ANFIS architecture which can be used for both T1 and IT2 models is presented.

Chapter 5

An Extended ANFIS Architecture and its Learning Properties for Type-1 and Interval Type-2 Models

5.1 Introduction

In this chapter, an extended novel ANFIS architecture is proposed to make it easy to implement T1 and IT2 ANFIS models, which is one of the objectives mentioned in Chapter 1. By incorporating an extra layer for the fuzzification process, the extended six-layer architecture is able to fit both T1 and IT2 models. Note that the direct approaches proposed in the previous chapter can be applied to the extended ANFIS architecture to improve computational efficiency.

As mentioned in Section 2.2.3, it is still an open question whether the iterative process of estimating the consequent parameters by the LSE method can be successful for IT2 ANFIS. The learning properties of the proposed architecture based on the least-squares estimate method are studied on selected T1 and IT2 ANFIS models. We show that the least-squares estimate method, in general, behaves differently for IT2 ANFIS models compared to T1 ANFIS models, producing larger errors for IT2 ANFIS.

The rest of this chapter is organised as follows. The six-layer extended architecture of ANFIS is presented in Section 5.2, where the operations in each layer are introduced for T1 and IT2 ANFIS separately. Section 5.3 discusses the learning algorithms of ANFIS and in particular the inappropriateness of LSE for IT2 ANFIS. By practical evaluations, the successfulness of LSE as the learning algorithm for T1 and IT2 ANFIS is further investigated in Section 5.4. The results are then discussed in Section 5.5. Finally, a summary of this chapter is given in Section 5.6.

5.2 An extended architecture of ANFIS

In the original five-layer architecture (see Fig. 2.8) of ANFIS presented in [5], only singleton fuzzification is considered, and there is no illustration of the fuzzification process of crisp inputs. Also, the inputs of nodes in *layer 1* and *layer 4* are not clearly defined. Fig. 5.1 shows an extended six-layer architecture of ANFIS which makes both the fuzzification process and the node inputs more explicit. The first layer in the original architecture has been split into two layers (*layer 0* and *layer 1*) in the extended architecture. In the extended architecture, nodes in *layer 0* are fuzzifiers that map crisp inputs into fuzzy sets. Afterwards, the membership grades of crisp inputs can be obtained in *layer 1*. The use of *layer 0* makes the fuzzification process clearer, especially when non-singleton fuzzification is applied. Also, it makes the extended architecture of ANFIS suitable for both T1 and IT2 ANFIS.

In an example of the extended ANFIS architecture, there are four fuzzy inferencing rules (TSK type) with two inputs (x_1, x_2) and one output (y). Specifically, the inferencing rules are:

IF x_1 is A_1 **AND** x_2 is A_3 , **THEN** f_1 ;

IF x_1 is A_1 **AND** x_2 is A_4 , **THEN** f_2 ;

IF x_1 is A_2 **AND** x_2 is A_3 , **THEN** f_3 ;

IF x_1 is A_2 **AND** x_2 is A_4 , **THEN** f_4 .

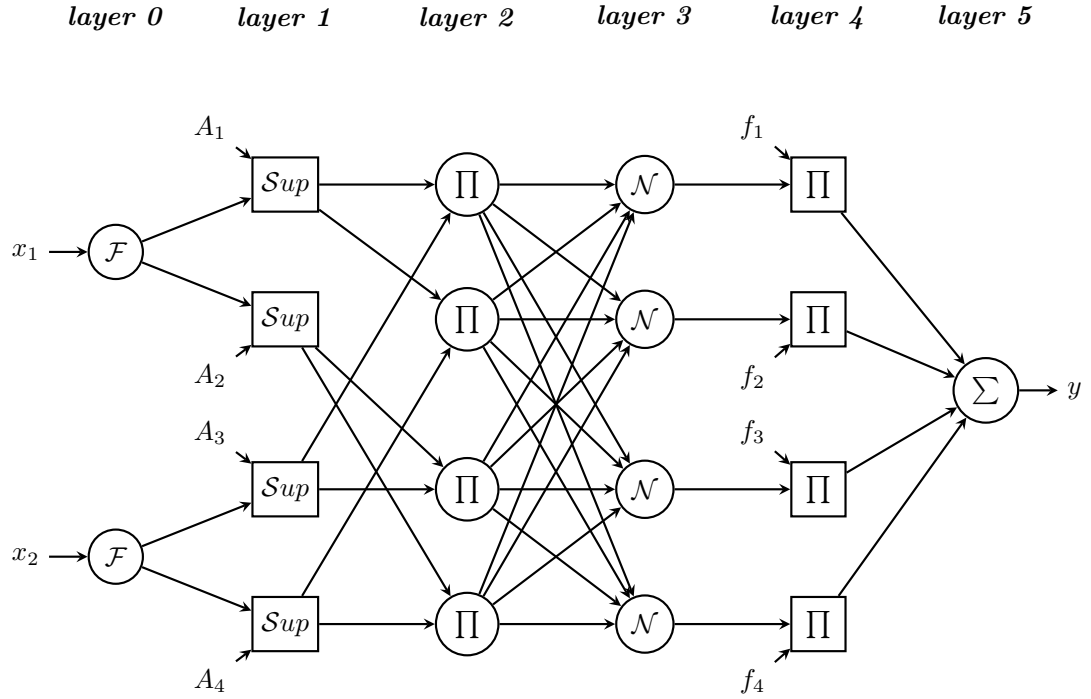


Fig. 5.1 **An example of the extended ANFIS architecture.** The symbol $\mathcal{F}$ denotes the operation of fuzzification. Sup in layer 1 denotes the operation of getting the membership grades for given inputs and antecedents. The symbols Π , $\mathcal{N}$ and Σ denote production, normalisation and summation respectively.

where A_1, A_2 and A_3, A_4 are the antecedent membership functions for input x_1 and x_2 respectively. f_i is the consequent linear function for the i^{th} rule, $i \in \{1, 2, 3, 4\}$. It should be noted that f_i is the output of each fuzzy rule and y is the output of the entire ANFIS model. Specifically, y is the weighted average of all the f_i .

5.2.1 T1 ANFIS

For T1 ANFIS, all antecedents are T1 fuzzy sets. For example, A_1 in the rule above can be represented as $\int_{x \in X_1} \mu_{A_1}(x)/x$ where X_1 is the universe of the primary variable x_1 . The consequents can be either constants or linear functions. The most commonly used type of consequents is the first-order linear function which, for example, can be

defined as:

$$f_i = p_i + q_i x_1 + r_i x_2$$

where p_i, q_i, r_i are the coefficients of the linear function, and they will be referred to as consequent parameters.

Layer 0 Every node i in this layer is a fuzzifier, for which the input is a crisp number and the output $O_{L_0}^i$ is a T1 fuzzy set F_i . The fuzzification method can be either singleton or T1 non-singleton. In the literature, most of the T1 ANFIS research and applications are based on singleton fuzzification, while T1 non-singleton is rarely used and can be seen in [99].

Layer 1 Every node i in this layer has two inputs, the fuzzy input F_j (the output of the incoming node from *layer 0*) and an antecedent fuzzy set A_k . The output of node i is the maximum membership grade of the intersection of F_j and A_k . It can be represented as:

$$O_{L_1}^i = \sup F_j \cap A_k$$

For example, for the first node of layer 1 in Fig. 5.1, F_j is F_1 and A_k is A_1 . Then,

$$O_{L_1}^1 = \sup_{x \in X_1} \mu_{F_1}(x) \star \mu_{A_1}(x)$$

where $\star$ is the t-norm operator which can be either *min* or *prod*. It should be noted that the calculation in this layer is different from [5] where singleton fuzzification is used. Two examples are given in Figs. 5.2 and 5.3 to illustrate $F_j \cap A_k$ for singleton and non-singleton fuzzification respectively.

Layer 2 to 5 The calculations from layer 2 to 5 are the same as that defined in [5], which have been reviewed in Section 2.2.3. The formulas for the outputs of nodes in

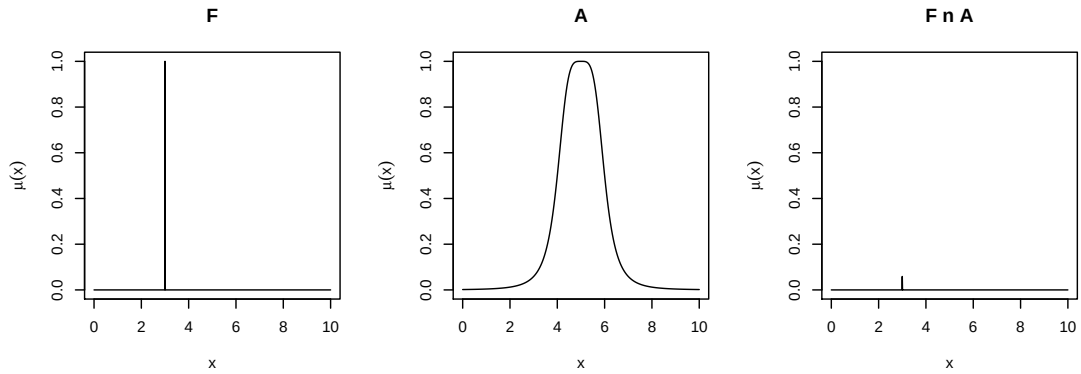


Fig. 5.2 An illustration of $F \cap A$ based on the *min* operator, where F denotes the fuzzy input generated by singleton fuzzification and A denotes the antecedent fuzzy set.

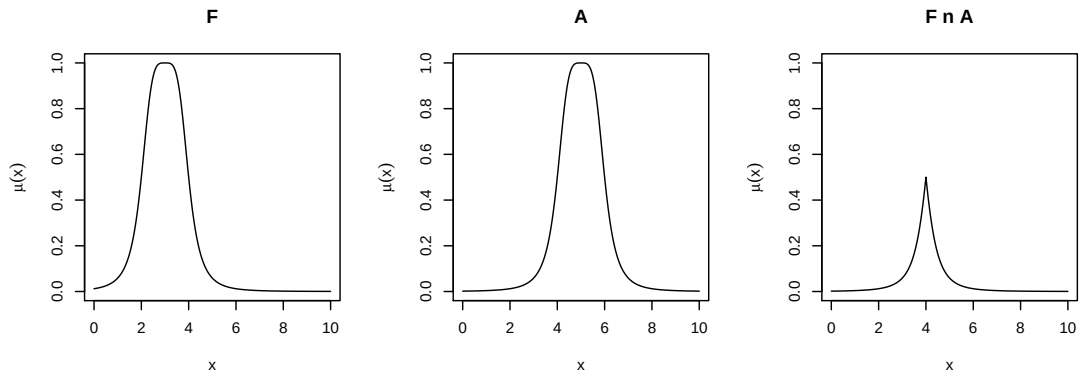


Fig. 5.3 An illustration of $F \cap A$ based on the *min* operator, where F denotes the fuzzy input generated by non-singleton fuzzification and A denotes the antecedent fuzzy set.

these layers are:

$$\begin{aligned}
 O_{L_2}^i &\equiv w_i = O_{L_1}^j O_{L_1}^k \\
 O_{L_3}^i &\equiv \bar{w}_i = \frac{O_{L_2}^i}{\sum_{j=1}^4 O_{L_2}^j} \\
 O_{L_4}^i &= O_{L_3}^i f_i = \bar{w}_i f_i \\
 O_{L_5} &\equiv y = \sum_{i=1}^4 O_{L_4}^i
 \end{aligned}$$

5.2.2 IT2 ANFIS

According to the definition of T2 fuzzy inference systems by Karnik et al. [100], if there exists an IT2 fuzzy set in the antecedent or consequent MFs of an ANFIS, then such an ANFIS can be called as IT2 ANFIS. For simplicity, in this chapter, all the antecedents are IT2 fuzzy sets. For example, A_1 in the rule above can be represented by two T1 MFs, which are lower MF $\underline{A}_1$ and upper MF $\bar{A}_1$. The consequent linear functions can also be defined as intervals such as $[\underline{f}_1, \bar{f}_1]$ for f_1 in Fig. 5.1. For example, as an equivalent representation in [16], $[\underline{f}_i, \bar{f}_i]$ for f_i can be defined as:

$$\begin{aligned}
 \underline{f}_i &= p_i + q_i x_1 + r_i x_2 - s_i - t_i |x_1| - u_i |x_2| \\
 \bar{f}_i &= p_i + q_i x_1 + r_i x_2 + s_i + t_i |x_1| + u_i |x_2|
 \end{aligned}$$

It should be noted that the limitation for the above definition is that s_i, t_i and u_i should be no less than zero in order to guarantee that $\bar{f}_i$ is no less than $\underline{f}_i$. This is also why $|x_1|$ and $|x_2|$ are used. Alternatively, the consequent linear functions can be the same as that defined for T1 ANFIS above. In this case, $\bar{f}_i$ is equal to $\underline{f}_i$.

Layer 0 Similar to T1 ANFIS, every node i in this layer is a fuzzifier. However, the fuzzification method can be singleton [10], T1 non-singleton [12, 16], as well as IT2 non-singleton [15, 101]. Thus, the output $O_{L_0}^i$ can be either a T1 or an IT2 fuzzy set. For example, if IT2 non-singleton is used for node i , then $O_{L_0}^i$ is an IT2 fuzzy set F_i , which can be represented as a lower MF $\underline{F}_i$ and an upper MF $\bar{F}_i$. If singleton or T1

non-singleton fuzzification is used, the output $O_{L_0}^i$ is a T1 fuzzy set. However, to be simple, such a T1 fuzzy set can also be considered as a special IT2 fuzzy set for which $\underline{F}_i$ is the same as $\bar{F}_i$.

Layer 1 In this layer, the output $O_{L_1}^i$ of node i is an interval $[O_{L_1}^i, \bar{O}_{L_1}^i]$, where:

$$\underline{O}_{L_1}^i = \sup \underline{F}_j \cap \underline{A}_k$$

$$\bar{O}_{L_1}^i = \sup \bar{F}_j \cap \bar{A}_k$$

For example, for the first node of layer 1 in Fig. 5.1, F_j is F_1 , and A_k is A_1 . Then,

$$\underline{O}_{L_1}^1 = \sup_{x \in X_1} \mu_{F_1}(x) \star \mu_{A_1}(x)$$

$$\bar{O}_{L_1}^1 = \sup_{x \in X_1} \mu_{\bar{F}_1}(x) \star \mu_{\bar{A}_1}(x)$$

Layer 2 The operation in this layer is similar to that for T1 ANFIS. The difference is that w_i here is an interval which can be represented by $[\underline{w}_i, \bar{w}_i]$. For instance, the output of the first node in this layer can be expressed as:

$$\underline{O}_{L_2}^1 \equiv w_1 = \underline{O}_{L_1}^1 \underline{O}_{L_1}^3$$

$$\bar{O}_{L_2}^1 \equiv \bar{w}_1 = \bar{O}_{L_1}^1 \bar{O}_{L_1}^3$$

Layer 3 For normalising the firing strengths obtained in *layer 2*, a single firing strength w'_i should be selected for each node i . Theoretically, there are infinite combinations of firing strengths, and thus it is impossible to select and compute all of the possible combinations. A common approach is to use the KM algorithms for selecting the combinations. As an alternative to the KM, the DA algorithm can be applied to such process. It will be explained later that the output of *layer 5* for IT2 ANFIS is constituted of $\underline{O}_{L_5}$ and $\bar{O}_{L_5}$, which can be considered respectively as c_l and c_r defined in Chapter 4. It is defined that $\underline{w}'_i$ is the selected firing strength for node i to get $\underline{O}_{L_5}$, and $\bar{w}'_i$ is selected for $\bar{O}_{L_5}$, where both $\underline{w}'_i$ and $\bar{w}'_i$ can be either $\underline{w}_i$ or $\bar{w}_i$. Then, the output $O_{L_3}^i$ can be

represented as $\underline{Q}_{L_3}^i$ and $\bar{O}_{L_3}^i$, where:

$$\underline{Q}_{L_3}^i \equiv \bar{w}_i = \frac{\underline{w}_i'}{\sum_{j=1}^4 \underline{w}_j'}$$

$$\bar{O}_{L_3}^i \equiv \bar{\bar{w}}_i = \frac{\bar{w}_i'}{\sum_{j=1}^4 \bar{w}_j'}$$

It should be noted that $\underline{w}_i'$ does not necessarily need to be smaller than $\bar{w}_i'$. Also, $\underline{Q}_{L_3}^i (\bar{w}_i)$ does not necessarily need to be smaller than $\bar{O}_{L_3}^i (\bar{\bar{w}}_i)$.

Layer 4 The output $\underline{O}_{L_4}^i$ of the i^{th} node in this layer can be represented as $\underline{Q}_{L_4}^i$ and $\bar{O}_{L_4}^i$, where:

$$\underline{Q}_{L_4}^i = \bar{w}_i \underline{f}_i = \underline{Q}_{L_3}^i \underline{f}_i$$

$$\bar{O}_{L_4}^i = \bar{\bar{w}}_i \bar{f}_i = \bar{O}_{L_3}^i \bar{f}_i$$

It should also be noted that $\underline{Q}_{L_4}^i$ does not necessarily need to be smaller than $\bar{O}_{L_4}^i$.

Layer 5 O_{L_5} , which is the final output y , is constituted of $\underline{Q}_{L_5}$ and $\bar{O}_{L_5}$. These can be represented as:

$$O_{L_5} \equiv y = q \underline{Q}_{L_5} + (1 - q) \bar{O}_{L_5}$$

where

$$\underline{Q}_{L_5} = \sum_{i=1}^4 \underline{Q}_{L_4}^i = \sum_{i=1}^4 \bar{w}_i \underline{f}_i$$

$$\bar{O}_{L_5} = \sum_{i=1}^4 \bar{O}_{L_4}^i = \sum_{i=1}^4 \bar{\bar{w}}_i \bar{f}_i$$

and as presented in [14], q is a constant, which is usually set to be 0.5 [16].

5.3 The learning algorithms of ANFIS

The learning algorithms of ANFIS are methods for updating the antecedent and consequent parameters during the training process. As it has been mentioned in Section 2.2.3, the most commonly used learning algorithms for ANFIS is a hybrid process by which the consequent parameters are updated by the LSE algorithm in the forward pass, and the antecedent parameters are updated by the gradient decent method in the backward pass.

5.3.1 The gradient decent method

The learning procedure of ANFIS based on the gradient decent method has been explicitly presented in [5]. This section briefly summarises the learning procedure based on the gradient decent method.

Assuming the given training data set has P entries, the energy function (training error) for the p^{th} entry of the training data is defined as:

$$E_p = (T_p - O_{L5,p})^2$$

where T_p is the target output and $O_{L5,p}$ is the actual output of ANFIS in layer 5. The over all energy function is:

$$E = \sum_{p=1}^P E_p$$

In order to develop a learning procedure that implements gradient descent in E over the parameters, the error rate needs to be calculated for the p^{th} data entry and for each output node. The error rate for the output node in layer 5 is:

$$\frac{\partial E_p}{\partial O_{L5,p}} = -2(T_p - O_{L5,p})$$

The error rate for the internal node i in layer k can be derived by the chain rule:

$$\frac{\partial E_p}{\partial O_{L_k,p}^i} = \sum_{m=1}^{\#(k+1)} \frac{\partial E_p}{\partial O_{L_{(k+1),p}}^m} \frac{\partial O_{L_{(k+1),p}}^m}{\partial O_{L_k,p}^i}$$

where $1 \leq k \leq 4$ and $\#(k+1)$ is the number of nodes in layer $k+1$. If α is a parameter which is updated based on the gradient decent method, we have

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial E_p}{\partial O^*} \frac{\partial O^*}{\partial \alpha}$$

where S is the set of nodes whose outputs depend on α . The derivative of the overall error E with respect to α is

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial E_p}{\partial \alpha}$$

Accordingly, the update formula for α is

$$\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha}$$

where η is a learning rate which can be expressed as

$$\eta = \frac{\lambda}{\sqrt{\sum_{\forall \alpha} (\frac{\partial E}{\partial \alpha})^2}}$$

where λ is the step size (the length of each gradient transition in the parameter space) which can be changed to vary the speed of convergence.

5.3.2 Linear equation for T1 ANFIS

For a single entry of the training data, the final output y of T1 ANFIS described in Section 5.2.1 can be represented by:

$$y = \vec{X} \vec{\theta}$$

where $\vec{X}$ and $\vec{\theta}$ are vectors which can be expressed as:

$$\begin{aligned}\vec{X} &= \begin{pmatrix} \bar{w}_1 & \bar{w}_1 x_1 & \bar{w}_1 x_2 & \dots & \bar{w}_4 & \bar{w}_4 x_1 & \bar{w}_4 x_2 \end{pmatrix} \\ \vec{\theta}^T &= \begin{pmatrix} p_1 & q_1 & r_1 & \dots & p_4 & q_4 & r_4 \end{pmatrix}\end{aligned}$$

5.3.3 Linear equation for IT2 ANFIS

For simplicity, the consequent linear functions for IT2 ANFIS are defined to be the same as those for T1 ANFIS in Section 5.2.1. Then, for a single entry of the training data, the final output y of IT2 ANFIS described above can also be represented by:

$$y = \vec{X} \vec{\theta}$$

where $\vec{X}$ and $\vec{\theta}$ are vectors which can be expressed as Equation 5.1 and 5.2 (the coefficient q for O_{L_5} is set to be 0.5).

$$\vec{X} = \begin{pmatrix} \frac{\bar{w}_1 + \bar{w}_1}{2} & \frac{\bar{w}_1 + \bar{w}_1}{2} x_1 & \frac{\bar{w}_1 + \bar{w}_1}{2} x_2 & \dots & \frac{\bar{w}_4 + \bar{w}_4}{2} & \frac{\bar{w}_4 + \bar{w}_4}{2} x_1 & \frac{\bar{w}_4 + \bar{w}_4}{2} x_2 \end{pmatrix} \quad (5.1)$$

$$\vec{\theta}^T = \begin{pmatrix} p_1 & q_1 & r_1 & \dots & p_4 & q_4 & r_4 \end{pmatrix} \quad (5.2)$$

It should be noted that, if the KM algorithm is applied, $\vec{\theta}$ should firstly be initialised in order to obtain the matrix X by properly selecting the rule firing strengths from the output interval $[\underline{w}_i, \bar{w}_i]$ for each node i in *layer 2*.

5.3.4 LSE and its inappropriateness for IT2 ANFIS

For training data with multiple entries, the output Y for both T1 and IT2 ANFIS can be expressed as the following matrix equation:

$$\vec{Y} = X \vec{\theta}$$

Hence, given the input matrix X and the output vector $\vec{Y}$, the parameters $\vec{\theta}$ can be easily estimated by the LSE algorithm. For T1 ANFIS, the estimated $\vec{\theta}$ can fit the training data with an optimised training error, which is the same as the estimation error (see $\vec{e}'$ defined below) obtained by the LSE. However, it should be noted that the LSE algorithm cannot guarantee an optimised training error for IT2 ANFIS. This is because the estimation error for IT2 ANFIS obtained by the LSE algorithm will almost certainly be different from the final training error for the training data. To clearly illustrate this issue, the estimation error $\vec{e}'$ and the training error $\vec{e}''$ are defined as:

$$\begin{aligned}\vec{e}' &= \vec{Y} - X' \vec{\theta} \\ \vec{e}'' &= \vec{Y} - X'' \vec{\theta}\end{aligned}$$

where $\vec{\theta}$ is estimated by the LSE algorithm with the input X' and the output $\vec{Y}$. It should be noted that when the same training data are used to validate the estimated $\vec{\theta}$, the output $\vec{Y}$ does not change for calculating $\vec{e}''$. However, the input matrix could be different and hence it is defined as X'' .

For T1 ANFIS, X'' is the same as X' . Thus, $\vec{e}''$ will be exactly the same as $\vec{e}'$. This means, for T1 ANFIS, the training error is the estimation error obtained by the LSE algorithm for the training data.

However, the situation for IT2 ANFIS is different from that for T1 ANFIS. For IT2 ANFIS, X' is determined by the initial $\vec{\theta}_0$ when the KM algorithms are applied for selecting the representative firing strengths in *layer 2*. After applying the LSE algorithm, the consequent parameters will be updated to $\vec{\theta}$, which is optimised for the input matrix X' . It should be noted that $\vec{\theta}$ will almost certainly be different from the initial $\vec{\theta}^0$. Thus, when the same training data are then used in the validation process, the input matrix X'' , which is dependent on the new parameters $\vec{\theta}$, will almost certainly be different from the input matrix X' used by the LSE algorithm. For example, $\bar{w}_1$ could be $\frac{w_1}{w_1 + \bar{w}_2 + \bar{w}_3 + \bar{w}_4}$ for X' , while it may be $\frac{w_1}{w_1 + \bar{w}_2 + w_3 + w_4}$ for X'' . Accordingly, $\vec{e}''$ will almost certainly be different from $\vec{e}'$ for IT2 ANFIS. This means, $\vec{\theta}$ is optimised by the LSE algorithm for X' and $\vec{Y}$. However, it does not fit X'' as well as it fits X' . In other words, for IT2 ANFIS,

the LSE method cannot be used to guarantee an optimised error for training data when the KM algorithms are applied for selecting the firing strengths in *layer 2*.

The above explanation of the inappropriateness of LSE for IT2 ANFIS can be summarised in Fig 5.4.

LSE for T1 ANFIS

$$\begin{aligned}
 \text{Training : } & \{\text{training data}\} \Rightarrow \{X', \vec{Y}\} \xrightarrow{\text{LSE}} \vec{\theta} \Rightarrow \vec{e}' = \vec{Y} - X'\vec{\theta} \\
 \text{Validation : } & \{\text{training data}, \vec{\theta}\} \Rightarrow \{X', \vec{Y}, \vec{\theta}\} \Rightarrow \vec{e}'' = \vec{Y} - X'\vec{\theta} \\
 \text{Comparison : } & \vec{e}' \equiv \vec{e}''
 \end{aligned}$$

LSE for IT2 ANFIS

$$\begin{aligned}
 \text{Training : } & \{\text{training data}, \vec{\theta}_0\} \xrightarrow{\text{KM}} \{X', \vec{Y}\} \xrightarrow{\text{LSE}} \vec{\theta} \Rightarrow \vec{e}' = \vec{Y} - X'\vec{\theta} \\
 \text{Validation : } & \{\text{training data}, \vec{\theta}\} \xrightarrow{\text{KM}} \{X'', \vec{Y}, \vec{\theta}\} \Rightarrow \vec{e}'' = \vec{Y} - X''\vec{\theta} \\
 \text{Comparison : } & \vec{e}' \neq \vec{e}''
 \end{aligned}$$

Fig. 5.4 An illustration of the LSE for its inappropriateness in IT2 ANFIS, compared to T1 ANFIS.

5.4 Evaluation

In Section 5.3, it has been discussed how the LSE method can be applied to T1 ANFIS to obtain an optimised error for the training data. However, such optimised error cannot be obtained by the LSE method in IT2 ANFIS in the same way as that in T1 ANFIS. In this section, the well-known classification problem with the IRIS data [102] is applied to further investigate this issue in practice.

For the evaluation models, five fuzzy rules are used where petal length (x_1) and petal width (x_2) are inputs, and species (f) are outputs. Each input has three membership

functions. Specifically, the five rules are:

IF x_1 is $A_{x_1,low}$ **AND** x_2 is $A_{x_2,low}$, **THEN** f_1

IF x_1 is $A_{x_1,mid}$ **AND** x_2 is $A_{x_2,mid}$, **THEN** f_2

IF x_1 is $A_{x_1,high}$ **AND** x_2 is $A_{x_2,high}$, **THEN** f_3

IF x_1 is $A_{x_1,mid}$ **AND** x_2 is $A_{x_2,high}$, **THEN** f_4

IF x_1 is $A_{x_1,high}$ **AND** x_2 is $A_{x_2,mid}$, **THEN** f_5

where the antecedents are gauss-bell shaped T1 and IT2 fuzzy sets for T1 and IT2 ANFIS models respectively. The consequent linear functions are defined as described in Section 5.2.1 for both T1 and IT2 ANFIS models. Thus, in this evaluation, T1 and IT2 ANFIS models have the same number of consequent parameters that are to be estimated by the LSE method.

5.4.1 Training with pure LSE

To clearly show how the LSE method works in T1 and IT2 ANFIS models, only the LSE method are used to tune the consequent parameters. The antecedent membership functions are arbitrarily initialised and fixed during the training process. All 150 instances of the IRIS data are used as the training data. The estimation error $\bar{e}'$ and the training error $\bar{e}''$ defined in Section 5.3.4 are monitored during training epochs based on both batch and online modes. With batch mode, parameters are updated only after the whole training dataset has been presented for each epoch. With online mode, parameters are updated immediately after each input-output pair of training data within each epoch.

For the batch mode, results are shown in Fig. 5.5. It can be observed that the corresponding estimation and training errors for the T1 model are the same. However, a clear difference can be identified between the estimation and training errors for the IT2 model. Also, errors obtained by the T1 ANFIS model do not change during the training epochs, while errors by the IT2 ANFIS model fluctuate. No clear convergence can be found in this figure. Some of the estimation errors for the IT2 ANFIS model are smaller

than the corresponding estimation errors by the T1 ANFIS model. However, almost all training errors of the IT2 ANFIS model are larger than those of the T1 ANFIS model.

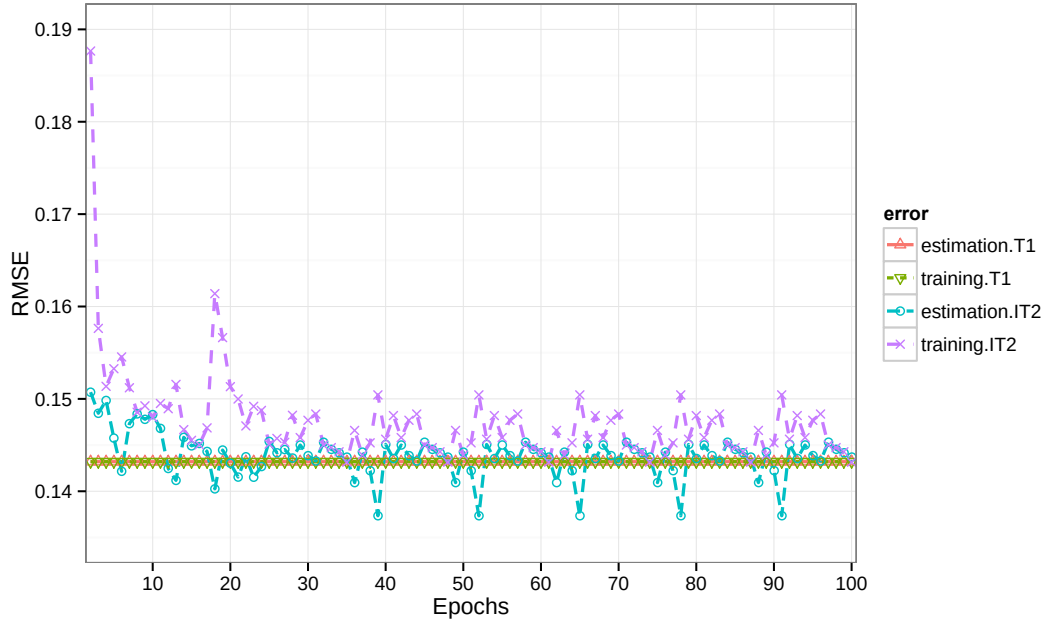


Fig. 5.5 The estimation and training error curves for the training data during the pure LSE training epochs with IRIS data (batch mode).

The results with online training mode are shown in Fig. 5.6. It can be noticed that the results of T1 ANFIS with online mode are the same as those with batch mode above. However, the IT2 model performs differently. Although the difference between estimation and training errors obtained by the IT2 ANFIS model still exist, they cannot be clearly identified especially after a couple of training epochs. Though not clear, error convergence can be found for the IT2 model. However, all the errors obtained by the IT2 ANFIS model are larger than those produced by the T1 ANFIS model.

5.4.2 Training with BP-LSE

More evaluations were with BP-LSE, the hybrid learning algorithm, to further examine how the LSE method performs together with the gradient decent method for T1 and IT2 ANFIS models. Specifically, the parameters in Layer 2 are updated by the BP algorithm,

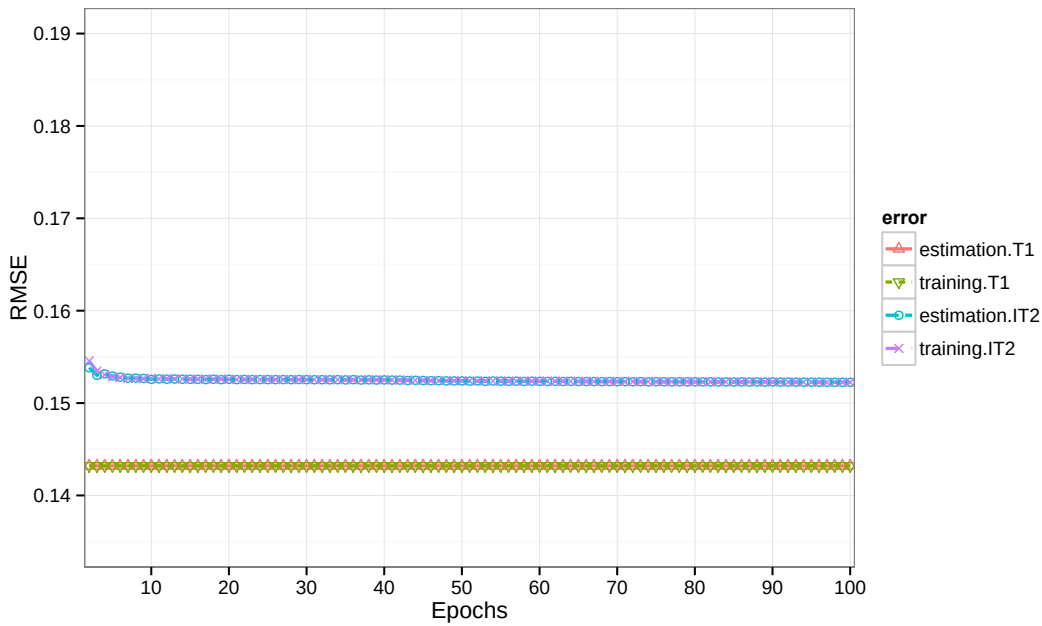


Fig. 5.6 The estimation and training error curves for the training data during the pure LSE training epochs with IRIS data (online mode).

and the parameters in Layer 4 are estimated by the LSE method. Results are shown in Figs. 5.7 and 5.8 for batch and online training modes respectively.

In batch mode, the estimation and training errors for each epoch for T1 ANFIS are still the same. However, it should be noted that such consistency disappears when the T1 ANFIS model is trained with online mode. In contrast, a clear difference between estimation and training errors for IT2 ANFIS can be observed in both batch and online modes.

With BP-LSE, fluctuations can be clearly observed from all the errors except that obtained by the T1 model in the batch mode. All errors converge during training epochs. It can be observed that, in this case, the T1 ANFIS model still gives smaller errors than the IT2 ANFIS model after a couple of training epochs.

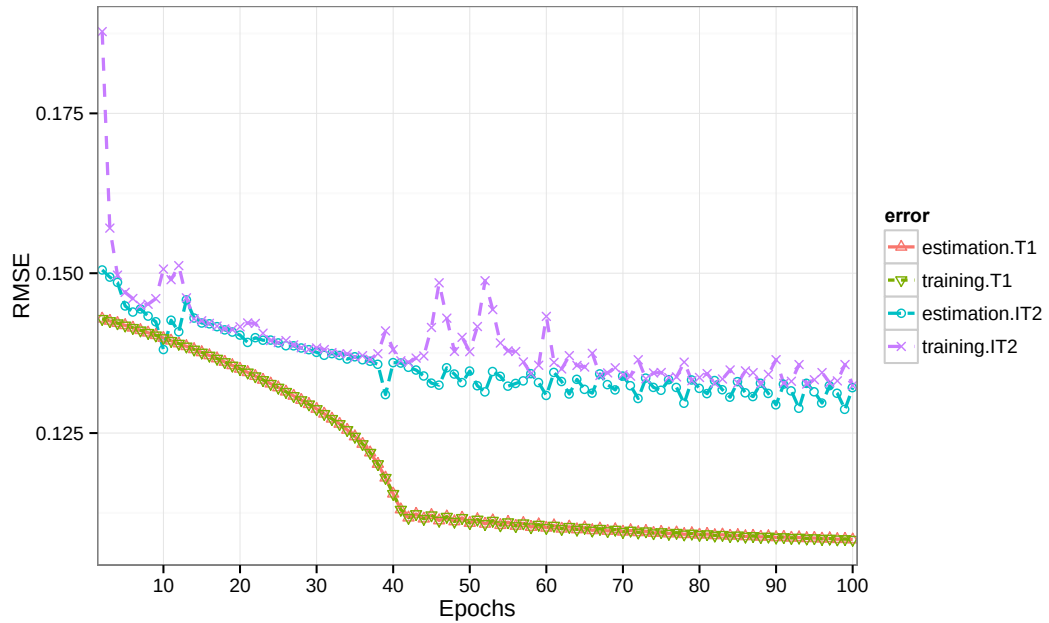


Fig. 5.7 The estimation and training error curves for the training data during the hybrid (BP-LSE) training epochs with IRIS data (batch mode).

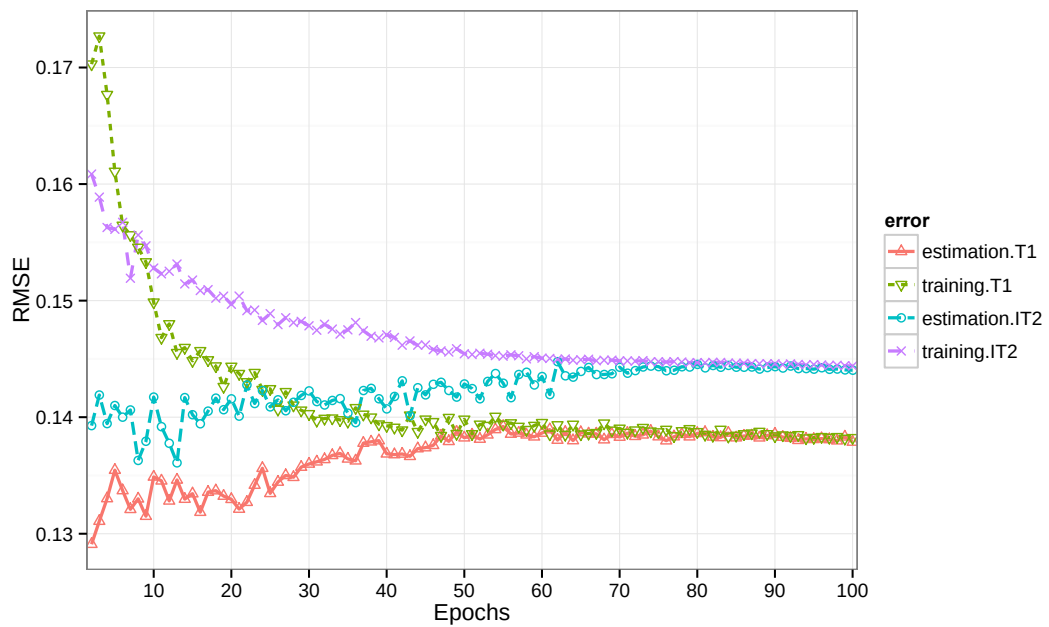


Fig. 5.8 The estimation and training error curves for the training data during the hybrid (BP-LSE) training epochs with IRIS data (online mode).

5.5 Discussion

An IT2 ANFIS can be considered as a combination of all possible embedded T1 ANFIS. According to the definition in Section 5.2.2, the output of IT2 ANFIS is, in fact, the output of its representative embedded T1 ANFIS, which is determined by the consequent parameters if the KM algorithms are applied for selecting the representative firing strengths. Thus, the optimisation of an IT2 ANFIS is, in fact, the optimisation of its representative T1 ANFIS.

As can be clearly observed from our results, the training errors by IT2 ANFIS are generally larger than the corresponding estimation errors. This is because the estimation errors are least square errors optimised by the LSE method for the selected representative embedded T1 ANFIS. However, as discussed in Section 5.3.4, since the consequent parameters are updated after LSE optimisation, the representative embedded T1 ANFIS will also be changed. Hence, the updated consequent parameters do not fit the new representative embedded T1 ANFIS. As a result, the training errors were generally larger than the estimation errors.

It should be noted that when the hybrid learning algorithm (BP-LSE) is applied in online mode, the estimation errors are also not equal to the training errors for T1 ANFIS (see Fig. 5.8). This is because the antecedent parameters are updated within each epoch for every entry of the training data. This makes X'' different from X' when errors are checked after all entries of the training data are learned for each learning epoch.

As can be observed in Fig. 5.8, it is interesting that the estimation errors gradually increase during the training epochs. This is a special property of the LSE method in online mode. The BP-LSE algorithm in online mode does not guarantee an optimised fit for the training data. Therefore the estimation errors for T1 ANFIS in Fig. 5.8 are larger than that in Fig. 5.7.

In Figs. 5.5 and 5.7, errors obtained by IT2 ANFIS during the training epochs show more fluctuation than errors obtained by T1 ANFIS. The fluctuation of errors is in essence related to the change of the matrix X during the training epochs. For T1 ANFIS, the change of the matrix X is only related to the change of antecedent parameters.

However, for IT2 ANFIS, it is related to the change in both antecedent and consequent parameters. Thus, the change of matrix X for IT2 ANFIS is potentially larger than that for T1 ANFIS. Consequently, the larger change of the matrix X leads to the larger fluctuation. This is also the reason for the smaller error fluctuations seen in online mode compared to batch mode: the change of parameters in online mode is relatively smaller than that in batch mode.

In general, IT2 ANFIS produces worse RMSE values than T1 ANFIS. This is because IT2 ANFIS is in fact not well optimised by the LSE method. As discussed in Section 5.3, the LSE method cannot guarantee an optimised error for IT2 ANFIS. Thus, the LSE method is not appropriate for IT2 ANFIS when the KM algorithms are applied for selecting the representative firing strengths in *layer 2*.

5.6 Summary

In this chapter, we have presented an extended six-layer ANFIS architecture suited to both T1 and IT2 ANFIS. By incorporating an extra layer, the fuzzification process is explicitly addressed compared to the original five-layer architecture. Both singleton and non-singleton fuzzification methods are supported by our extended architecture. Through a detailed discussion, we have shown that the LSE method does not exhibit the same behaviour for KM-based IT2 ANFIS compared to T1 ANFIS. This is supported in practice by results presented on the IRIS classification problem. In our evaluation, T1 ANFIS models generally produce smaller errors than IT2 ANFIS models when they are trained by either the pure LSE or the hybrid BP-LSE algorithms. Thus, the LSE method seems not suitable as the training algorithm for IT2 ANFIS models when the KM algorithms are applied for selecting the representative firing strengths in *layer 2*. However, results on a simple dataset is obviously not enough to make a definitive conclusion. It needs to be further investigated with more datasets.

In the next chapter, more experiments are made to compare the performance of T1 and IT2 models further. Also, more accuracy measures, including the new measure

proposed in Chapter 3, will be used to enable fair comparisons and to provide more insight of the results.

Chapter 6

Experimental Comparisons between T1 and IT2 ANFIS

6.1 Introduction

In the previous chapter, we proposed an extended ANFIS architecture and showed that IT2 ANFIS produced larger errors than T1 ANFIS on the well-known IRIS classification problem. In this chapter, more experiments on both synthetic and real-world data are conducted to investigate and compare the performance of T1 and IT2 ANFIS further. Also, in addition to RMSE, other accuracy measures including UMBRAE, discussed in Chapter 3, are also used in comparisons to provide more insights of the results.

As mentioned in Chapter 2.2.3, it has been suggested in [18] that IT2 FIS, in principle, can potentially provide better (and certainly no worse) performance than T1 FIS. This was suggested to be guaranteed by initialising the IT2 membership functions by optimised T1 membership functions. To explore whether such initialisation methods work for IT2 ANFIS in practice, in this chapter, IT2 ANFIS is optimised in three different ways. Our results show that in some circumstances the performance of IT2 ANFIS can be improved when it is initialised with blurred optimised T1 ANFIS parameters. However, in most cases, IT2 ANFIS does not provide a clear performance improvement compared to T1 ANFIS, especially on Mackey-Glass data with large noise.

The remainder of this chapter is organised as follows: Section 6.2 briefly introduces the architecture of ANFIS models used in this chapter; In Section 6.3, experiments evaluating model performance on both synthetic and real-world data are described. The results of these experiments are discussed in Section 6.4. Finally, a summary and suggestions for the future work are provided in Section 6.5.

6.2 ANFIS Architecture

In this chapter, the architecture of T1 and IT2 ANFIS models is based on the work in Chapter 5. Specifically, the antecedent membership functions of T1 ANFIS models are based on the generalised bell-shaped function defined as:

$$\mu(x) = \frac{1}{1 + \left(\left(\frac{x-c}{a} \right)^2 \right)^b}$$

where x is the crisp input and $\{a, b, c\}$ are real-valued parameters, $a, b > 0$. Similarly, the lower and upper membership functions for IT2 ANFIS are also based on generalised bell-shaped functions (see Fig. 6.1 for example), defined as:

$$\begin{aligned} \underline{\mu}'(x) &= \frac{1}{1 + \left(\left(\frac{x-c'}{\underline{a}'} \right)^2 \right)^{b'}} \\ \bar{\mu}'(x) &= \frac{1}{1 + \left(\left(\frac{x-c'}{\bar{a}'} \right)^2 \right)^{b'}} \end{aligned}$$

where $\{\underline{a}', \bar{a}', b', c'\}$ are real-valued parameters, $\bar{a}' > \underline{a}', \underline{a}', b' > 0$.

The consequent membership functions for both T1 and IT2 ANFIS are the same. They are based on first-order linear function f_i (for the i^{th} rule), defined as below (for a model with two inputs x_1 and x_2):

$$f_i = p_i + q_i x_1 + r_i x_2$$

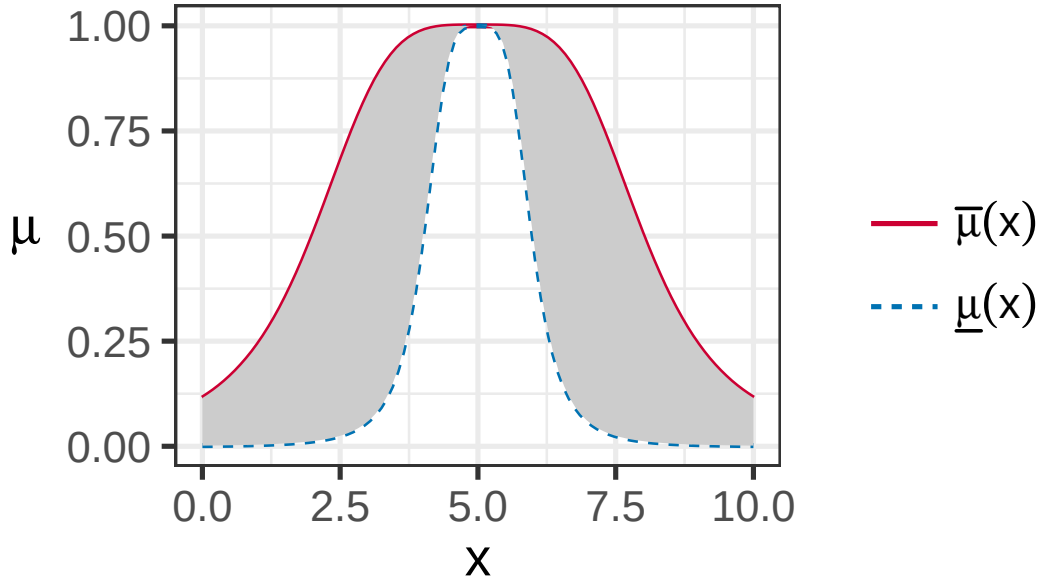


Fig. 6.1 An illustration of the IT2 membership function where $\{\underline{a}' = 1, \bar{a}' = 3, b' = 2, c' = 5\}$.

where p_i, q_i, r_i are the coefficients.

Before optimisation, parameters of T1 and IT2 models are initialised in different ways. Different notations are used in this chapter in order to identify how they are initialised. Specifically, all parameters of T1 ANFIS models are initialised from scratch, and this type of initialisation is denoted by AT1-SCRATCH. IT2 ANFIS models are initialised in three different ways: (i), similar to T1 ANFIS, all IT2 membership functions are initialised from scratch (AT2-SCRATCH); (ii), IT2 membership functions are initialised by optimised T1 membership functions (AT2-OPT). Specifically, the upper and lower membership functions $\underline{\mu}'(x)$ and $\bar{\mu}'(x)$ are set to be the same as corresponding T1 membership function $\mu(x)$; (iii), IT2 membership functions are initialised by blurring optimised T1 membership functions (AT2-BLUR). Specifically, $\underline{a}' = 0.9a$, $\bar{a}' = 1.1a$, $b' = b$ and $c' = c$.

In this chapter, both T1 and IT2 ANFIS models are optimised by the hybrid learning algorithms based on back-propagation and the LSE algorithm. The training process for optimising T1/IT2 ANFIS models is described as follows:

1. Build and initialise T1/IT2 model *anfis* with arbitrarily or specified antecedent and consequent parameters.
2. Set the optimised error $e = \infty$ and the optimised (output) model $anfis' = anfis$.
3. Loop the following sub-steps (training epoch) until e equals to zero or it reaches the predefined maximum training epochs (500 epochs in this chapter):
 - (a) If the model is optimised by sole back-propagation approach, goto Step 3d.
 - (b) In the forward pass, estimate the consequent parameters by the LSE algorithm with training data.
 - (c) Update *anfis* with the estimated consequent parameters.
 - (d) Evaluate the updated model *anfis* with validation data to obtain the corresponding validation error e' (If there are no validation data, then use training data instead in this step.).
 - (e) If $e' < e$, set $e = e'$ and $anfis' = anfis$.
 - (f) In the back propagation process, update the antecedent parameters of *anfis* by the gradient decent method. If required, the consequent parameters can also be updated by the gradient decent method.
4. Set $anfis'$ as the output of the optimisation process.

6.3 Evaluation

To make comparisons between T1 and IT2 ANFIS modes, both synthetic and real-world data are used. Specifically, the well-known MG equation is used to generate synthetic data. For the real-world data, two widely used benchmark problems in the fuzzy logic community are used: the estimation of the low-voltage electrical line length and the estimation of the medium-voltage electrical line maintenance cost in towns.

Comparisons have also been made with a simple method and two commonly used algorithms: linear regression (LR) and support-vector-machines (SVM). The LR and

SVM models are implemented with built-in R functions using default parameters. Results from the literature are also presented for comparisons.

The simple method used in this chapter is described as follows: (i) From all the input variables of the training data, select a single input variable x_i which has the highest correlation with the output y ; (ii) For all the entries of the training data, calculate the median of the ratio y/x_i as the constant coefficient p ; (iii) use px_i as the prediction. Basically, for the MG data, this simple method is doing similar to the naïve method [29], which uses the last available value as the prediction. Thus, we call it naïve method in this chapter.

Five different accuracy measures are used for the results. Specifically, they are RMSE, MASE, MRAE, GMRAE and UMBRAE. The naïve method mentioned above is used to calculate all relative errors.

6.3.1 Mackey-Glass time series prediction

Mackey-Glass time series

The settings to generate MG time series in this chapter are selected from the literature [5, 81, 67]. Specifically, the fourth-order Runge-Kutta method is used and the initial settings are: time step = 0.1; $x(0) = 1.2$ and $x(t) = 0$ for $t < 0$; $\tau = 17$. $x(t)$ is thus derived for $0 \leq t \leq 1200$. In this chapter, both noise-free and noisy MG data are used. For the noisy MG series, five different signal-to-noise ratios are used. They are 0dB, 10dB, 20dB, 30dB and 40dB respectively. The well-known formula in [17] for the signal-to-noise ratio (SNR) is:

$$SNR = 10 \log_{10} \left(\frac{\sigma_s^2}{\sigma_n^2} \right)$$

where σ_s is the standard deviation of the signal and σ_n is the standard deviation of the noise.

Input-output data pairs

1000 input-output data pairs are extracted from the above series for $t = 118$ to 1117 . For each t , $x(t-18)$, $x(t-12)$, $x(t-6)$ and $x(t)$ are used as inputs and $x(t+6)$ is the single output. The first 500 data pairs are used as training set, and remaining 500 data pairs are used as testing set.

ANFIS modelling

All T1 and IT2 ANFIS models are built with 4 inputs (2 membership functions for each input), 1 output and 16 rules.

Results

The results on training and testing data based on different measures (RMSE, MASE, MRAE, GMRAE and UMBRAE) are shown in Tables 6.1 to 6.5. Key results from the literature [5, 81, 67] are also presented in Table 6.1. To provide more insight of the comparisons, standard deviations of both MG series and the added noise are also provided in Table 6.1. Specifically, $SD(mg)$ is the standard deviation of the MG series without noise. $SD(noise)$ is the standard deviation of the added noise. $SD(series)$ is the standard deviation of the MG series with added noise.

In general, it can be clearly observed in Table 6.1 that ANFIS models produced smaller errors than other models in most cases. There is no clear difference between T1 and IT2 ANFIS models. For the noise-free MG data, AT1-SCRATCH produced the same RMSE as T1-ANFIS(Jang). For MG with 0dB noise, our ANFIS models (e.g. AT1-SCRATCH) produced clearly smaller training errors, but larger testing errors than LR and SVM. Results are similar to above with different measures (see Tables 6.2 to 6.5).

Table 6.1 Training/Testing RMSE by different Models for predicting the MG series with different noise.

	0dB	10dB	20dB	30dB	40dB	Noise-free
SD(mg)	0.2267	0.2267	0.2267	0.2267	0.2267	0.2267
SD(noise)	0.2391	0.0756	0.0239	0.0076	0.0024	—
SD(series)	0.3233	0.2363	0.2271	0.2266	0.2266	0.2267
T1-ANFIS(Jang)	—/—	—/—	—/—	—/—	—/—	0.0016/0.0015
T1-ANFIS(Castro)	—/0.2910	—/0.1031	—/0.0333	—/0.0115	—/—	—/0.0070
T1-SA(Almaraashi)	—/0.3273	—/0.1549	—/0.0777	—/0.0352	—/0.2031	—/0.0090
naïve	0.3803/0.3661	0.2090/0.2115	0.1882/0.1903	0.1858/0.1865	0.1860/0.1862	0.1858/0.1857
LR	0.2852/0.2841	0.1345/0.1352	0.1002/0.1014	0.0968/0.0967	0.0970/0.0962	0.0972/0.0962
SVM	0.2644/0.2888	0.0965/0.0981	0.0338/0.0343	0.0152/0.0149	0.0132/0.0128	0.0139/0.0134
AT1-SCRATCH	0.2527/0.3600	0.0892/0.1150	0.0301/0.0376	0.0107/0.0126	0.0040/0.0046	0.0016/0.0015
AT2-SCRATCH	0.2417/0.3357	0.0881/0.1118	0.0298/0.0366	0.0106/0.0131	0.0041/0.0048	0.0020/0.0019
AT2-OPT	0.2492/0.3427	0.0887/0.1139	0.0298/0.0393	0.0106/0.0128	0.0039/0.0045	0.0015/0.0014
AT2-BLUR	0.2485/0.3426	0.0872/0.1201	0.0296/0.0419	0.0104/0.0131	0.0039/0.0046	0.0015/0.0014

Table 6.2 Training/Testing MASE by different Models for predicting MG series with different level of noise.

	0dB	10dB	20dB	30dB	40dB	Noise-free
naïve	1.0000/0.9657	1.0000/1.0450	1.0000/1.0259	1.0000/1.0097	1.0000/1.0046	1.0000/1.0015
LR	0.7771/0.7548	0.6551/0.6318	0.5370/0.5447	0.5243/0.5250	0.5250/0.5209	0.5263/0.5202
SVM	0.6918/0.7721	0.4459/0.4555	0.1713/0.1724	0.0791/0.0760	0.0730/0.0699	0.0778/0.0747
AT1-SCRATCH	0.6713/0.8971	0.4180/0.5137	0.1525/0.1843	0.0546/0.0628	0.0198/0.0224	0.0078/0.0073
AT2-SCRATCH	0.6439/0.8781	0.4161/0.5100	0.1531/0.1833	0.0533/0.0661	0.0204/0.0236	0.0089/0.0086
AT2-OPT	0.6654/0.8630	0.4164/0.5128	0.1515/0.1885	0.0544/0.0643	0.0194/0.0220	0.0074/0.0069
AT2-BLUR	0.6605/0.8562	0.4142/0.5363	0.1499/0.1981	0.0535/0.0653	0.0195/0.0224	0.0076/0.0072

Table 6.3 Training/Testing MRAE by different Models for predicting MG series with different level of noise.

	0dB	10dB	20dB	30dB	40dB	Noise-free
naïve	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000
LR	2.6582/10.170	4.8573/1.8982	2.6052/2.1878	2.7785/2.9343	2.5052/2.0456	2.4256/2.8208
SVM	2.2707/8.4874	3.2276/1.3413	0.7424/0.5595	0.4675/0.3207	0.2997/0.2490	0.3142/0.3999
AT1-SCRATCH	2.3549/16.024	1.7821/1.6807	0.4566/1.5381	0.2857/0.4133	0.0941/0.1413	0.0619/0.0609
AT2-SCRATCH	2.3047/13.774	1.9896/2.4244	0.4095/1.2809	0.2674/0.4390	0.0920/0.1494	0.0680/0.7248
AT2-OPT	2.3836/17.421	1.9880/1.8314	0.4498/1.5057	0.2897/0.4139	0.0893/0.1345	0.0378/0.4082
AT2-BLUR	2.2356/15.251	1.9550/3.2646	0.4466/1.5762	0.3359/0.4472	0.1017/0.1453	0.0520/0.7066

Table 6.4 Training/Testing GMRAE by different Models for predicting MG series with different level of noise.

	0dB	10dB	20dB	30dB	40dB	Noise-free
naïve	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000
LR	0.8116/0.7844	0.6893/0.5384	0.5582/0.5432	0.5492/0.5365	0.5562/0.5296	0.5605/0.5451
SVM	0.6787/0.8058	0.4322/0.3919	0.1628/0.1606	0.0738/0.0707	0.0778/0.0731	0.0820/0.0779
AT1-SCRATCH	0.6713/0.8717	0.4014/0.4222	0.1413/0.1735	0.0528/0.0600	0.0185/0.0206	0.0075/0.0071
AT2-SCRATCH	0.6483/0.8685	0.4129/0.4438	0.1442/0.1693	0.0508/0.0640	0.0192/0.0219	0.0078/0.0075
AT2-OPT	0.6733/0.8486	0.4037/0.4227	0.1414/0.1720	0.0525/0.0628	0.0182/0.0201	0.0069/0.0065
AT2-BLUR	0.6611/0.8487	0.4177/0.4512	0.1406/0.1769	0.0495/0.0613	0.0188/0.0205	0.0073/0.0072

Table 6.5 Training/Testing UMBRAE by different Models for predicting MG series with different level of noise.

	0dB	10dB	20dB	30dB	40dB	Noise-free
naïve	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000
LR	0.8478/0.8168	0.7224/0.6292	0.6305/0.6131	0.6121/0.6064	0.6158/0.6062	0.6170/0.6093
SVM	0.7355/0.8459	0.5337/0.4933	0.2651/0.2563	0.1535/0.1411	0.1478/0.1353	0.1559/0.1519
AT1-SCRATCH	0.7424/0.9079	0.5190/0.5267	0.2395/0.2693	0.1088/0.1298	0.0511/0.0631	0.0223/0.0239
AT2-SCRATCH	0.7223/0.9023	0.5112/0.5358	0.2353/0.2744	0.1081/0.1334	0.0512/0.0613	0.0260/0.0300
AT2-OPT	0.7507/0.8865	0.5160/0.5232	0.2371/0.2716	0.1082/0.1305	0.0493/0.0625	0.0236/0.0246
AT2-BLUR	0.7456/0.8792	0.5189/0.5515	0.2367/0.2816	0.1088/0.1333	0.0502/0.0627	0.0233/0.0245

6.3.2 Low voltage electrical line length estimation

The measurement of electricity line length is useful for many aspects such as the maintenance costs of the network. Compared to the measurement of high and medium-voltage lines, low-voltage lines, which are normally distributed in cities and villages, are more difficult to be measured directly. Therefore, an indirect method for determining the length of low-voltage lines is required. In this section, T1 and IT2 ANFIS models are built to estimate the total length of low-voltage line installed in a rural town.

Data

The dataset used in this section can be obtained from <http://decsai.ugr.es/~casillas/fmlib/ele1-2-495.html>. There are 495 samples in the dataset. Each sample has three attributes, where the first two are inputs, and the other one is the output. Specifically, the inputs are the number of inhabitants in the town and the mean of the distances from the centre of the town to the three furthest clients in it. The output is the total length of low-voltage line installed in the town. Five example samples of the dataset are presented in Table 6.6. The original dataset was randomly divided into five different subsets, with 99 samples in each subset [103]. By joining four of these subsets in a training set and keeping the fifth subset as a test set, five different partitions has been built at the ratio of 80% to 20% to serve as the training and testing data respectively [104]. In this chapter, five-fold cross-validation is applied for all ANFIS models with all the five data partitions provided on the website.

ANFIS modelling

All T1 and IT2 ANFIS models here have 2 inputs (2 membership functions for each input), 1 output and 4 rules.

Table 6.6 **Five example data samples for the problem of low-voltage electrical line length estimation.**

Inhabitants	Distance to Clients	Length of Low Voltage Line
10	648.330017	1773
32	383.329987	1104
4	126.669998	392
36	776.669983	2087
34	343.329987	1224

Results

The RMSE on training and testing data are shown in Table 6.7. The best cross-validation results (referred to as COR [104]) on the data website and the key results (referred to as HA-PAES-MG- K_{max}) from the recent literature [105] are also presented in the table. Note that results from the literature were MSE and they have been converted to RMSE in this chapter. Errors based on other accuracy measures are presented in Table 6.8.

As can be seen in Table 6.7, all T1 and IT2 ANFIS models produced similar training RMSE errors, which are clearly smaller than all other models except HA-PAES-MG- K_{max} . In contrast, the testing RMSE errors by ANFIS models are relatively larger. Note that the testing RMSE by AT2-SCRATCH is particularly larger than other models. However, other measures except MRAE gives similar testing errors for AT2-SCRATCH compared to those for other ANFIS models. Even, the testing errors measured by GMRAE and UMBRAE for AT2-SCRATCH are smaller than the training errors. This is opposite to the results by other measures.

6.3.3 Medium voltage electrical line maintenance cost estimation

This problem involves the estimation of the minimum maintenance costs which are based on a model of the optimal electrical network for Spanish towns. It was originally proposed in [106]. In this problem, there are four input variables, which are the sum of

Table 6.7 Training/Testing RMSE on low-voltage electricity data (5-fold cross-validation).

Training/Testing RMSE	
COR	585.9334/637.2598
HA-PAES-MG-K_{max}	532.2894/619.7451
naïve	739.0226/736.6744
LR	631.2659/644.4992
SVM	582.3314/625.9389
AT1-SCRATCH	562.3696/703.8635
AT2-SCRATCH	570.1036/925.2913
AT2-OPT	562.3460/702.8623
AT2-BLUR	566.5998/794.1014

Table 6.8 Training/Testing errors on low-voltage electricity data (5-fold cross-validation) based on other accuracy measures.

	MASE	MRAE	GMRAE	UMBRAE
naïve	1.0000/1.0038	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000
LR	0.8722/0.8808	3.6052/2.1217	0.9115/0.8775	0.9094/0.8899
SVM	0.7910/0.8512	4.1702/2.1488	0.8379/0.8672	0.8521/0.8888
AT1-SCRATCH	0.7971/0.9121	3.6923/2.2302	0.8256/0.8778	0.8554/0.8982
AT2-SCRATCH	0.8194/0.9296	4.4620/2.5970	0.8985/0.8823	0.9005/0.8945
AT2-OPT	0.7971/0.9114	3.6910/2.2287	0.8254/0.8770	0.8555/0.8980
AT2-BLUR	0.8064/0.9421	3.6365/2.2072	0.8443/0.8929	0.8724/0.9052

the lengths of all streets in the town, the total area of the town, the area that is occupied by buildings, and the energy supply to the town.

Data

The medium-voltage electricity dataset can be downloaded from the web page <http://decsai.ugr.es/~casillas/fmlib/ele2-4-1056.html>. The original dataset has 1056 samples, where 5 samples in the dataset are illustrated in Table 6.9. The whole dataset has been randomly divided into 5 different subsets (four of them with 211 samples and one of

them with 212 samples). By joining four of these subsets in a training dataset and keeping the fifth subset as test dataset, five different partitions has been built at the ratio of 80% to 20% to serve as the training and testing data respectively [104]. In this chapter, five-fold cross-validation is applied. Thus, all the five data partitions provided on the website are used.

Table 6.9 Five example data samples for the problem of medium-voltage electrical line maintenance cost estimation.

Street Length	Area of Town	Building Area	Energy Supply	Maintenance costs
11.00	3.30	54.959999	55.00	4329.330078
4.00	1.20	19.980000	40.00	2016.439941
0.90	0.27	4.500000	1.80	249.419998
2.00	1.20	19.980000	10.00	1044.219971
2.00	1.80	19.980000	30.00	1761.920044

ANFIS modelling

All T1 and IT2 ANFIS models here have 4 inputs (2 membership functions for each input), 1 output and 16 rules.

Results

The RMSE on training and testing data are shown in Table 6.10. The best results listed on the data website (referred to as CHMV01 [107]) and the best results (referred to as HA-PAES-MG- K_{max} and FS-MOGFS+TUN) from the recent literature [105, 108] are also presented in the table. Note that results from the literature were MSE and they have been converted to RMSE in this chapter. Errors based on other accuracy measures are presented in Table 6.11.

As can be observed in the result tables, all T1 and IT2 ANFIS models produced much smaller errors than other models. Errors produced by AT1-SCRATCH are smaller than those by AT2-SCRATCH. AT2-OPT is similar to AT1-SCRATCH. Among all

models, AT2-BLUR performed the best. Note that there are many ∞ values for MRAE and GMRAE since zero errors are not trimmed for calculations.

Table 6.10 Training/Testing RMSE on medium-voltage electricity data (5-fold cross-validation).

Training/Testing RMSE	
CHMV01	135.9265/131.4838
HA-PAES-MG-K_{max}	126.9449/140.8892
FS-MOGFS+TUN	132.6876/140.2997
naïve	665.6653/665.9088
LR	163.4591/164.5443
SVM	121.8285/126.9291
AT1-SCRATCH	78.9684/89.6619
AT2-SCRATCH	85.5813/99.9945
AT2-OPT	77.1671/88.4542
AT2-BLUR	69.9055/82.8083

Table 6.11 Training/Testing errors on medium-voltage electricity data (5-fold cross-validation) based on other accuracy measures.

	MASE	MRAE	GMRAE	UMBRAE
naïve	1.0000/1.0003	1.0000/1.0000	1.0000/1.0000	1.0000/1.0000
LR	0.3022/0.3036	∞ /1.1034	∞ /0.4159	0.5128/0.5130
SVM	0.2309/0.2355	∞ /1.6828	∞ /0.3013	0.4333/0.4330
AT1-SCRATCH	0.1422/0.1605	∞ /0.9630	∞ /0.2080	0.3074/0.3279
AT2-SCRATCH	0.1497/0.1711	∞ /0.9643	∞ /0.2127	0.3321/0.3548
AT2-OPT	0.1384/0.1575	∞ /0.8712	∞ /0.2115	0.3031/0.3245
AT2-BLUR	0.1217/0.1441	∞ /0.8579	∞ /0.1833	0.2725/0.3053

6.3.4 Statistical comparison

For the MG data, statistical comparisons of different models have been made based on the Wilcoxon signed rank test, which is a non-parametric statistical hypothesis test. The significance level is 0.05. Results are shown in Tables 6.12 and 6.13.

Table 6.13 Statistical comparisons of models (A and B) based on absolute errors of testing data.

Model A	MG series						Electricity		Model B
							Low voltage	Medium voltage	
	0dB	10dB	20dB	30dB	40dB	Noisy-free			
naïve	^	^	^	^	^	^	^	^	LR
naïve	^	^	^	^	^	^	^	^	SVM
naïve	^	^	^	^	^	^	^	^	AT1-SCRATCH
naïve	^	^	^	^	^	^	^	^	AT2-SCRATCH
naïve	^	^	^	^	^	^	^	^	AT2-OPT
naïve	^	^	^	^	^	^	^	^	AT2-BLUR
LR	≈	^	^	^	^	^	^	≈	SVM
LR	∨	^	^	^	^	^	^	^	AT1-SCRATCH
LR	∨	^	^	^	^	^	^	^	AT2-SCRATCH
LR	∨	^	^	^	^	^	^	^	AT2-OPT
LR	∨	^	^	^	^	^	^	^	AT2-BLUR
SVM	∨	∨	≈	^	^	^	^	^	AT1-SCRATCH
SVM	∨	∨	≈	^	^	^	^	^	AT2-SCRATCH
SVM	∨	∨	∨	^	^	^	^	^	AT2-OPT
SVM	≈	∨	∨	^	^	^	^	^	AT2-BLUR
AT1-SCRATCH	≈	≈	≈	≈	≈	∨	≈	∨	AT2-SCRATCH
AT1-SCRATCH	≈	≈	≈	≈	≈	^	≈	≈	AT2-OPT
AT1-SCRATCH	^	∨	∨	≈	≈	≈	^	^	AT2-BLUR
AT2-SCRATCH	≈	≈	≈	≈	≈	^	≈	^	AT2-OPT
AT2-SCRATCH	≈	≈	≈	≈	≈	^	≈	^	AT2-BLUR
AT2-OPT	≈	∨	∨	≈	≈	∨	^	^	AT2-BLUR

It is based on the Wilcoxon signed rank test at 0.05 significance level, where $>$ means statistically larger, $<$ means statistically smaller and $\approx$ means no significant difference. The null hypothesis is that there is no significant difference between errors of Models A and B. One-tailed tests were used to examine whether the errors of Model A are significantly greater (or less) than the errors of Models B.

6.4 Discussion

As can be observed in Tables 6.1, 6.7 and 6.10, AT2-SCRATCH models, for which IT2 membership functions were initialised from scratch, produced mostly larger RMSE than AT1-SCRATCH models. Specifically, on the low and medium-voltage electricity data, all the training and testing RMSE values of AT2-SCRATCH models are larger than those of AT1-SCRATCH. This behaviour is consistent with that described in [80].

It has been discussed in [18] that IT2 FIS can be guaranteed to be no worse than T1 FIS by initialising IT2 FIS with optimised T1 FIS. However, it has to be mentioned that this guarantee is only valid for training or validation data. Note that testing data are always unknown and a model can never be guaranteed to fit the testing data as well as it fitted the training or validation data. As can be seen in Tables 6.1, 6.7 and 6.10, all the training RMSE by AT2-OPT models are no larger than the training RMSE by AT1-SCRATCH. However, some testing RMSE of AT2-OPT models are larger than corresponding RMSE of AT1-SCRATCH. Also, it is found that even for the training RMSE, AT2-OPT does not provide a clear performance improvement, especially for the low-voltage electricity data. This may be because T1 ANFIS has reached a local optimum and optimising IT2 ANFIS starting from this local optimum may get stuck.

To address such local optimum issue, AT2-BLUR models are initialised from blurred optimised T1 ANFIS membership functions. Though AT2-BLUR models perform similar to AT2-OPT models for MG series and the low-voltage data, it can be observed in Table 6.10 that AT2-BLUR models produced clearly smaller RMSE than other models. This behaviour is encouraging, and it needs to be further investigated with more data to figure out in which circumstances AT2-BLUR models are better. On the other hand, we have only tried one simple method to blur T1 membership functions for generating IT2 membership functions. In future work, it would be interesting to investigate the effect of different blurring methods.

As mentioned above, errors have also been presented based on different accuracy measures to provide more insights. Since these accuracy measures have been thoroughly discussed in Chapter 3, only a brief discussion, as more evidence, is made here to avoid

repetitions. A clear issue for MASE can be found in the results is that the testing MASE errors for the naïve method are not equal to one. As discussed in Chapter 3, these are somewhat confusing results which may be intuitively interpreted as indicating that the naïve method performs better or worse than the naïve method itself! It has been shown in Chapter 3 that RMSE, MRAE and GMRAE can be easily distorted by a single large outlier. This can also be reflected by the results in this chapter. For example, in Table 6.7, the testing RMSE by AT2-SCRATCH is 925.29. In fact, it would only be around 600 if the 60th row of the testing data in partition 1 were removed from the results. In Table 6.11, there are many ∞ for the training MRAE and GMRAE. This is caused by a zero error prediction on the training data by the naïve method. Hence, it is clear that MRAE and GMRAE cannot be used when zero errors are involved. In contrast, UMBRAE can be applied without the need to trim zero or large outlier errors.

Another interesting point is that our ANFIS models produced much better results (about 40% smaller RMSE) on the medium-voltage electricity data than the best results reported in the literature. The results of ANFIS models on other datasets are also competitive. As can be concluded in Tables 6.12 and 6.13, T1 ANFIS are significantly better than SVM and LR. This indicates that ANFIS is a very good approach and it is worth more attention and investigations for its remarkable ability.

Furthermore, Tables 6.12 and 6.13 show that there is no significant difference between T1 and IT2 ANFIS models on most of our comparisons. This is particularly clear on MG data with large noise, and on the low electricity data. It is also found that on these data, all the methods including the naïve method produce similar results. Hence, it may indicate that on such data there would be no benefit for using IT2 ANFIS.

From the results, we have also found a potential issue for our ANFIS models. As can be seen from Table 6.1, our ANFIS models produced larger testing errors than other models on MG data with large noise. For example, when the signal-to-noise ratio is 0dB, the testing RMSE for the model AT1-SCRATCH was about 0.36 while it was around 0.28 for LR and SVM models. However, in contrast, the training RMSE for the model AT1-SCRATCH was smaller than LR and SVM models. This seems to be an over-fitting

problem. Note that in this chapter, datasets were divided into only training and testing sets so that our results can be compared with literature. Normally, the over-fitting issue can be addressed by using a validation set during the training process.

Though testing errors are more important than training errors, it is arguable that in some circumstances training errors can also indicate the potential capability of a model. In Table 6.1, all the training errors of our ANFIS models are smaller than other models. Moreover, in fact, they are very close to the standard deviation of the added noise. Note that it is an intuition that random noise cannot be predicted. In other words, the smallest RMSE of a model predicting something with random noise should be roughly equal to the standard deviation of the random noise. This provides some indication that ANFIS models may be able to provide better testing performance if the over-fitting issue can be sufficiently addressed.

In conclusion, though IT2 ANFIS might be expected to be better as it has more model parameters than T1 ANFIS, we have shown that this is not always the case, and it is not easy to optimise IT2 ANFIS from scratch. Thus, the use of IT2 ANFIS should be carefully considered since it may not bring a clear performance improvement compared to T1 ANFIS. On the other hand, initialising IT2 ANFIS based on an optimised T1 ANFIS could help in optimising IT2 ANFIS. In particular, it is suggested to initialise IT2 ANFIS based on blurred membership functions of an optimised T1 ANFIS. It should be noted that even based on the above strategy, IT2 ANFIS in some circumstances may not be able to provide a clear performance improvement compared to T1 ANFIS. We do not suggest to use IT2 ANFIS on datasets with large noise when a sophisticated algorithm (e.g. LR or SVM) does not provide clearly better performance than a relatively simple method (e.g. the naïve method).

6.5 Summary

In this chapter, a detailed investigation of the performance comparison between T1 and IT2 ANFIS has been made. Compared to the previous research, more data and more

optimisation strategies for IT2 ANFIS have been used. Specifically, experiments have been conducted on both synthetic and real-world data. According to the results, we have shown that IT2 ANFIS was not easy to optimise from scratch. Thus, the choice of IT2 ANFIS should be carefully considered since it may not produce a clear performance improvement compared to T1 ANFIS. Normally, T1 ANFIS should be enough to provide good results, especially on the data when some easy-to-apply methods such as LR and SVM perform similarly to the naïve method. When IT2 ANFIS is used, we suggest that IT2 ANFIS should be initialised based on blurred membership functions of an optimised T1 ANFIS. We have shown that ANFIS models were very competitive compared to other commonly used models such as LR and SVM. In the future, it would be interesting to try more variations of the initialisation method for IT2 ANFIS based on an optimised T1 ANFIS. It would also be interesting to see comparisons of T1 and IT2 ANFIS, as well as other methods such as LR and SVM, in both accuracy and time efficiency with larger datasets.

The next chapter draws conclusions and limitations that have resulted from this thesis and presents directions for future work.

Chapter 7

Conclusions

This chapter concludes the thesis by summarising the main points and outcomes. Also, this chapter presents a discussion about limitations and future work.

7.1 Thesis Summary

As stated in Chapter 1, the aim of this thesis was to introduce a novel framework for implementing and evaluating T1 and IT2 ANFIS models. To reach this aim, the following objectives were identified:

- *Make it easy to implement T1 and IT2 ANFIS models.*

This is a fundamental prerequisite for the evaluation of T1 and IT2 ANFIS models. Particularly, for non-expert users, it would be useful to provide an easy-to-use software toolbox for them to build and evaluate T1 and IT2 ANFIS models.

- *Reduce the computational costs of IT2 ANFIS.*

It is the fact that IT2 models, with the same number of rules and fuzzy membership functions, are often more computationally expensive than T1 models. This has been a crucial obstacle to the application of IT2 FIS. Thus, it would be of significant value to reduce the computational costs for IT2 ANFIS.

- *Enable fair comparison for T1 and IT2 ANFIS models.*

It is well known that many existing accuracy measures suffer from one or more issues such as poor resistance to outliers and scale dependence. To make fair comparisons, it is necessary to have a suitable measure which does not suffer from the common issues of existing accuracy measures.

- *Evaluate T1 and IT2 ANFIS models.*

Some studies have indicated that IT2 FIS can potentially provide better (and certainly no worse) performance than T1 FIS. However, it was still an open question that whether IT2 ANFIS can provide significantly better performance than T1 ANFIS in practice.

In Chapter 2, we have provided an overview of background material and existing literature that are used and referred to in this thesis. By providing a brief review of fuzzy sets and systems, common notations and definitions for fuzzy logic were established. Since type reduction has been considered to be a bottleneck for the efficiency of a type-2 FIS, a review of the well-known Karnik-Mendel (KM) algorithm and its enhancements has been made. It is well known that all existing KM related algorithms are iterative-based methods. It was still a challenge to improve the efficiency of these algorithms. After a review of ANFIS, it was found that a clear architecture for IT2 ANFIS has not been presented in the literature though some studies have been made on IT2 ANFIS. Also, very few comparisons have been made between the performance of T1 and IT2 ANFIS models. Since proper accuracy measures are very important to conduct objective comparisons of different models, a review of commonly used accuracy measures in the literature has been made focusing on their advantages and disadvantages.

In Chapter 3, to fulfil the third objective described above, a new accuracy measure, called UMBRAE, was proposed to address the issues of commonly used accuracy measures. Evaluation on the proposed measure and other related measures has been made on both synthetic and real-world data. It has been illustrated that UMBRAE combines the best features of various alternative measures without having their common drawbacks.

In Chapter 4, a direct approach, called DA, based on derivatives for determining the switch points without multiple iterations has been proposed to reduce the computational cost of IT2 ANFIS. A mathematical proof has been provided to show that DA correctly determines the switch points for the lower and upper bounds of the centroid of an IT2 fuzzy set. Experimental studies showed that DA clearly outperforms EKM and EIASC for $N \gtrsim 100$ in R. Since it has been pointed out by Prof Jerry Mendel that the performance of EIASC in Section 4.6 is contradicted to the results in [35]. Further investigations have been made to verify these results. These investigations have demonstrated that the various algorithms behave very differently in R and Matlab, and recommendations of the alternative algorithms in various implementation languages are being developed.

In Chapter 5, an extended novel ANFIS architecture was proposed to make it easy to implement T1 and IT2 ANFIS models, which is the first objective of this thesis as re-iterated above. Note that the direct approaches proposed in Chapter 4 were applied to the extended ANFIS architecture to improve computational efficiency. It was pointed out that it was still an open question as to whether the iterative process of estimating the consequent parameters by the LSE method can be successful for IT2 ANFIS. By theoretical analysis and experimental comparisons, we examined this question and showed that the LSE method behaves differently for IT2 ANFIS compared to T1 ANFIS. On average, IT2 ANFIS produced larger RMSE than T1 ANFIS on the IRIS data.

In Chapter 6, more experiments on both synthetic and real-world data were conducted to further investigate and compare the performance of T1 and IT2 ANFIS. Also, in addition to RMSE, other accuracy measures including UMBRAE, were also used in comparisons to provide more insights of the results. It was suggested in [18] that IT2 FIS can be guaranteed to be better than T1 FIS when IT2 membership functions are initialised based on optimised T1 membership functions. To explore whether such initialisation methods work for IT2 ANFIS in practice, in this chapter, IT2 ANFIS is optimised in three different ways. Our results showed that in some circumstances the performance of IT2 ANFIS can be improved when it is initialised based on blurred optimised T1 ANFIS

parameters. However, in most cases, IT2 ANFIS did not provide a clear performance improvement compared to T1 ANFIS according to statistical comparisons. These results indicate that while Mendel's theoretical result remains true, significant performance gains may be rare in practice, limiting the value of the additional computational complexity of IT2 ANFIS.

In the following sections, we conclude the contributions of this thesis, and discuss the limitations of this research and possible directions for further work.

7.2 Contributions

This section summarises the key contributions of this thesis.

A New Accuracy Measure based on Bounded Relative Error

Various accuracy measures have been proposed as the best to use in the past decades. However, many of these measures are not generally applicable due to various issues experienced under certain rare circumstances, such as being infinite or undefined, which may produce misleading results.

In Chapter 3, we proposed a new accuracy measure, UMBRAE, based on bounded relative error. It has been demonstrated that UMBRAE combines the best features of various alternative existing measures in literature without having their common drawbacks. With selectable benchmark, UMBRAE can provide an informative and interpretable result based on bounded relative error. It is less sensitive to forecasting outliers than other measures. It is also symmetric and scale-independent. Though it has been commonly accepted that there cannot be any single best accuracy measure, we suggest that UMBRAE is a good choice for general use when evaluating the performance of forecasting methods.

As an aside, in personal communications, it has recently emerged that UMBRAE has been recommended by Prof Armstrong and Dr Green, who are leading researchers

on forecasting methods and applications, on ForPrin.com and in their working paper “Forecasting Methods and Principles: Evidence-Based Checklists [109]”.

A Direct Approach for Determining the Switch Points

The KM algorithm was the first algorithm proposed to determine the switch points when computing the centroids of IT2 fuzzy sets [31]. Many enhanced algorithms have been proposed to improve the computational efficiency of the KM algorithm. However, all of these algorithms were still based on iterative procedures.

In Chapter 4, we introduced a direct approach, DA, which can be used to determine the switch points based on derivatives without multiple iterations. By empirical simulations, it has been shown that DA clearly outperforms EKM and EIASC for $N \gtrsim 100$ based on our implementations in R.

By further investigations on the different performance of EIASC in R and Matlab, it has been found that many operations such as loops behave differently in R and Matlab. Based on these findings, we have also introduced an enhanced DA algorithm and a non-derivative based direct approach DAND. Compared to other algorithms, DAND ranked first in R and second only to EIASC in Matlab in terms of the computational time based on Monte-Carlo simulations. Furthermore, as DAND does not require multiple iterations, its calculation duration can be determined reliably at implementation time.

An Extended ANFIS Architecture for both T1 and IT2 Models

A fundamental prerequisite for the evaluation of T1 and IT2 ANFIS models is the capability to implement them reliably and efficiently. As mentioned in Section 2.2.3, though many studies have been made on IT2 ANFIS, a clear architecture for IT2 ANFIS has not been presented in the literature compared to that of T1 ANFIS presented in [5]. This somehow has been an obstacle to the research of IT2 ANFIS and its application to real-world problems.

In Chapter 5, we presented an extended novel six-layer ANFIS architecture suited to both T1 and IT2 ANFIS. By incorporating an extra layer 0, the fuzzification process is explicitly addressed compared to the original five-layer architecture. Also, with the extra layer, both singleton and non-singleton fuzzification methods are supported by the extended ANFIS architecture.

A fuzzy logic toolkit for R (named FuzzyR) which includes ANFIS tools has been made available online (<https://CRAN.R-project.org/package=FuzzyR>). With the toolbox, T1 and IT2 ANFIS models can be easily built and evaluated. This would be extremely useful for non-expert users.

An Investigation of the Inappropriateness of LSE for IT2 ANFIS

It was suggested in [17] that the LSE method cannot be applied to an IT2 FIS. This is because the fuzzy basis function expansion, which is the starting point for the LSE method, cannot be obtained without knowing consequent parameters in advance. A possible solution would be to choose some initial values for consequent parameters. However, after an update of consequent parameters, the fuzzy basis function expansion would need to be recomputed. It was still an open question that whether such an iterative process can be successful.

In Chapter 5, through a detailed discussion, we have shown that the LSE method does not exhibit the same behaviour for KM-based IT2 ANFIS compared to T1 ANFIS. This was supported in practice by results presented on the IRIS classification problem. In our evaluation, T1 ANFIS models generally produced smaller errors than IT2 ANFIS models when they were trained by either the pure LSE or the hybrid BP-LSE algorithms. Thus, the LSE method seems not suitable as the learning algorithm for IT2 ANFIS models when the KM algorithms are applied for selecting the representative firing strengths in layer 2. However, given that there is relatively modest empirical evidence on this, it is still too early to make a definitive conclusion. It needs to be further investigated with more empirical studies.

Experimental Comparisons between T1 and IT2 ANFIS

While there has been an increased interest in the research on IT2 ANFIS, very few comparisons have been made between the performance of T1 and IT2 ANFIS models, especially on accessible or reproducible datasets.

In Chapter 5, we showed that IT2 ANFIS produced, on average, larger errors than T1 ANFIS based on the well-known IRIS classification problem. In Chapter 6, more experiments on both synthetic and real-world data were conducted to further investigate and compare the performance of T1 and IT2 ANFIS. The results indicated that IT2 ANFIS was not easy to optimise from scratch. Thus, the choice of IT2 ANFIS should be carefully considered since it may not produce a clear performance improvement compared to T1 ANFIS. Normally, T1 ANFIS should be enough to provide good results, especially on the data when some easy-to-apply methods such as LR and SVM perform similarly to the naïve method. When IT2 ANFIS is used, we suggest that IT2 ANFIS should be initialised based on blurred membership functions of an optimised T1 ANFIS.

7.3 Limitations

This section presents some of the limitations of the work presented in this thesis.

Outlier Resistance of Accuracy Measures

In Chapter 3, we concluded that UMBRAE is more resistant to outliers than other accuracy measures. However, the first experimental comparison in Section 3.3.1 is based on one single outlier for each time series. Whilst such an experiment was sufficient to demonstrate how the new accuracy measure is sensitive to an outlier in comparison with the existing measures, this experiment clearly does not fully explore the behaviour of the new measure under all outlier conditions. Though many of the time series in the M3-Competition datasets, used in the later comparisons in Section 3.3.2, do contain

multiple outliers, the outlier resistance property of accuracy measures in relation to the number and detailed nature of multiple outliers has not been investigated in this thesis.

The Computational Efficiency of Algorithms

In Chapter 4, we showed that DAND was the most efficient in R and second only to EIASC in Matlab. However, it should be noted that the efficiency of an algorithm is closely related to its implementation and the platform used in practice. It means the results may change while the platform changes. Computational complexity comparisons are platform-independent. However, they may not be able to reflect the actual efficiency rankings of algorithms in practice.

The Gradient Descent Method

As introduced in Chapter 5, the gradient decent method is used to update antecedent parameters. It should be noted that the use of the gradient decent method requires mathematical formulas for derivatives of the output with respect to each of the antecedent parameters. These formulas are different for different types of membership functions, and they would become even more complicated when non-singleton fuzzification is applied. In this thesis, for the simplicity of implementation, all antecedent membership functions are generalised bell-shaped, and only singleton fuzzification is applied.

7.4 Future Work

This section presents some potential new directions and future work based on this thesis.

Further Investigation on UMBRAE

Although we have shown that UMBRAE has many advantages, its statistical properties have not been well studied. For example, the manner in which UMBRAE reflects the properties of the distribution of errors is unclear. Moreover, one possible underlying

drawback for UMBRAE is that the bounded error used by UMBRAE will reach the maximum value 1.0 when the benchmark error ($Y_t - F_t^*$) is equal to zero even if the forecast is good. This may produce a biased estimate especially when the benchmark method produces a large number of zero errors. Although this drawback may not be relevant for the majority of real-world data, we would like to address this issue in the future.

Possible Enhancement on DA

It should be noted that the DA algorithm is, in fact, a brute force method which requires the calculations of all partial derivatives. However, it is a first attempt to determine the switch points deterministically by derivatives. While it can be noted in Fig. 4.1 that the partial derivatives are in ascending order and the switch points are located where the sign of partial derivatives changes, it would be interesting to investigate the possibility to determine the switch points without calculating all the partial derivatives.

Comparisons between T1 and IT2 ANFIS

As mentioned in the previous section, only generalised bell-shaped membership functions are used in this thesis. It would be interesting to explore how the performance of T1 and IT2 ANFIS varies if different types of membership functions are used.

We would also like to try more initialisation methods for IT2 ANFIS. For example, the blurring method used in Chapter 6 is to shift a T1 fuzzy set up and down by a fixed amount 10%, which is selected arbitrarily, of the membership grades. It would be worth exploring different degrees of blurring to examine which is the best to use.

On the other hand, although comparisons have been made on different synthetic and real-world datasets, the datasets used in this thesis are relatively small. In the future, it would be interesting to make evaluations on more and larger datasets.

References

- [1] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [2] O. Castillo and P. Melin, “A review on the design and optimization of interval type-2 fuzzy controllers,” *Applied Soft Computing*, vol. 12, no. 4, pp. 1267–1278, 2012.
- [3] D. Soria, J. M. Garibaldi, A. R. Green, D. G. Powe, C. C. Nolan, C. Lemetre, G. R. Ball, and I. O. Ellis, “A quantifier-based fuzzy classification system for breast cancer patients,” *Artificial Intelligence in Medicine*, vol. 58, no. 3, pp. 175–84, 2013.
- [4] A. Pourabdollah, C. Wagner, J. H. Aladi, and J. M. Garibaldi, “Improved Uncertainty Capture for Nonsingleton Fuzzy Systems,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1513–1524, 2016.
- [5] J.-S. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [6] J. Mendel, H. Hagsras, W.-W. Tan, W. W. Melek, and H. Ying, *Introduction To Type-2 Fuzzy Logic Control: Theory and Applications*, 1st ed. Wiley-IEEE Press, 2014.
- [7] R. I. John and C. Czarnecki, “A type 2 adaptive fuzzy inferencing system,” in *Proceedings IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, 1998, pp. 2068–2073.
- [8] R. I. John and C. Czarnecki, “An adaptive type-2 fuzzy system for learning linguistic membership grades,” in *Proceedings IEEE International Conference on Fuzzy Systems*, vol. 3, 1999, pp. 1552–1556.
- [9] G. M. Mendez and O. Castillo, “Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm,” in *Proceedings IEEE International Conference on Fuzzy Systems*, 2005, pp. 230–235.
- [10] G. Mendez and I. Juarez, “First-order interval type-2 TSK fuzzy logic systems using a hybrid learning algorithm,” *WSEAS Transactions on Computers*, vol. 4, no. 4, pp. 378–384, 2005.
- [11] G. Mendez and M. De Los Angeles Hernandez, “Interval Type-2 ANFIS,” in *Innovations in Hybrid Intelligent Systems*, E. Corchado, J. Corchado, and A. Abraham, Eds. Springer Berlin Heidelberg, 2007, vol. 44, pp. 64–71.

- [12] G. Mendez and M. Hernandez, "Interval type-1 non-singleton type-2 fuzzy logic systems are type-2 adaptive neuro-fuzzy inference systems." *International Journal of Reasoning-based Intelligent Systems*, vol. 2, no. 2, pp. 95–99, 2010.
- [13] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Extended Kalman Filter Based Learning Algorithm for Type-2 Fuzzy Logic Systems and Its Experimental Evaluation," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4443–4455, 2012.
- [14] E. Kayacan, O. Cigdem, and O. Kaynak, "Sliding Mode Control Approach for Online Learning as Applied to Type-2 Fuzzy Neural Networks and Its Experimental Evaluation," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 9, pp. 3510–3520, 2012.
- [15] G. Méndez and M. Hernández, "Hybrid learning mechanism for interval A2-C1 type-2 non-singleton type-2 Takagi-Sugeno-Kang fuzzy logic systems," *Information Sciences*, vol. 220, pp. 149–169, 2013.
- [16] M. Hernandez, P. Melin, G. Méndez, O. Castillo, and I. López-Juarez, "A hybrid learning method composed by the orthogonal least-squares and the back-propagation learning algorithms for interval A2-C1 type-1 non-singleton type-2 TSK fuzzy logic systems," *Soft Computing*, vol. 19, no. 3, pp. 661–678, 2015.
- [17] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall, 2001.
- [18] J. M. Mendel, "General type-2 fuzzy logic systems made simple: a tutorial," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 5, pp. 1162–1182, 2014.
- [19] J. G. De Gooijer and R. J. Hyndman, "25 Years of Time Series Forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.
- [20] Z.-K. Gao and N.-D. Jin, "A directed weighted complex network for characterizing chaotic dynamics from time series," *Nonlinear Analysis: Real World Applications*, vol. 13, no. 2, pp. 947–952, 2012.
- [21] Z.-K. Gao, Y.-X. Yang, P.-C. Fang, Y. Zou, C.-Y. Xia, and M. Du, "Multiscale complex network for analyzing experimental multivariate time series," *Europhysics Letters*, vol. 109, no. 3, p. 30005, 2015.
- [22] Z.-K. Gao, M. Small, and J. Kurths, "Complex network analysis of time series," *Europhysics Letters*, vol. 116, no. 5, p. 50001, 2016.
- [23] Z.-K. Gao, Q. Cai, Y.-X. Yang, W.-D. Dang, and S.-S. Zhang, "Multiscale limited penetrable horizontal visibility graph for analyzing nonlinear time series," *Scientific Reports*, vol. 6, p. 35622, 2016.
- [24] J. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," *International Journal of Forecasting*, vol. 8, no. 1, pp. 69–80, 1992.

- [25] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting*, vol. 9, no. 4, pp. 527–529, 1993.
- [26] P. Goodwin and R. Lawton, "On the asymmetry of the symmetric MAPE," *International Journal of Forecasting*, vol. 15, no. 4, pp. 405–408, 1999.
- [27] J. S. Armstrong and R. Fildes, "Making progress in forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 433–441, jan 2006.
- [28] A. Davydenko and R. Fildes, "Measuring forecasting accuracy: The case of judgmental adjustments to SKU-level demand forecasts," *International Journal of Forecasting*, vol. 29, no. 3, pp. 510–522, 2013.
- [29] S. Makridakis and M. Hibon, "The M3-Competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, pp. 451–476, 2000.
- [30] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
- [31] N. Karnik and J. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 1–4, pp. 195–220, 2001.
- [32] M. Melgarejo, "A fast recursive method to compute the generalized centroid of an interval type-2 fuzzy set," in *Proceedings North American Fuzzy Information processing Society*, 2007, pp. 190–194.
- [33] K. Duran, H. Bernal, and M. Melgarejo, "Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set," in *Proceedings North American Fuzzy Information Processing Society*, 2008, pp. 1–5.
- [34] D. Wu and J. M. Mendel, "Enhanced Karnik–Mendel algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.
- [35] D. Wu and M. Nie, "Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems," in *Proceedings IEEE International Conference on Fuzzy Systems*, 2011, pp. 2131–2138.
- [36] G. J. A. KLIR and T. A. A. FOLGER, *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, 1988.
- [37] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.
- [38] D. Dubois and H. M. Prade, *Fuzzy Sets and Systems: Theory and Applications*, ser. Mathematics in science and engineering. Academic Press, 1980.
- [39] R. R. Yager, "FUZZY SUBSETS OF TYPE-II IN DECISIONS," *JOURNAL OF CYBERNETICS*, vol. 10, no. 1-3, pp. 137–159, 1980.
- [40] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," pp. 117–127, 2002.

- [41] J. M. Mendel, F. Liu, and D. Zhai, “ α -Plane Representation for Type-2 Fuzzy Sets: Theory and Applications,” *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1189–1207, 2009.
- [42] C. Wagner and H. Hagra, “Toward General Type-2 Fuzzy Logic Systems Based on zSlices,” *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 4, pp. 637–660, 2010.
- [43] H. Hamrawi, S. Coupland, and R. John, “Type-2 Fuzzy Alpha-Cuts,” *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 3, pp. 682–692, 2017.
- [44] J. M. Mendel, R. I. John, and F. Liu, “Interval Type-2 Fuzzy Logic Systems Made Simple,” *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.
- [45] D. Wu and J. M. Mendel, “Designing practical interval type-2 fuzzy logic systems made simple,” in *Proceedings IEEE International Conference on Fuzzy Systems*, 2014, pp. 800–807.
- [46] L. A. Zadeh, “Outline of a New Approach to the Analysis of Complex Systems and Decision Processes,” pp. 28–44, 1973.
- [47] E. H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, jan 1975.
- [48] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” pp. 116–132, 1985.
- [49] M. Sugeno and G. T. Kang, “Structure identification of fuzzy model,” *Fuzzy Sets and Systems*, vol. 28, no. 1, pp. 15–33, 1988.
- [50] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions, 2nd Edition*. Springer International Publishing, 2017.
- [51] G. C. Mouzouris and J. M. Mendel, “Nonsingleton fuzzy logic systems: theory and application,” *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 1, pp. 56–71, 1997.
- [52] C. Fu, A. Sarabakha, E. Kayacan, C. Wagner, R. John, and J. M. Garibaldi, “Novel, similarity-based non-singleton fuzzy logic control for improved uncertainty handling in quadrotor UAVs,” in *Proceedings IEEE International Conference on Fuzzy Systems*, 2017.
- [53] M. Nie and W. W. Tan, “Towards an efficient type-reduction method for interval type-2 fuzzy logic systems,” in *Proceedings IEEE International Conference on Fuzzy Systems*, 2008, pp. 1425–1432.
- [54] S. Greenfield and F. Chiclana, “Type-reduction of the discretised interval type-2 fuzzy set: What happens as discretisation becomes finer?” pp. 102–109, 2011.
- [55] S. Greenfield, F. Chiclana, R. John, and S. Coupland, “The sampling method of defuzzification for type-2 fuzzy sets: Experimental evaluation,” *Information Sciences*, vol. 189, pp. 77–92, apr 2012.

- [56] J. Li, R. John, S. Coupland, and G. Kendall, "On Nie-Tan operator and type-reduction of interval type-2 fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. PP, no. 99, p. 1, 2017.
- [57] J. M. Mendel, "On KM Algorithms for Solving Type-2 Fuzzy Set Problems," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 426–446, 2013.
- [58] J. M. Mendel and M. R. Rajati, "On Computing Normalized Interval Type-2 Fuzzy Sets," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 5, pp. 1335–1340, 2014.
- [59] M. Nie and W. W. Tan, "Ensuring the Centroid of an Interval Type-2 Fuzzy Set," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 950–963, 2015.
- [60] T. Kumbasar, "Revisiting KM algorithms: A Linear Programming approach," in *Proceedings IEEE International Conference on Fuzzy Systems*, 2015, pp. 1–6.
- [61] F. Liu and J. M. Mendel, "Aggregation using the fuzzy weighted average as computed by the Karnik-Mendel algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 1–12, 2008.
- [62] H. Wu and J. M. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 5, pp. 622–639, 2002.
- [63] J. M. Mendel and X. Liu, "New closed-form solutions for Karnik-Mendel algorithm+defuzzification of an interval type-2 fuzzy set," in *Proceedings IEEE International Conference on Fuzzy Systems*, 2012, pp. 1–8.
- [64] M. Nie and W. W. Tan, "Closed form formulas for computing the centroid of a general type-2 fuzzy set," in *Proceedings IEEE International Conference on Fuzzy Systems*, 2015, pp. 1–8.
- [65] S. Chakraborty, A. Konar, A. Ralescu, and N. R. Pal, "A Fast Algorithm to Compute Precise Type-2 Centroids for Real-Time Control Applications," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 340–353, 2015.
- [66] S. M. Salaken, A. Khosravi, S. Nahavandi, and D. Wu, "Effect of different initializations on EKM algorithm," in *Proceedings IEEE International Conference on Fuzzy Systems*, 2015, pp. 1–6.
- [67] M. Almaraashi and R. John, "Tuning fuzzy systems by simulated annealing to predict time series with added noise," in *proceedings UK Workshop on Computational Intelligence*, 2010, pp. 1–5.
- [68] M. Almaraashi and R. John, "Tuning of type-2 fuzzy systems by simulated annealing to predict time series," in *Proceedings World Congress on Engineering*, vol. 2, 2011, pp. 976–980.
- [69] M. Almaraashi, R. John, and A. Hopgood, "Automatic learning of general type-2 fuzzy logic systems using simulated annealing," pp. 2384–2390, 2014.

- [70] M. Almaraashi, R. John, A. Hopgood, and S. Ahmadi, "Learning of interval and general type-2 fuzzy logic systems using simulated annealing: Theory and practice," *Information Sciences*, vol. 360, pp. 21–42, 2016.
- [71] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
- [72] X.-Z. Wang, Y.-L. He, L.-C. Dong, and H.-Y. Zhao, "Particle swarm optimization for determining fuzzy measures from data," *Information Sciences*, vol. 181, no. 19, pp. 4230–4252, oct 2011.
- [73] Y. Maldonado, O. Castillo, and P. Melin, "Particle swarm optimization of interval type-2 fuzzy systems for FPGA applications," *Applied Soft Computing*, vol. 13, no. 1, pp. 496–508, jan 2013.
- [74] J. Sun, X. Wu, V. Palade, W. Fang, C.-H. Lai, and W. Xu, "Convergence analysis and improvements of quantum-behaved particle swarm optimization," *Information Sciences*, vol. 193, pp. 81–103, jun 2012.
- [75] S. Yang, M. Wang, and L. Jiao, "A quantum particle swarm optimization," pp. 320–324 Vol.1, 2004.
- [76] M. Xi, J. Sun, and W. Xu, "An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 751–759, nov 2008.
- [77] A. Azadeh, M. Saberi, A. Gitiforouz, and Z. Saberi, "A hybrid simulation-adaptive network based fuzzy inference system for improvement of electricity consumption estimation," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11 108–11 117, 2009.
- [78] L.-Y. Wei, T.-L. Chen, and T.-H. Ho, "A hybrid model based on adaptive-network-based fuzzy inference system to forecast Taiwan stock market," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13 625–13 631, 2011.
- [79] A. K. Lohani, R. Kumar, and R. D. Singh, "Hydrological time series modeling: A comparison between adaptive neuro-fuzzy, neural network and autoregressive techniques," *Journal of Hydrology*, vol. 442–443, pp. 23–35, 2012.
- [80] C. Chen, R. John, J. Twycross, and J. M. Garibaldi, "An extended ANFIS architecture and its learning properties for type-1 and interval type-2 models," in *Proceedings IEEE International Conference on Fuzzy Systems*, 2016, pp. 602–609.
- [81] J. R. Castro, O. Castillo, P. Melin, and A. Rodríguez-Díaz, "A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks," *Information Sciences*, vol. 179, no. 13, pp. 2175–2193, 2009.
- [82] R. Fildes, "The evaluation of extrapolative forecasting methods," *International Journal of Forecasting*, vol. 8, no. 1, pp. 81–98, 1992.

- [83] M. P. Clements and D. F. Hendry, "On the limitations of comparing mean square forecast errors," *Journal of Forecasting*, vol. 12, no. 8, pp. 617–637, 1993.
- [84] J. S. Armstrong and R. Fildes, "Correspondence on the selection of error measures for comparisons among forecasting methods," *Journal of Forecasting*, vol. 14, no. 1, pp. 67–71, 1995.
- [85] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler, "The accuracy of extrapolation (time series) methods: Results of a forecasting competition," *Journal of Forecasting*, vol. 1, no. 2, pp. 111–153, 1982.
- [86] Z. O. Olaofe, "A 5-day wind speed & power forecasts using a layer recurrent neural network (LRNN)," *Sustainable Energy Technologies and Assessments*, vol. 6, pp. 1–24, 2014.
- [87] I. Svalina, V. Galzina, R. Lujić, and G. Šimunović, "An adaptive network-based fuzzy inference system (ANFIS) for the forecasting: The case of close price indices," *Expert Systems with Applications*, vol. 40, no. 15, pp. 6055–6063, 2013.
- [88] M. A. Boyacioglu and D. Avci, "An Adaptive Network-Based Fuzzy Inference System (ANFIS) for the prediction of stock market return: The case of the Istanbul Stock Exchange," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7908–7912, 2010.
- [89] A. Esfahanipour and W. Aghamiri, "Adapted neuro-fuzzy inference system on indirect approach TSK fuzzy rule base for stock market analysis," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4742–4748, 2010.
- [90] S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting: Methods and Applications*, ser. Wiley series in management. Wiley, 1998.
- [91] J. S. Armstrong, "Measures of Accuracy," in *Long-Range Forecasting: From Crystal Ball to Computer*, ser. A Wiley-Interscience Publication. Wiley, 1985, pp. 346–354.
- [92] K. Ord, "Commentaries on the M3-Competition An introduction , some comments and a scorecard," *International Journal of Forecasting*, vol. 17, no. 4, pp. 537–541, 2001.
- [93] Z. Chen and Y. Yang, "Assessing forecast accuracy measures," 2004.
- [94] P. J. Fleming and J. J. Wallace, "How not to lie with statistics: the correct way to summarize benchmark results," *Communications of the ACM*, vol. 29, no. 3, pp. 218–221, 1986.
- [95] D. J. Wright, G. Capon, R. Pagé, J. Quiroga, A. A. Taseen, and F. Tomasini, "Evaluation of forecasting methods for decision support," *International Journal of Forecasting*, vol. 2, no. 2, pp. 139–152, 1986.
- [96] C. Chatfield, "Apples, oranges and mean square error," *International Journal of Forecasting*, vol. 4, no. 4, pp. 515–518, 1988.

- [97] J. Armstrong, "Evaluating forecasting methods," in *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Springer US, 2001, vol. 30, pp. 443–472.
- [98] C. Chen, R. John, J. Twycross, and J. Garibaldi, "A direct approach for determining the switch points in the Karnik-Mendel algorithm," *IEEE Transactions on Fuzzy Systems*, vol. PP, no. 99, p. 1, 2017.
- [99] G. Méndez and M. Hernandez, "Hybrid learning for interval type-2 fuzzy logic systems based on orthogonal least-squares and back-propagation methods," *Information Sciences*, vol. 179, no. 13, pp. 2146–2157, 2009.
- [100] N. Karnik, J. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.
- [101] H. MonirVaghefi, M. Rafiee Sandgani, and M. Aliyari Shoorehdeli, "Interval Type-2 Adaptive Network-based Fuzzy Inference System (ANFIS) with Type-2 non-singleton fuzzification," in *Proceedings Iranian Conference on Fuzzy Systems*, 2013, pp. 1–6.
- [102] J. Bezdek, J. Keller, R. Krishnapuram, L. Kuncheva, and N. Pal, "Will the real iris data please stand up?" *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 3, pp. 368–369, 1999.
- [103] O. Cordon, F. Herrera, and P. Villar, "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 667–674, 2001.
- [104] J. Casillas, O. Cordon, and F. Herrera, "COR: a methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 4, pp. 526–537, 2002.
- [105] C. H. Nguyen, V. T. Hoang, and V. L. Nguyen, "A discussion on interpretability of linguistic rule based systems and its application to solve regression problems," *Knowledge-Based Systems*, vol. 88, pp. 107–133, 2015.
- [106] O. Cordon, F. Herrera, and L. Sánchez, "Solving Electrical Distribution Problems Using Hybrid Evolutionary Data Analysis Techniques," *Applied Intelligence*, vol. 10, no. 1, pp. 5–24, 1999.
- [107] O. Cordon, F. Herrera, L. Magdalena, and P. Villar, "A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base," *Information Sciences*, vol. 136, no. 1-4, pp. 85–107, 2001.
- [108] A. S. Koshiyama, M. M. B. R. Vellasco, and R. Tanscheit, *A Novel Genetic Fuzzy System for Regression Problems*. Springer International Publishing, 2016, pp. 85–101.
- [109] J. S. Armstrong and K. C. Green, "Forecasting Methods and Principles: Evidence-Based Checklists," 2017.